12• Computational Science and Engineering

Algorithmic Differentiation and Differencing Abstract | Full Text: PDF (291K)

Bessel Functions Abstract | Full Text: PDF (165K)

Boundary-Value Problems Abstract | Full Text: PDF (297K)

Calculus Abstract | Full Text: PDF (1837K)

Chaos Time Series Analysis Abstract | Full Text: PDF (169K)

Chaotic Systems Control Abstract | Full Text: PDF (274K)

Convolution Abstract | Full Text: PDF (293K)

Correlation Theory Abstract | Full Text: PDF (242K)

Describing Functions Abstract | Full Text: PDF (657K)

Duality, Mathematics Abstract | Full Text: PDF (227K)

Eigenvalues and Eigenfunctions Abstract | Full Text: PDF (253K)

Elliptic Equations, Parallel Over Successive Relaxation Algorithm Abstract | Full Text: PDF (229K)

Equation Manipulation Abstract | Full Text: PDF (519K)

Fourier Analysis Abstract | Full Text: PDF (223K)

Function Approximation Abstract | Full Text: PDF (432K)

Gaussian Filtered Representations of Images Abstract | Full Text: PDF (575K)

Geometry Abstract | Full Text: PDF (115K)

Graph Theory Abstract | **Full Text:** PDF (177K)

Green's Function Methods Abstract | **Full Text:** PDF (238K)

Hadamard Transforms Abstract | **Full Text:** PDF (386K)

Hankel Transforms Abstract | **Full Text:** PDF (361K)

Hartley Transforms Abstract | **Full Text:** PDF (162K)

Hilbert Spaces Abstract | **Full Text:** PDF (223K)

Hilbert Transforms Abstract | **Full Text:** PDF (449K)

Homotopy Algorithm for Riccati Equations Abstract | **Full Text:** PDF (367K)

Horn Clauses Abstract | **Full Text:** PDF (211K)

Integral Equations Abstract | **Full Text:** PDF (180K)

Integro-Differential Equations Abstract | **Full Text:** PDF (197K)

Laplace Transforms Abstract | **Full Text:** PDF (184K)

Least-Squares Approximations Abstract | **Full Text:** PDF (422K)

Linear Algebra Abstract | **Full Text:** PDF (271K)

Lyapunov Methods Abstract | **Full Text:** PDF (258K)

Minimization Abstract | **Full Text:** PDF (109K)

Minmax Techniques Abstract | **Full Text:** PDF (529K)

Nomograms Abstract | **Full Text:** PDF (129K)

Nonlinear Equations Abstract | **Full Text:** PDF (169K)

Number Theory Abstract | Full Text: PDF (974K)

Ordinary Differential Equations Abstract | Full Text: PDF (461K)

Polynomials Abstract | Full Text: PDF (280K)

Probabilistic Logic Abstract | Full Text: PDF (110K)

Probability Abstract | Full Text: PDF (368K)

Process Algebra Abstract | Full Text: PDF (264K)

Random Matrices Abstract | Full Text: PDF (385K)

Roundoff Errors Abstract | Full Text: PDF (185K)

Switching Functions Abstract | Full Text: PDF (142K)

Temporal Logic Abstract | Full Text: PDF (152K)

Theory of Difference Sets Abstract | Full Text: PDF (228K)

Time-Domain Analysis Abstract | Full Text: PDF (418K)

Transfer Functions Abstract | Full Text: PDF (201K)

Traveling Salesperson Problems Abstract | Full Text: PDF (207K)

Vectors Abstract | Full Text: PDF (174K)

Walsh Functions Abstract | Full Text: PDF (172K)

Wavelet Methods for Solving Integral and Differential Equations Abstract | Full Text: PDF (208K)

Wavelet Transforms Abstract | Full Text: PDF (379K)

***z* transforms** Abstract | Full Text: PDF (201K)

Algorithmic Differentiation and Differencing

Standard Article

Louis B. Rall[1] and George F. Corliss[2]

[1]Department of Mathematics University of Wisconsin-Madison, Madison, WI

[2]Department of Electrical and Computer Engineering Marquette University, Milwaukee, WI

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2468

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (291K)

Abstract

The sections in this article are

- An Example
- Algorithmic Generation of Taylor Coefficients
- Solution of Initial-Value Problems
- First-Order Partial Derivatives
- Gradients in Reverse Mode
- Code Lists, Programs, and Computational Graphs
- Differentiation Arithmetics
- Some Applications of Automatic Differentiation
- Implementation of Automatic Differentiation
- Algorithmic Differencing

| | |

Copyright © 1999-2008 All Rights Reserved.

ALGORITHMIC DIFFERENTIATION AND DIFFERENCING

Values of derivatives and Taylor coefficients of functions are required in various computational applications of mathematics to engineering and science. The traditional method for evaluation of derivatives is to use *symbolic differentiation,* in which the rules of differentiation are applied to transform formulas for functions into formulas for their derivatives. Then derivative values are calculated by evaluating these formulas.

Algorithmic differentiation (AD) is an alternative method for evaluation of derivatives. AD is based on the sequence of basic operations, that is, the algorithm used to evaluate the function to be differentiated. Each step in such a sequence consists of an arithmetic operation or the evaluation of some intrinsic function such as the sine or square root. The rules of differentiation are then applied to transform this sequence into a sequence of operations for evaluation of the desired derivative. Thus, AD transforms the *algorithm* for evaluation of a function into an algorithm for evaluation of its derivatives. Since evaluation of functions on digital computers is carried out by means of algorithms in the form of subroutines or programs, AD is particularly suitable in this case. Hence, the historical designation "automatic differentiation" as the process was intended for use on computers. These terms for AD are equivalent.

The processes of algorithmic and symbolic differentiation are based on the same definitions and theorems of differential calculus. They differ in their goals. The purpose of symbolic differentiation being production of formulas for derivatives, while the purpose of AD is computation of values of derivatives. Hence, AD is also referred to as "computational differentiation." Their starting points also differ. Symbolic differentiation begins with formulas, and AD begins with algorithms. If the function to be differentiated is expressed as a formula, then an equivalent algorithm for its evaluation must be derived before AD can be applied. Automatic methods for conversion of formulas to algorithms are well known (1) and are used to produce internal algorithms by calculators and computer programs which accept formulas as input. On the other hand, AD is applicable to functions which are *only* defined algorithmically, as by computer subroutines or programs. For many computational purposes, such as the solution of linear systems of equations, efficient algorithms are preferred to formulas. AD generally requires less computational effort than symbolic differentiation followed by formula evaluation even for functions defined by formulas. The algorithmic approach to derivatives also applies to accurate evaluation of divided differences as described in the final section of this article.

In this article the basic idea of AD is first illustrated by a simple example. This is followed by sections on automatic generation of Taylor series and its application to the computational solution of initial value problems for ordinary differential equations. Subsequent sections deal with evaluating partial derivatives, including gradients, Jacobians, and Hessians, along with various applications, including estimation of sensitivities, solution of nonlinear systems of equations, and optimization. See Ref. 2 for an introduction to AD and its applications.

AN EXAMPLE

To begin on familiar ground, first consider the application of AD to a function defined by a formula. Suppose a circuit, the details of which are unimportant, produces the amplitude-modulated current $I(t)$ given by

$$I(t) = A \sin \Omega t \sin \omega t \tag{1}$$

as a function of time t, where the amplitude A and the frequencies Ω, ω are known constants pertaining to the circuit. If this current flows through a device with inductance L, then the corresponding voltage drop is given by

$$E(t) = L\dot{I}(t) = L\frac{d}{dt}(A \sin \Omega t \sin \omega t) \tag{2}$$

Suppose we want to construct a graph of $I(t)$ and $E(t)$ by evaluating $I(t)$, $E(t) = L\dot{I}(t)$ for a number of values of t and connecting the resulting points to obtain smooth curves. First, consider the evaluation of $I(t)$ itself. Although formula (1) defines $I(t)$, it does not give an explicit step-by-step procedure to compute its value for given t. A straightforward method is to compute the quantities $s_1, \ldots, s_7$ given by

$$\begin{aligned} s_1 &= t \\ s_2 &= \Omega s_1 \\ s_3 &= \sin s_2 \\ s_4 &= A s_3 \\ s_5 &= \omega s_1 \\ s_6 &= \sin s_5 \\ s_7 &= s_4 s_6 \end{aligned} \tag{3}$$

For a given value of t, it is evident that $s_7 = I(t)$. It follows that Eq. (1) and Eq. (3) are equivalent but different representations of the same function. In fact, given Eq. (3), Eq. (1) is obtained by literal substitution for the values of $s_1, \ldots, s_7$, starting with $s_1 = t$. The algorithmic representation in Eq. (3) of the function is called a *code list* (3), because early computers and programmable calculators required this kind of explicit list of operations to evaluate a function. Computers and calculators that accept formulas as input convert them internally to a sequence similar to Eqs. (3) to carry out the evaluation process.

Now the value of the derivative $\dot{I}(t)$ is computed by applying the rules of differentiation to the code list (3) rather than to Eq. (1). This is implemented in several ways. The earliest is *interpretation* of the code list, introduced by Moore (4) and later by Wengert (5). In this method, the code list is used to construct a corresponding sequence of calls to subroutines that compute the appropriate derivative values. For example, if $s = uv$ appears as an entry in the code list, then the values of $u, v, \dot{u}, \dot{v}$ are sent to a subroutine that returns the value $\dot{s} = u\dot{v} + \dot{u}v$. In terms of the usual differentiation formulas, the result of this process is

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 2007 John Wiley & Sons, Inc.

the sequence $\dot{s}_1, \ldots, \dot{s}_7$ given by

$$\begin{aligned} \dot{s}_1 &= 1 \\ \dot{s}_2 &= \Omega \dot{s}_1 \\ \dot{s}_3 &= \dot{s}_2 \cos s_2 \\ \dot{s}_4 &= A \dot{s}_3 \\ \dot{s}_5 &= \omega \dot{s}_1 \\ \dot{s}_6 &= \dot{s}_5 \cos s_5 \\ \dot{s}_7 &= s_4 \dot{s}_6 + \dot{s}_4 s_6 \end{aligned} \tag{4}$$

As in the case of Eq. (3) for evaluating $I(t)$, it is evident that the result of Eq. (4) is $\dot{s}_7 = \dot{I}(t)$. Thus, it is possible to compute the value of the derivative of a function directly from a code list for evaluating the function. Furthermore, literally evaluating the sequence in Eq. (4) gives the formula

$$\dot{I}(t) = A(\omega \sin \Omega t \cos \omega t + \Omega \cos \Omega t \sin \omega t)$$

So in this sense, automatic and symbolic differentiation of a function are equivalent. However, it is important to note that AD is used to compute *numerical* values of $s_1, \ldots, s_7$ and $\dot{s}_1, \ldots, \dot{s}_7$ rather than literal values. Certain values from the code list for evaluating the function are required for evaluating its derivative, in this case s_2, s_4, s_5, s_6.

Another method for automatic differentiation uses the fact that the formulas for derivatives as used in Eq. (4) can themselves be represented by code lists. For example, the derivative of $s = uv$ would be computed in the three steps $d_1 = u\dot{v}$, $d_2 = \dot{u}v$, $d_3 = d_1 + d_2$ from the previously obtained values of $u, v, \dot{u}, \dot{v}$. Then these sublists are inserted at the appropriate place to obtain a code list for the derivative of the original function. Application of this method to Eq. (3) gives

$$\begin{aligned} d_1 &= 1 && (= \dot{s}_1) \\ d_2 &= \Omega d_1 && (= \dot{s}_2) \\ d_3 &= \cos s_2 \\ d_4 &= d_2 d_3 && (= \dot{s}_3) \\ d_5 &= A d_4 && (= \dot{s}_4) \\ d_6 &= \omega d_1 && (= \dot{s}_5) \\ d_7 &= \cos s_5 \\ d_8 &= d_6 d_7 && (= \dot{s}_6) \\ d_9 &= s_4 d_3 \\ d_{10} &= d_5 s_6 \\ d_{11} &= d_9 + d_{10} && (= \dot{s}_7) \end{aligned} \tag{5}$$

This process is called *code transformation* because it transforms a code list for the function into a code list for its derivative. The same method is used to transform Eq. (6) into a code list for the second derivative $\ddot{I}(t)$, if desired, and so on. An early example of code transformation appears in Ref. 3; see also Ref. 6. A third way to implement automatic differentiation, *operator overloading,* is described later.

However implemented, it follows from the chain rule for derivatives that AD succeeds if and only if the function represented by the code list is differentiable at the given value of t. Nondifferentiability causes the step-by-step evaluation of the derivative to break down at some stage, for example, because of attempted division by zero or evaluating a standard function for an invalid argument.

Before leaving this simple example, it is also important to note that AD can be used to compute values of *differentials* as well as derivatives (6). If the value of d_1 in the code list in Eq. (6) is taken to be $d_1 = \tau$ instead of $d_1 = 1$, then the result is $d_{11} = \dot{I}(t)\tau$. By definition, this is the value of the differential $dI = \dot{I}(t)dt$ for dt equal to the given value τ. In some applications, dI is used as an approximation to the increment $\Delta I = I(t + \tau) - I(t)$ for $\tau = dt$ small. If the values of $I(t_0)$ and $\dot{I}(t_0)\tau$ are computed with $\tau = t - t_0$, then the results are also the values of the first two terms of the Taylor series expansion of $I(t)$ at $t = t_0$,

$$I(t) = I(t_0) + \dot{I}(t_0)(t - t_0) + \cdots$$

Next, we show how AD is used to obtain values of as many subsequent terms of the Taylor series as desired for a sufficiently differentiable function, in particular, for series solution of initial-value problems for ordinary differential equations.

ALGORITHMIC GENERATION OF TAYLOR COEFFICIENTS

Suppose that the function $x(t)$ has a convergent Taylor series expansion at $t = t_0$, at least for $|t - t_0|$ sufficiently small. This expansion is written

$$x(t) = a_0 + a_1 + \cdots + a_n + \cdots = \sum_{n=0}^{\infty} a_n \tag{6}$$

where

$$a_n = a_n(t) = \frac{1}{n!} x^{(n)}(t_0)(t - t_0)^n, \quad n = 0, 1, 2, \ldots \tag{7}$$

The numbers $a_0, a_1, \ldots$ are called the *normalized* Taylor coefficients of $x(t)$ expanded at $t = t_0$ with increment $\tau = t - t_0$. For computational purposes, it is convenient to identify the function value $x(t)$ with the vector of its normalized Taylor coefficients, $x(t) \leftrightarrow (a_0, a_1, \ldots, a_n, \ldots)$. If C is a constant, then $C \leftrightarrow (C, 0, 0, \ldots)$, and for the independent variable t, $t \leftrightarrow (t_0, t - t_0, 0, \ldots)$.

Normalized Taylor coefficients can be evaluated by AD for functions defined algorithmically. This method is also called *recursive generation* of normalized Taylor coefficients, and a few special applications prior to the computer era are known (6). By 1959, this technique was incorporated in computer programs created by R. E. Moore and his coworkers at Lockheed Aviation (7).

For simplicity, assume that the function to be expanded is defined by a code list. This reduces the problem to calculating normalized Taylor coefficients of the results of arithmetic operations and standard functions, given the coefficients of their arguments. Typical formulas for these are given later, and more details are in Refs. 6 and 7. The computations involved are *numerical* and are carried out very rapidly on a digital computer. This permits carrying out the Taylor expansion to a much higher degree than ordinarily possible by symbolic methods, which are of course limited

to functions defined by formulas.

Suppose that the normalized Taylor coefficients for the functions $x(t)$ and $y(t)$ expanded at $t = t_0$ with increment $\tau = t - t_0$ are given. The coefficients of the result $z(t) = x(t) \circ y(t)$, where $\circ$ denotes an arithmetic operation, are obtained directly by power series arithmetic. Let $a_0, a_1, \ldots, a_n, \ldots$ and $b_0, b_1, \ldots, b_n, \ldots$ be the coefficients of the series for the operands $x(t)$, $y(t)$, respectively. The coefficients $c_0, c_1, \ldots, c_n, \ldots$ of the result $z(t) = x(t) \pm y(t)$ are given by

$$c_n = a_n \pm b_n, \quad n = 0, 1, \ldots \tag{8}$$

for addition and subtraction, respectively, and by the convolutional formula

$$c_n = \sum_{k=0}^{n} a_k b_{n-k} = \sum_{k=0}^{n} a_{n-k} b_k \tag{9}$$

for multiplication, $z(t) = x(t)y(t)$. The formula for division, $z(t) = x(t)/y(t)$, is obtained by using Eq. (11) for the product $y(t)z(t) = x(t)$, which gives a system of linear equations to be solved for $c_0, c_1, \ldots$. These equations are $b_0 c_0 = a_0$, $b_0 c_1 + b_1 c_0 = a_1$, and so on. If $b_0 \neq 0$, then they are solved in turn for the general coefficient of the result,

$$c_n = \left(a_n - \sum_{k=0}^{n-1} b_{n-k} c_k \right) \frac{1}{b_0}, \quad n = 0, 1, \ldots \tag{10}$$

This formula is *recursive* because the previously computed values of $c_0, \ldots, c_{n-1}$ are needed for calculating c_n, whereas the formulas for addition, subtraction, and multiplication depend only on the coefficients of their arguments.

Generating normalized Taylor coefficients is also required for standard functions, for example, for $z(t) = \sin x(t)$ given the coefficients of $x(t)$. In principle, this is done by substituting of the power series for $x(t)$ in the power series for the sine function and collecting coefficients of like powers of $\tau = t - t_0$, but the algebra is extremely cumbersome. An efficient method formulated by Moore (4) (see Refs. 6 and 7), uses differential equations satisfied by the standard functions. This technique is described in general in the next section. Here, the basic idea is illustrated by the exponential function

$$y(t) = \exp[x(t)] = e^{x(t)}$$

which satisfies the first-order linear differential equation

$$\dot{y}(t) = e^{x(t)} \dot{x}(t) = y(t) \dot{x}(t) \tag{11}$$

Given the normalized Taylor coefficients $a_0, a_1, \ldots$ of $x(t)$ expanded at $t = t_0$ with increment $\tau = t - t_0$, the corresponding coefficients $b_0, b_1, \ldots$ of $y(t)$ are found as follows: First, note that at $t = t_0$, the initial condition $y(t_0) = \exp[x(t_0)]$, that is, $b_0 = \exp(a_0)$. Next, formal term-by-term differentiation of the series for $x(t)$ gives $\overset{2}{x}(t) \leftrightarrow [a_1/\tau, 2a_2/\tau, \ldots, (n+1)a_{n+1}/\tau, \ldots]$ and a similar vector of coefficients for $\overset{2}{y}(t)$. It follows from the differential equation in Eq. (14) that the coefficient $(n+1)b_{n+1}/\tau$ of $\overset{2}{y}(t)$ is equal to the corresponding coefficient in the series for the product $y(t)\overset{2}{x}(t)$ given by Eq. (11). After multiplying by τ and dividing by $(n+1)$, the result is

$$b_{n+1} = \frac{1}{n+1} \sum_{k=0}^{n} (n - k + 1) a_{n-k+1} b_k, \quad n = 0, 1, 2, \ldots \tag{12}$$

As with division, this formula gives b_{n+1} in terms of $b_0, \ldots, b_n$ and the known coefficients of $x(t)$ and hence is recursive. Starting with the initial value $b_0 = \exp(a_0)$, Eq. (15) gives $b_1 = a_1 b_0$, and so on. Note that the exponential function is evaluated only *once* to obtain b_0. Calculating subsequent coefficients is purely arithmetical.

Automatically generating Taylor coefficients is easily implemented by using the algorithmic representation of the function (such as by a code list) to construct calls to subroutines for arithmetic operations and standard functions. Another method is operator overloading, which is described later.

Because computation is inherently finite, the results actually obtained are the coefficients $a_0, \ldots, a_d$ of the Taylor polynomial

$$T_d x(t) = \sum_{n=0}^{d} a_n = \sum_{n=0}^{d} \frac{1}{n!} x^{(n)}(t_0)(t - t_0)^n \tag{13}$$

of degree d rather than the complete Taylor series for a typical function $x(t)$. As before, it is convenient to use the correspondence $T_d x(t) \leftrightarrow (a_0, a_1, \ldots, a_d)$ between a Taylor polynomial and the $(d+1)$-dimensional vector of its coefficients. For given values of t_0, t, AD is used to generate Taylor polynomials of high degree d with a reasonable amount of effort, compared with symbolic differentiation when the latter is applicable. For example, consider calculating $I_{100}(t)$ by AD from the code list in Eq. (3) compared with applying symbolic differentiation 100 times to Eq. (1).

From calculus, the goodness of the approximation of $x(t)$ by $T_d x(t)$ is given by the *remainder term,*

$$\begin{aligned} R_d(t_0)x(t) = x(t) - T_d x(t) &= \frac{1}{d!} \int_{t_0}^{t} x^{(d+1)}(\theta) \left(\frac{t - \theta}{t - t_0} \right)^d d\theta \\ &= x^{(d+1)}(\vartheta) \frac{(t - t_0)^{d+1}}{(d+1)!}, \quad t_0 < \vartheta < t \end{aligned} \tag{14}$$

Moore has shown that automatically generating the Taylor coefficient by using interval arithmetic is a computational procedure that yields guaranteed bounds for the remainder term. For details, see Ref. 7.

SOLUTION OF INITIAL-VALUE PROBLEMS

The principal application of automatically generating Taylor coefficients is not to known functions but rather to unknown functions defined by initial-value problems for ordinary differential equations. The simplest example is for a single, first-order equation

$$\dot{x}(t) = f[t, x(t)], \quad x(t_0) = a_0 \tag{15}$$

where the known function $f(t, x)$ is defined by an algorithm, such as a code list, and the initial value a_0 is given. The method works as follows: The Taylor coefficients $(t_0, t - t_0, 0, \ldots, 0)$ of t are known, and suppose that the coefficients $(a_0, a_1, \ldots, a_d)$ of $T_d x(t)$ have been computed. Then, AD is

used to obtain the coefficients $(b_0, b_1, \ldots, b_d)$ of the Taylor polynomial $T_d f[t, T_d x(t)]$. From the Taylor series for $x(t)$ and the differential, Eq. (18) it follows that

$$a_{d+1} = \frac{1}{d+1} b_d (t - t_0), \quad d = 0, 1, 2, \ldots \tag{16}$$

so the series for $x(t)$ is extended as long as the coefficients b_d can be calculated. This process starts with the initial value a_0. Then because $b_0 = f(t_0, a_0)$, $a_1 = b_0(t - t_0)$, and so on. Generally speaking, the value of $t - t_0$ is small, so multiplication by it and division by $(d + 1)$ reduce the effect of any error in calculating of b_d on the value of the subsequent Taylor coefficient a_{d+1} of $x(t)$.

Initial-value problems for higher order equations

$$x^{(n)}(t) = f[t, x(t), \dot{x}(t), \ldots, x^{(n-1)}(t)] \tag{17}$$

with $x^{(k)}(t_0)$ given for $k = 0, \ldots, n - 1$, are handled in much the same way as the first-order problem. Here,

$$a_k = \frac{1}{k!} x^{(k)}(t_0)(t - t_0)^k, \quad k = 0, 1, \ldots, n - 1$$

so the coefficients of the Taylor polynomial $x_{n-1}(t)$ are known. From these, the coefficients $b_0, \ldots, b_{n-1}$ of f_{n-1} $[t, x(t), \ldots, x^{(n-1)}(t)]$ are calculated. Then Eq. (19) gives a_n and likewise subsequent coefficients of $x(t)$. Another method is to transform higher order differential equations into a system of first-order equations. This is done by the substitutions $x_k(t) = x^{(k)}(t)$, $k = 1, \ldots, n - 1$ that give

$$\dot{x}_1(t) = x_2(t)$$

and

$$\dot{x}_2(t) = x_3(t)$$
$$\cdots$$
$$\dot{x}_{n-1}(t) = f[t, x_1(t), \ldots, x_{n-1}(t)]$$

This is a special case of the general first-order system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}[t, \boldsymbol{x}(t)] \tag{18}$$

where $\boldsymbol{x}(t) = [x_1(t), \ldots, x_m(t)]$ and $\boldsymbol{f}(t) = [f_1(t, \boldsymbol{x}(t)], \ldots, f_m[t, \boldsymbol{x}(t)]$ are m-dimensional vectors of functions of t. It is assumed that $\boldsymbol{f}[t, \boldsymbol{x}(t)]$ is a known function of its arguments. Given the initial condition

$$\boldsymbol{x}(t_0) = \boldsymbol{a}_0 = (a_{10}, a_{20}, \ldots, a_{m0})$$

the Taylor expansion of $\boldsymbol{x}(t)$ is carried out similarly as for a single equation, but of course more arithmetic is involved (7). It is assumed as before that the functions $f_1(t), \ldots, f_m(t)$ have representations suitable for automatic differentiation.

For example, recurrence relationships for the standard functions $c(t) = \cos x(t)$ and $s(t) = \sin x(t)$, are obtained from the first-order system

$$\dot{c}(t) = -s(t)\dot{x}(t)$$

and

$$\dot{s}(t) = c(t)\dot{x}(t)$$

which these functions satisfy, together with the initial conditions $c(t_0) = \cos x(t_0)$, $s(t_0) = \sin x(t_0)$, that is, $c_0 = \cos a_0$, $s_0 = \sin a_0$, where $c(t) \leftrightarrow (c_0, c_1, \ldots)$, $s(t) \leftrightarrow (s_0, s_1, \ldots)$, and $x(t) \leftrightarrow (a_0, a_1, \ldots)$. The results are

$$c_{n+1} = -\frac{1}{n+1} \sum_{k=0}^{n} (n - k + 1) a_{n-k+1} s_k$$

and (19)

$$s_{n+1} = \frac{1}{n+1} \sum_{k=0}^{n} (n - k + 1) a_{n-k+1} c_k$$

$n = 0, 1, 2, \ldots$ (see Refs. 6 and 7). Equations (11) and (19) are sufficient to compute an arbitrary number of Taylor coefficients of the function defined by the code list in Eq. (3), given the values of t_0 and t.

Further work on solving differential equations by automatically generating Taylor series has been done by Chang and Corliss (8) and the computer program ATOMFT (9) developed for this purpose. Using interval arithmetic for bounding solutions of initial-value problems, as originated by Moore (see Ref. 7), runs into a technical problem called the "wrapping effect," when applied to systems. This causes interval error bounds to increase unrealistically rapidly. Moore (10) proposed automatic coordinate transformations to alleviate this problem. Further work by Lohner (11) produced an efficient method for minimizing the wrapping effect that is implemented in the computer program AWA.

FIRST-ORDER PARTIAL DERIVATIVES

Automatic differentiation is also effective for evaluating partial derivatives and Taylor coefficients of functions of several variables. For example, suppose that the resonant frequency $f = f(R, L, C)$ of a certain circuit is defined by the formula

$$f = \frac{1}{2\pi} \sqrt{\frac{1}{LC} - \left(\frac{R}{L}\right)^2} \tag{20}$$

AD requires algorithmic representations of functions, in this case, the code list

$$\begin{aligned} s_1 &= R \\ s_2 &= L \\ s_3 &= C \\ s_4 &= s_2 s_3 \\ s_5 &= 1/s_4 \\ s_6 &= s_1/s_2 \\ s_7 &= \mathrm{sqr}(s_6) \\ s_8 &= s_5 - s_7 \\ s_9 &= \mathrm{sqrt}(s_8) \\ s_{10} &= (1/2\pi) s_9 \end{aligned} \tag{21}$$

for evaluating $f(R, L, C)$ at given values of R, L, and C. In Eq. (30), the standard functions $\mathrm{sqr}(s) = s^2$ and $\mathrm{sqrt}(s) = \sqrt{s}$ have been introduced for convenience, and it is assumed that the value of the constant $1/2\pi$ is available as a single quantity.

Values of the partial derivatives $\partial f/\partial R$, $\partial f/\partial L$, and $\partial f/\partial R$ are useful in a number of applications. For example, $\partial f/\partial R$ can be taken as a measure of the *sensitivity* of the value

of f to a change in R with L and C held constant because $\Delta f = f(R + \Delta R, L, C) - f(R, L, C)$ is approximately equal to $(\partial f/\partial R)\ \Delta R$ in this case, and similarly for the other variables. Partial derivatives are also used to estimate the impact of round-off error on final results of calculations (12), and the *gradient* of f, which is the vector

$$\nabla f = \left(\frac{\partial f}{\partial R}, \frac{\partial f}{\partial L}, \frac{\partial f}{\partial C}\right)$$

appears in optimization and other problems.

The obvious, but usually not the most efficient way to evaluate partial derivatives is to apply the rules for differentiation to the code list for the function, as in the case of ordinary derivatives and Taylor coefficients. In terms of differentials, this gives

$$\begin{aligned}
ds_1 &= dR \\
ds_2 &= dL \\
ds_3 &= dC \\
ds_4 &= s_2 ds_3 + s_3 ds_2 \\
ds_5 &= -ds_4/\mathrm{sqr}(s_4) \\
ds_6 &= (ds_1 - s_6 ds_2)/s_2 \\
ds_7 &= 2s_6 ds_6 \\
ds_8 &= ds_5 - ds_7 \\
ds_9 &= ds_8/(2s_9) \\
ds_{10} &= (1/2\pi)\, ds_9
\end{aligned} \tag{22}$$

The result of evaluating Eq. (32) along with Eq. (30) is the *total differential* $df = ds_{10}$ of f,

$$df = \frac{\partial f}{\partial R}\, dR + \frac{\partial f}{\partial L}\, dL + \frac{\partial f}{\partial C}\, dC$$

(see Refs. 5 and 6). This result is the same as the normalized Taylor coefficient f_1 of f computed from the Taylor polynomials of degree one with coefficients (R, dR), (L, dL), and (C, dC), respectively. It is evident that the value of $\partial f/\partial R$ is obtained from Eq. (32) for $dR = 1$, $dL = 0$, $dC = 0$, and similarly for the other two partial derivatives of f. Thus, the computational sequence Eq. (32) has to be repeated three times to obtain the components of the gradient ∇f of f. This method is called the *forward* mode of AD because the intermediate calculations are done in the same order as in the code list in Eq. (30) for evaluating the function. An often more efficient method is the *reverse* mode described in the next section.

Note that if $(LC)^{-1} < (R/L)^2$, then evaluating f in real arithmetic by Eq. (30) breaks down at s_9 because of a negative argument for the square root, whereas for $(LC)^{-1} = (R/L)^2, f = 0$ but differentiation breaks down at ds_9 because of the indicated division by $s_9 = 0$.

As pointed out by Wengert (5), higher partial derivatives can be recovered from Taylor coefficients. If the Taylor polynomials with coefficients $(R, dR, 0)$, $(L, dL, 0)$, and $(C, dC, 0)$ are substituted for the respective variables, then the value of the second normalized Taylor coefficient $f_2 = f_2(dR, dL, dC)$ of the function is given by

$$f_2 = \frac{1}{2}\left(\frac{\partial^2}{\partial R^2}\, dR^2 + \frac{\partial^2 f}{\partial L^2}\, dL^2 + \frac{\partial^2 f}{\partial C^2}\, dC^2\right) + \frac{\partial^2 f}{\partial R \partial L}\, dR dL + \frac{\partial^2 f}{\partial R \partial C}\, dR dC + \frac{\partial^2 f}{\partial L \partial C}\, dL dC$$

Thus, for $dR = 1$, $dL = dC = 0$, the value of the second partial derivative with respect to R is given by $\partial^2 f/\partial R^2 = 2f_2(1, 0, 0)$, and similarly for $\partial^2 f/\partial L$ and $\partial^2 f/\partial^C$. Then the values of the mixed, second partial derivatives are computed by solving linear equations. For example, the choice $dR = dL = 1$, $dC = 0$ gives

$$\frac{\partial^2}{\partial R \partial L} = f_2(1, 1, 0) - 2f_2(1, 0, 0) - 2f_2(0, 1, 0)$$

The method of code transformation (6) is likewise applicable to evaluating individual partial derivatives or gradient vectors. The idea is to obtain code lists from Eq. (32) that contain the needed entries. For example, the code list

$$\begin{aligned}
dr_1 &= dR \\
dr_2 &= dr_1/s_2 \\
dr_3 &= 2s_6 \\
dr_4 &= dr_2 dr_3 \\
dr_5 &= -dr_4 \\
dr_6 &= 2s_9 \\
dr_7 &= dr_5/dr_6 \\
dr_8 &= (1/2\pi)\, dr_7
\end{aligned}$$

produces the differential coefficient $dr_8 = (\partial f/\partial R)dR$. Similar code lists for $(\partial f/\partial L)dL$ and $(\partial f/\partial C)dC$ can be adjoined to the code list in Eq. (30) for $f(R, L, C)$. This increases the length of the code list by approximately a factor of three and results in the corresponding increase in computational effort for evaluating the gradient, compared to the value of the function alone. Once the code lists for the first partial derivatives have been formed, they are used to construct code lists for second partial derivatives, and so on. This leads eventually to a large amount of code compared to the repetitive generation of Taylor coefficients described previously.

GRADIENTS IN REVERSE MODE

The reverse mode of automatic differentiation appears in the 1976 paper by Linnainmaa (13), the Ph.D. thesis of Speelpennig (14), and later in the paper (12) by Iri. As the name implies, this computation proceeds in the reverse order of the sequence of operations used to evaluate the function. For example, consider the function $f(R, L, C)$ defined by the code list in Eq. (30). The first partial derivatives of this function are

$$\frac{\partial f}{\partial R} = \frac{\partial s_{10}}{\partial s_1}$$

$$\frac{\partial f}{\partial L} = \frac{\partial s_{10}}{\partial s_2}$$

and

$$\frac{\partial f}{\partial C} = \frac{\partial s_{10}}{\partial s_3}$$

These quantities are obtained by differentiating s_{10} beginning with s_{10}, then working backward through the code list:

$$\begin{aligned}
&\frac{\partial s_{10}}{\partial s_{10}} = 1 \\
&\frac{\partial s_{10}}{\partial s_9} = \frac{1}{2\pi} \\
&\frac{\partial s_{10}}{\partial s_8} = \frac{\partial s_{10}}{\partial s_9}\frac{\partial s_9}{\partial s_8} = \frac{\partial s_{10}}{\partial s_9}\frac{1}{2s_9} \\
&\frac{\partial s_{10}}{\partial s_7} = \frac{\partial s_{10}}{\partial s_8}\frac{\partial s_8}{\partial s_7} = -\frac{\partial s_{10}}{\partial s_8} \\
&\frac{\partial s_{10}}{\partial s_6} = \frac{\partial s_{10}}{\partial s_7}\frac{\partial s_7}{\partial s_6} = \frac{\partial s_{10}}{\partial s_7}2s_6 \\
&\frac{\partial s_{10}}{\partial s_5} = \frac{\partial s_{10}}{\partial s_8}\frac{\partial s_8}{\partial s_5} = \frac{\partial s_{10}}{\partial s_8} \\
&\frac{\partial s_{10}}{\partial s_4} = \frac{\partial s_{10}}{\partial s_5}\frac{\partial s_5}{\partial s_4} = -\frac{\partial s_{10}}{\partial s_5}\frac{1}{\mathrm{sqr}(s_4)} \\
&\frac{\partial s_{10}}{\partial s_3} = \frac{\partial s_{10}}{\partial s_4}\frac{\partial s_4}{\partial s_3} = \frac{\partial s_{10}}{\partial s_4}s_2 \\
&\frac{\partial s_{10}}{\partial s_2} = \frac{\partial s_{10}}{\partial s_6}\frac{\partial s_6}{\partial s_2} + \frac{\partial s_{10}}{\partial s_4}\frac{\partial s_4}{\partial s_2} = -\frac{\partial s_{10}}{\partial s_6}\frac{s_1}{\mathrm{sqr}(s_2)} + \frac{\partial s_{10}}{\partial s_4}s_3 \\
&\frac{\partial s_{10}}{\partial s_1} = \frac{\partial s_{10}}{\partial s_6}\frac{\partial s_6}{\partial s_1} = \frac{\partial s_{10}}{\partial s_6}\frac{1}{s_2}
\end{aligned} \tag{23}$$

In this simple example, 19 arithmetic operations are required to evaluate the partial derivatives of f in reverse mode, whereas the forward mode based on Eq. (32) takes 24 operations. The following analysis indicates that the savings are generally greater as the number of independent variables increases.

In general, suppose that the function $f = f(x_1, \ldots, x_m)$ of m variables is represented by the code list $s = (s_1, \ldots, s_n)$, where the values of $x_1, \ldots, x_m$ are assigned to $s_1, \ldots, s_m$, respectively. The forward and reverse modes of AD result from applying the chain rule to s in different ways. In the forward mode,

$$\frac{\partial t_i}{\partial x_j} = \sum_{k \in K_i} \frac{\partial t_i}{\partial t_k}\frac{\partial t_k}{\partial x_j}, \quad i = 1, \ldots, n: \quad j = 1, \ldots, m \tag{24}$$

where K_i denotes the set of indices $k < i$ such that s_i depends explicitly on s_k. Consequently, there are mn quantities to evaluate in this case. For the reverse mode, let I_j denote the set of indices $i > j$ such that s_i depends explicitly on s_j. Then,

$$\frac{\partial s_n}{\partial s_n} = 1$$

and

$$\frac{\partial s_n}{\partial s_j} = \sum_{i \in I_j} \frac{\partial s_n}{\partial s_i}\frac{\partial s_i}{\partial s_j}, \quad j = n-1, \ldots, 1 \tag{25}$$

giving a total of $(n-1)$ quantities to evaluate. Thus the computational effort in the reverse mode is independent of the number of variables m instead of increasing proportionally as in the forward mode. A more detailed analysis of complexity takes into account that the sets K_i contains at most two indices, whereas I_j may contain as many as $(n-j)$ (15).

The method of code transformation applied to the computation in Eq. (39) gives a code list for the gradient in the reverse mode;

$$\begin{aligned}
g_1 &= 1/2\pi && (= \partial s_{10}/\partial s_9) \\
g_2 &= 2s_9 \\
g_3 &= 1/g_2 \\
g_4 &= g_1 g_3 && (= \partial s_{10}/\partial s_8) \\
g_5 &= -g_4 && (= \partial s_{10}/\partial s_7) \\
g_6 &= 2s_6 \\
g_7 &= g_5 g_6 && (= \partial s_{10}/\partial s_6) \\
g_8 &= \mathrm{sqr}(s_4) \\
g_9 &= 1/g_8 \\
g_{10} &= g_4 g_9 \\
g_{11} &= -g_{10} && (= \partial s_{10}/\partial s_4) \\
g_{12} &= s_2 g_{11} && (= \partial s_{10}/\partial s_3 = \partial f/\partial C) \\
g_{13} &= \mathrm{sqr}(s_2) \\
g_{15} &= g_7 g_{14} \\
g_{16} &= -g_{15} \\
g_{17} &= g_{11} s_3 \\
g_{18} &= g_{16} + g_{17} && (= \partial s_{10}/\partial s_2 = \partial f/\partial L) \\
g_{19} &= 1/s_2 \\
g_{20} &= g_7 g_{19} && (= \partial s_{10}/\partial s_1 = \partial f/\partial R)
\end{aligned} \tag{26}$$

where the trivialities $\partial s_{10}/\partial s_{10} = 1$ and $\partial s_{10}/\partial s_5 = \partial s_{10}/\partial s_8$ have been omitted from the computation. Now if desired, reverse mode AD is applied to the code list in Eq. (43) to obtain higher partial derivatives.

CODE LISTS, PROGRAMS, AND COMPUTATIONAL GRAPHS

In early papers on AD, it was simply assumed implicitly that the function of interest is expressed as a composition of elementary operations to which the rules of differentiation are applied. Then the chain rule guarantees that this composite function is correctly differentiated. Explicit formulation of code lists followed a little later (3). Precise definitions were given by Kedem (16), who also extended the idea of AD from code lists to computer subroutines and entire programs.

Technically speaking, a *code list* is a sequence $s = \{s_1, \ldots, s_n\}$ in which each *entry* s_i is (1) an *assignment* of the form $s_i = t$, where the value of t is known, (2) arithmetic operation $s_i = s_j \circ s_k, j, k < i$ involving previous entries, where $\circ$ denotes addition, subtraction, multiplication, or division, or (3) $s_i = \phi(s_j), j < i$, where s_j is a previous entry and the function ϕ is one of a known set of *standard functions* (or *library functions*), such as sine, cosine, and square root, available as subroutines or built into computer hardware. Before the advent of electronic calculators and computers, functions were also evaluated in this way with tabulated or easily computed functions comprising the set of standard functions but without much attention to the actual

process. AD depends on an explicit formulation of the sequence of steps in the evaluation process and, of course, differentiability of the individual operations and standard functions. These steps consist of specifying one or more input variables, followed by the calculating of intermediate variables and finally the output variables, giving the desired function values.

Because computers require exact specification of the sequence of operations to be performed, one of the first advances in computer science was formula translation, that is, conversion of formulas to equivalent code lists for functional evaluation (1). In addition, the advent of computers focused attention on algorithms, that is, step-by-step methods for functional evaluation rather than formulas. For example, the solutions of linear systems of equations are functions of the coefficients of the system matrix and the components of the right-hand side. Cramer's rule gives formulas for these solutions in terms of determinants, but these are essentially useless for actual computation. Instead, linear systems are generally solved by an elimination algorithm (17). If the data of the problem depend on one or more variables, then AD is applied to this process to obtain values of derivatives of the solutions. The same applies to other functions defined by algorithms, as embodied in computer subroutines or entire programs.

In the previous sections, the principles of AD were developed for functions defined by code lists, sometimes called "straight-line code." Generally, computer subroutines and programs contain loops and branches in addition to straight-line code. Although these do not affect AD, in principle, certain practical problems arise [see Ref. 16 and the paper by Fischer (18)].

A loop is simply a set of instructions repeated a fixed or variable number of times. Thus, a loop can be "unrolled" into a segment of straight-line code which is longer than the original by the same factor. If the number of repetitions is fixed, this presents no essential difficulty other than the usual ones of computational time and storage required.

A branch occurs in a computational routine if different sets of instructions are executed under different conditions. For example, the value of $\mathrm{abs}(x) = |x|$ for real x is calculated to be x if $x \geq 0$, or $-x$ otherwise. If a branch occurs, then AD yields the value of the derivative of the function computed by the branch actually taken, provided, of course, that this function is differentiable. For the standard function $\mathrm{abs}(x)$, $\mathrm{abs}'(x) = -1$ for $x < 0$, $\mathrm{abs}'(x) = 1$ for $x > 0$, whereas AD terminates for $x = 0$ if properly implemented.

A useful tool for analyzing computer programs is the *computational graph,* introduced by L. V. Kantorovich (19). For example, Fig. 1 shows diagrammatically how to evaluate the function given by Eq. (29) according to the equivalent code list in Eq. (30). The nodes of this graph indicate the operations to be performed on the input variables. Now the automatic differentiation process is visualized as transformation of the computational graph corresponding to the code transformation described before. This transformation is carried out in forward mode by Eq. (8) or reverse mode by Eq. (15). Because the input variables are conventionally placed at the bottom of the computational graph, the reverse mode is referred to as "top down" whereas the forward mode is "bottom up" in this terminology.

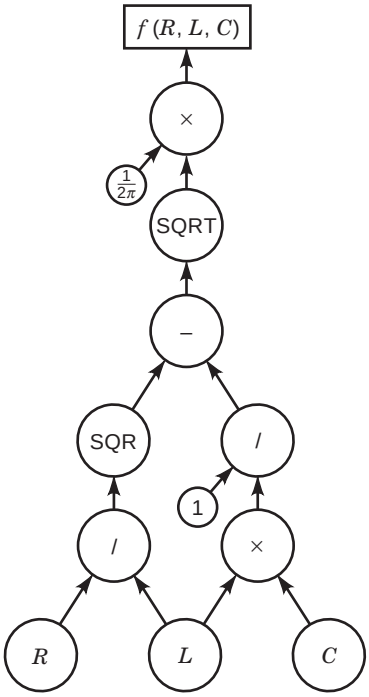


Figure 1. Computational graph of $f(R, L, C)$.

Computational graphs form what are called *directed acyclic graphs* (20). Known results from the theory of these graphs are used to analyze the automatic differentiation process and its complexity (12, 15). To automate the results of graph theory, the nodes of a computational graph are numbered, and the edge from node i to node j is designated by the ordered pair (i, j). Then the operation performed at node i determines the result of differentiation. This leads to a matrix representation of the process of automatic differentiation. See Ref. 2 for a matrix-vector formulation of the forward and reverse modes of gradient computation.

DIFFERENTIATION ARITHMETICS

The process of automatic differentiation has an equivalent formulation as a mathematical system in which the operations yield values of derivatives in addition to values of functions (22). It is evident that a code list such as Eq. (3) can be evaluated if arithmetic operations and standard functions are defined for the quantities involved. For example, complex or interval arithmetic (7) could be used to evaluate Eq. (3) instead of real arithmetic. Instead, consider the set of ordered pairs $U = (u, \dot{u})$ where with addition and subtraction are defined by $U \pm V = (u \pm v, \dot{u} \pm \dot{v})$, respectively, multiplication by $UV = (uv, u\dot{v} + v\dot{u})$, and division by $U/V = \{u/v, [\dot{u} - (u/v)\dot{v}]/v\}$ for $v \neq 0$. In this system, the sine function is defined as $\sin U = (\sin u, \dot{u} \cos u)$. Now if the evaluation of Eq. (3) begins with $s_1 = (t, 1)$ and constants, such as Ω represented by $(\Omega, 0)$, then the result is $s_7 = (I(t), \dot{I}(t))$ which gives the values of the function and its derivative. Here, the rules of differentiation are built into

the definitions of the arithmetic operations and standard functions.

A direct generalization of the previous example is *Taylor arithmetic.* Here, the elements are $(d+1)$-dimensional vectors $U=(u_0, u_1, \ldots, u_d)$ corresponding to the coefficients of a Taylor polynomial of degree d. In this arithmetic, addition and subtraction are defined by Eq. (10), and multiplication and division are given by Eqs. (11) and (10), respectively. Representations of standard functions are derived as previously, for example, $\exp(u_0, \ldots, u_d)=(v_0, \ldots, v_d)$, and $v_0=\exp(u_0)$ and $v_1, \ldots, v_d$ are given by Eq. (15). In this arithmetic, the independent variable is represented by $(t_0, t-t_0, 0, \ldots, 0)$ for Taylor expansion at t_0, and constants, such as Ω, by $(\Omega, 0, \ldots, 0)$. With these as inputs, evaluating Eq. (3) in Taylor arithmetic gives the coefficients of the Taylor polynomial of degree d of $I(t)$ expanded at t_0. More generally, if an arbitrary Taylor polynomial is given as the input variable, then the result of the evaluation process is the corresponding Taylor polynomial of the output.

Differentiation arithmetics are also available for automatically evaluating functions of several variables and their partial derivatives. The simplest is *gradient* arithmetic with elements $(f, \nabla f)$, where $\nabla f=(f_1, \ldots, f_m)$ is an m-dimensional vector. Arithmetic operations in this arithmetic are defined by

$$(f, \nabla f) \pm (g, \nabla g) = (f \pm g, \nabla f \pm \nabla g)$$
$$(f, \nabla f)(g, \nabla g) = (fg, f\nabla g + g\nabla f)$$
$$(f, \nabla f)/(g, \nabla g) = \{f/g, [\nabla f - (f/g)\nabla g]/g\}, g \neq 0$$

as before. If $\phi(x)$ is a differentiable standard function of the single variable x, then the definition of this function in gradient arithmetic is $\phi(f, \nabla f)=[\phi(f), \phi'(f)\ \nabla f]$ by the chain rule. The independent variables $x_1, \ldots, x_m$ are represented by $(x_i, \boldsymbol{e}_i)$, where $\boldsymbol{e}_i$ is the ith unit vector, $i=1, \ldots, m$, and constants c by $(c, \mathbf{0})$ because the gradient of a constant is the zero vector $\mathbf{0}=(0, \ldots, 0)$. Thus evaluating Eq. (30) in gradient arithmetic with the inputs $s_1=[R, (1, 0, 0)]$, $s_2=[L, (0, 1, 0)]$, $s_3=[C, (0, 0, 1)]$ gives the output $(f, \nabla f)$, the values of the function $f=f(R, L, C)$, and its gradient vector. Gradient arithmetic also applies if the input variables are functions of other variables. As long as the values and gradients of the input variables are known, the values and gradients of the output variables are computed correctly by gradient arithmetic.

Straightforward evaluation of a code list, such as Eq. (30) in gradient arithmetic is a forward-mode computation, often less efficient than reverse mode. This comparison applies to serial computation. If a parallel computer with sufficient capacity to compute the components of $(s, \nabla s)$ simultaneously is available, then only one evaluation of the code list in gradient arithmetic is required. When it is simpler to program just the parallel evaluation of ∇s, then two passes through the code list are required, one for the function value and the next for its gradient.

Differentiation arithmetics for higher partial derivatives are constructed according to the same pattern. The $(m \times m)$ symmetrical matrix

$$\boldsymbol{Hf} = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$$

of second partial derivatives is called the *Hessian* of the function $f=f(x_1, \ldots, x_m)$ of m variables. The corresponding *Hessian arithmetic* is based on the definition of arithmetic operations and standard functions for the triples $(f, \nabla f, \boldsymbol{Hf})$ representing the value, gradient vector, and Hessian matrix of a function. For details, see Refs. 2 and 22.

When based on real or complex arithmetic, differentiation arithmetics form a mathematical system called a commutative ring with identity. Performed in interval arithmetic (7), differentiation arithmetics give lower and upper bounds for the Taylor coefficients or partial derivatives to take into account the possibilities of inexact data and round-off error in the computation. Bounds on Taylor coefficients are useful for determining the accuracy of Taylor polynomial approximations to solutions of differential equations and other functions.

SOME APPLICATIONS OF AUTOMATIC DIFFERENTIATION

Automatic differentiation is applicable to the wide variety of computational problems that require evaluation of derivatives. The solution of initial-value problems has been described previously. Other applications of AD to scientific and engineering problems are in the conference proceedings Refs. 23 and 24. Here, brief descriptions of applying AD to solving nonlinear systems of equations, optimization, implicit differentiation, and differentiation of inverse functions are given.

Computational solution of nonlinear equations is usually carried out by iterative algorithms that yield a sequence of improved approximations to solutions, if successful. For a single equation $f(x)=0$, *Newton's method,*

$$x_{n+1} = x_n - [f'(x_n)]^{-1} f(x_n), \quad n = 0, 1, 2, \ldots$$

yields a sequence that converges rapidly to a solution $x=x^*$ of the equation, if the initial approximation x_0 is good enough and $f'(x^*) \neq 0$ (3). This method generalizes immediately to the m-dimensional problem $\boldsymbol{f}(\boldsymbol{x})=\mathbf{0}$, where $\boldsymbol{x}=(x_1, \ldots, x_m)$ and $bf\, f(\boldsymbol{x})=[f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})]$. The derivative of the transformation $\boldsymbol{f}$ is represented by the $(m \times m)$ Jacobian matrix

$$\boldsymbol{f'}(\boldsymbol{x}) = \left[\frac{\partial f_i(\boldsymbol{x})}{\partial x_j}\right]$$

and the Newton step $\Delta \boldsymbol{x}_n = \boldsymbol{x}_{n+1} - \boldsymbol{x}_n$ is obtained by solving the linear system of equations

$$\boldsymbol{f'}(\boldsymbol{x}_n)\Delta \boldsymbol{x}_n = -\boldsymbol{f}(\boldsymbol{x}_n)$$

The rows of the Jacobian matrix $\boldsymbol{f'}(\boldsymbol{x}_n)$are the gradients of the component functions $f_i(\boldsymbol{x}_n)$ and are computable by AD in forward or reverse mode. Conditions for the convergence of Newton's method and bounds for the error $\boldsymbol{x}^* - \boldsymbol{x}_n$ are verified on the basis of evaluation of the Hessians $\boldsymbol{Hf}_i(\boldsymbol{x}_n)$ by AD in interval arithmetic (3). It is also possible to compute Newton steps by solving a large, sparse, linear system of equations based on differentiating the code list for values of the component functions (25).

A simple optimization problem is to find maximum or minimum values of a real function $\phi(x) = \phi(x_1, \ldots, x_m)$ of m variables. If this function is differentiable, then these extremal values are found at one or more of the critical points of the function, which are the solutions of the generally nonlinear system of equations $\nabla\phi(\boldsymbol{x}) = \boldsymbol{0}$. Once the values of the function $\boldsymbol{f}(\boldsymbol{x}) = \nabla\phi(\boldsymbol{x})$ and its Jacobian matrix $\boldsymbol{f}'(\boldsymbol{x}) = H\phi(\boldsymbol{x})$ are obtained by AD, calculating critical points proceeds by Newton's method or some other optimization method based on evaluating the Newton step. Optimization involving constraint functions is handled similarly by AD, after introducing Lagrange multipliers (22).

In addition to functions defined explicitly in terms of the input variables, AD is also applicable to functions defined implicitly. For example, suppose that the relationship $f(x, t) = 0$ defines $x = x(t)$. From the calculation of $\nabla f(x, t)$ by AD, the value of $\ddot{x}(t)$ is given by the usual formula $\ddot{x}(t) = -(\partial f/\partial t)/(\partial f/\partial x)$. Similarly, for relationships of the form $f(x_1, \ldots, x_m) = 0$, the gradient $\nabla f(\boldsymbol{x})$ obtained by AD furnishes the coefficients of linear systems of $(m - 1)$ equations for the various partial derivatives $\partial x_i/\partial x_j$.

A special case of implicit differentiation is the differentiation of inverse functions, which are usually not known explicitly, but are obtained by solving equations. In the case of one variable, this means solving the equation $f(y) = x$ for $y = f^{-1}(x) = g(x)$ by Newton's method or some other iterative procedure (3). The iteration is continued until the solution y is considered satisfactorily accurate according to some criterion giving a stopping condition. In principle, AD can be applied to the iterative procedure to obtain corresponding approximations to derivatives and values of the inverse function. However, a more efficient and likely more accurate method is to obtain the value of $f'(x)$ by AD from the known algorithm for $f(x)$, from which $g'(x) = 1/f'(x)$ gives the derivative of the inverse function. It is interesting to note that the value of $g'(x)$ is obtained in this case without the need to evaluate $g(x)$. This applies also to vector-valued functions of several variables in m dimensions. If $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{f}^{-1}(\boldsymbol{x})$ and if the Jacobian matrix $\boldsymbol{f}'(\boldsymbol{x})$ calculated by AD is invertible, then $\boldsymbol{g}'(\boldsymbol{x}) = [\boldsymbol{f}'(\boldsymbol{x})]^{-1}$.

IMPLEMENTATION OF AUTOMATIC DIFFERENTIATION

Methods for implementing AD are interpretation, code transformation, and operator overloading. Interpretive procedures take the code list for a function as input, analyze each entry, and then use the appropriate subroutine to compute the result. This approach is useful, for example, in interactive programs that accept functions entered from the keyboard of a computer. The corresponding code lists are generated and the desired derivatives computed by interpretation of the code list. Interpretation was also used in early AD programs in which the functions to be differentiated were provided by subroutines (6). In noninteractive programs, interpretation is generally less efficient than code transformation.

Code transformation is generally carried out by precompilation. A program written for function evaluation is analyzed and code for the desired derivatives is inserted at the appropriate places. Then the resulting program is compiled for efficient execution. Current examples of precompilers for AD are PADRE2 (26), ADOL-F (27), and ADIFOR 2.0 (28) for programs written in FORTRAN, and ADOL-C (29) for C programs.

The use of precompilers requires some caution. This is particularly true when dealing with what is called "legacy" code which was written previously by someone else without differentiation in mind. Functions are often approximated very accurately by piecewise rational or highly oscillatory functions, but AD applied to these algorithms can yield nonsensical derivative values. See Refs. 16 and 18 for a discussion of problems which arise in the use of AD.

The idea of *operator overloading* is a natural reflection of a common practice in mathematics. For example, the symbol "+" is used to denote the addition of diverse quantities, such as integers, real or complex numbers, vectors, matrices, functions, and so on. The idea is essentially the same in each instance, but the actual operation to be performed differs. Without thinking much about it, a person uses the meaning of the addition symbol aappropriate to the *type* of addends considered. However, computers are ordinarily built to add only integers or floating-point numbers, and early computer languages reflected this limitation of the meaning of + and the types of addends permitted. Later, languages, such as C++ (30), were developed which allow extending the meaning of operator symbols to types of operands defined by the programmer. This is called "overloading" the symbol. The overloaded operations and functions are carried out by appropriate subroutines. The compiler checks types of operands and constructs calls to these subroutines. See Refs. 31 and 32 for examples of automating differentiation arithmetics by operator overloading. In C++, AD is implemented by "class libraries" containing the appropriate definitions, operators, and functions for various differentiation arithmetics (30).

Use of operator overloading to implement AD simplifies programming because functions and subroutines are written in the usual way and the compiled program produces derivative and functional values. Here, differentiation is done at compile time because the compiler generates the sequences of calls to subroutines for evaluating functions and their derivatives. The price for this convenience is that the final computation is carried out in forward mode with the corresponding possible loss of efficiency. As mentioned before, this may not be a drawback in parallel computation.

ALGORITHMIC DIFFERENCING

The algorithmic method also facilitates accurate computation of divided differences

$$f[x + h, x] = \frac{f(x + h) - f(x)}{h},$$

see Refs. 33 and 34. Direct computation of $f[x + h, x]$ by subtraction followed by division is problematical in finite-precision arithmetic as h approaches zero due to the fact that $f(x + h)$ and $f(x)$ will agree to more significant digits, and the difference will eventually consist of only round-off error which is then multiplied by the large number

$1/h$. This difficulty is avoided in algorithmic differencing (AΔ) by the use of expressions for divided differences of the arithmetic operations which are numerically stable for $|h|$ small, and approach the values of their derivatives as $h \to 0$ as required by mathematical theory. The postfix operator $[x + h, x]$ is defined for differentiable functions $f(x)$ by

$$f[x+h,x] = \begin{cases} (f(x+h) - f(x))/h & \text{if } h \neq 0, \\ f'(x) & \text{if } h = 0. \end{cases} \tag{27}$$

For composite functions $(f \circ g)(x) = f(g(x))$, the *chain rule*

$$(f \circ g)[x+h,x] = f[g(x+h), g(x)] \cdot g[x+h,x] \tag{28}$$

holds as for derivatives. This guarantees that starting the algorithm with the divided difference of the input (for example, $x[x + h, x] = 1$) will yield the divided difference of the output. The AΔ rules for arithmetic operations and intrinsic functions are obtained as in elementary calculus as expressions which give derivatives as $h \to 0$. For example, the divided-difference expression for the quotient is

$$(f/g)[x+h,x] = (f[x,x+h] - (f/g)g[x,x+h])/(g+h),$$

which gives the derivative formula $(f/g)' = (f' - (f/g)g')/g$ for $h = 0$. Similarly, if $f(x) = \arctan g(x)$, one has

$$f[x,x+h] = \frac{1}{h}\arctan\left(\frac{hg[x,x+h]}{1+g(u+g[x,x+h])}\right),$$

for $h \neq 0$, and the derivative

$$f[x,x] = f'(x) = \frac{u'(x)}{1+g^2(x)}$$

for $h = 0$, and so on.

The arithmetic operations and standard functions for AD are the special cases of those for AΔ with $h = 0$. Thus, a computer program to implement AΔ can be used to provide values of derivatives, divided differences (or differences $f(x+h) - f(x) = h\, f[x+h,x]$) of equivalent accuracy for a wide range of values of h.

BIBLIOGRAPHY

1. A. V. Aho, R. Sethi, J. D. Ullman, *Compilers: Principles, Techniques, and Tools*, Reading, MA: Addison-Wesley, 1988.
2. L. B. Rall, G. F. Corliss, Introduction to automatic differentiation, inM. Berz *et al.* (eds.), *Computational Differentiation, Techniques, Applications, and Tools*, Philadelphia: SIAM, 1996.
3. L. B. Rall, *Computational Solution of Nonlinear Operator Equations*, New York: Wiley, 1969. Reprint, Huntington, NY: Krieger, 1979.
4. R. E. Moore, *Interval Arithmetic and Automatic Error Analysis in Digital Computing*, Ph.D. Thesis, Stanford University, 1962.
5. R. E. Wengert, A simple automatic derivative evaluation program, *Commun. ACM*, **7** (8): 463–464, 1964.
6. L. B. Rall, *Automatic Differentiation: Techniques and Applications*, New York: Springer, 1981.
7. R. E. Moore, *Methods and Applications of Interval Analysis*, Philadelphia: SIAM, 1979.
8. Y. F. Chang, G. F. Corliss, Solving ordinary differential equations using Taylor series, *ACM Trans. Math. Softw.*, **8**: 114–144, 1982.
9. Y. F. Chang, G. F. Corliss, ATOMFT: Solving ODE's and DAE's using Taylor series, *Comput. Math. Appl.*, **28**: 209–233, 1994.
10. R. E. Moore, Automatic local coordinate transformations to reduce the growth of error bounds in interval computation of solutions of ordinary differential equations, inL. B. Rall (ed.), *Error in Digital Computation*, Vol. 2, New York: Wiley, 1965.
11. R J. Lohner, Enclosing the solutions of ordinary initial and boundary value problems, inE. W. Kaucher, U. W. Kulisch, andC. Ullrich (eds.), *Computer Arithmetic: Scientific Computation and Programming Languages*, Stuttgart: Wiley-Teubner, 1987.
12. M. Iri, Simultaneous computation of function, partial derivatives and estimates of rounding errors—Complexity and practicality, *Jpn. J. Appl. Math.*, **1**: 223–252, 1984.
13. S. Linnainmaa, Taylor expansion of the accumulated rounding error, *BIT*, **16**: 146–160, 1976.
14. B. Speelpennig, *Computing Fast Partial Derivatives of Functions Given by Algorithms*, Ph.D. Thesis, University of Illinois, 1980.
15. A. Griewank, Some bounds on the complexity of gradients, Jacobians, and Hessians, inP. M. Pardalos (ed.), *Complexity in Nonlinear Optimization*, Singapore: World Scientific, 1993.
16. G. Kedem, Automatic differentiation of computer programs, *ACM Trans. Math. Softw.*, **6**: 150–165, 1980.
17. G. Forsythe, C. B. Moler, *Computer Solutions of Linear Algebraic Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1967.
18. H. Fischer, Special problems in automatic differentiation, inA. Griewank andG. F. Corliss (eds.), *Automatic Differentiation of Algorithms, Theory, Implementation, and Application*, Philadelphia: SIAM, 1992.
19. L. V. Kantorovich, On a mathematical symbolism convenient for performing mathematical calculations, Russian, *Dokl. Akad. Nauk USSR*, **113**: 738–741, 1957.
20. C. W. Marshall, *Applied Graph Theory*, New York: Wiley, 1971.
21. L. B. Rall, Gradient computation by matrix multiplication, inH. Fischer, B. Riedmüller, andS. Schäffler (eds.), *Applied Mathematics and Parallel Computing*, Heidelberg: Physica-Verlag, 1996.
22. L. B. Rall, Differentiation arithmetics, inC. Ullrich (ed.), *Computer Arithmetic and Self-Validating Numerical Methods*, New York: Academic Press, 1990.
23. A. Griewank, G. F. Corliss (eds.), *Automatic Differentiation of Algorithms, Theory, Implementation, and Applications*, Philadelphia: SIAM, 1992.
24. M. Berz *et al.* (eds.), *Computational Differentiation, Techniques, Applications, and Tools*, Philadelphia: SIAM, 1996.
25. A. Griewank, Direct calculation of Newton steps without accumulating Jacobians, inT. F. Coleman andY. Li (eds.), *Large-Scale Numerical Optimization*, Philadelphia: SIAM, 1990.
26. K. Kubota, PADRE2—Fortran precompiler for automatic differentiation and estimates of rounding errors, inM. Berz *et al.* (eds.), *Computational Differentiation, Techniques, Applications, and Tools*, Philadelphia: SIAM, 1996.
27. D. Shiriaev, A. Griewank, ADOL-F: Automatic differentiation of Fortran codes, inM. Berz *et al.* (eds.), *Computational Differentiation, Techniques, Applications, and Tools*, Philadelphia: SIAM, 1996.

28. C. Bischof, A. Carle, Users' experience with ADIFOR 2.0, inM. Berz *et al.* (eds.), *Computational Differentiation, Techniques, Applications, and Tools*, Philadelphia: SIAM, 1996.
29. D. W. Juedes, A taxonomy of automatic differentiation tools, inA. Griewank andG. F. Corliss (eds.), *Automatic Differentiation of Algorithms, Theory, Implementation, and Applications*, Philadelphia: SIAM, 1991.
30. B. Stroustrup, *The C++ Programming Language*, Reading, MA: Addison-Wesley, 1987.
31. L. B. Rall, Differentiation and generation of Taylor coefficients in Pascal-SC, inU. W. Kulisch andW. L. Miranker (eds.), *A New Approach to Scientific Computation*, New York: Academic Press, 1983.
32. L. B. Rall, Differentiation in Pascal-SC, type GRADIENT,*ACM Trans. Math. Softw.*, **10**: 161–184, 1984.
33. L. B. Rall and T. W. Reps, Algorithmic differencing, In U. Kulisch, R. Lohner, and A. Facius (eds.), *Perspectives on Enclosure Methods*, Springer-Verlag, Vienna, 2001.
34. T. W. Reps and L. B. Rall, Computational divided differencing and divided-difference arithmetics, *Higher-Order and Symbolic Computation*, **16**, 93–149, 2003.

LOUIS B. RALL
GEORGE F. CORLISS
Department of Mathematics
University of
Wisconsin-Madison, 480
Lincoln Drive, Madison, WI
Department of Electrical and
Computer Engineering
Marquette University,
Milwaukee, WI

Bessel Functions
Standard Article
Frank B. Gross[1]
[1]Florida State University, Tallahassee, FL
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2402
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (165K)

Abstract

The sections in this article are

Mathematical Foundation and Background of Bessel Functions

Derivation of a New Bessel Function Approximation

Copyright © 1999-2008 All Rights Reserved.

BESSEL FUNCTIONS

Friedrich Wilhelm Bessel led a fascinating and productive life (1–4). Bessel was born on July 22, 1784, in Minden, Westphalia, and died of cancer on March 17, 1846, in Königsberg, Prussia. His father was a civil servant, and his mother was the daughter of a minister. Bessel had two brothers and six sisters.

He began his education at the Gymnasium in Minden but left at the age of 14 after having difficulty learning Latin. His brothers went on to receive University degrees, finding careers as judges in high courts. Bessel became an apprentice in an import–export business. He independently studied textbooks, educating himself in Latin, Spanish, English, geography, navigation, astronomy, and mathematics.

In 1804 Bessel wrote a paper calculating the orbit of Halley's comet. His paper so impressed the comet expert Heinrich Olbers, that Olbers encouraged him to continue the work and become a professional astronomer. In 1806 Bessel obtained a position in the Lilienthal observatory near Bremen. In 1809 he was appointed the Director and Professor of Astronomy at the observatory in Königsberg. Commensurate with the position, he was awarded an honorary degree by Karl F. Gauss at the University of Göttingen.

In 1811 Bessel was awarded the Lalande Prize from the Institute of France for his refraction tables. In 1815 he was awarded a prize by the Berlin Academy of Sciences for his work in determining precession from proper star motions. Also, in 1825, he was elected as a Fellow of the Royal Society of London.

Perhaps his most famous accomplishment was solving a three-century dream of astronomers—the determination of the parallax of a star. However, of special interest to engineers, physicists, and mathematicians was the development of the Bessel function. His functions were derived in the study of the movement and perturbation of bodies under mutual gravitation. In 1824 his functions were used in a treatise on elliptic planetary motion.

The Bessel functions are frequently found in problems involving circular cylindrical boundaries. They arise in such fields as electromagnetics, elasticity, fluid flow, acoustics, and communications.

MATHEMATICAL FOUNDATION AND BACKGROUND OF BESSEL FUNCTIONS

Bessel's differential equation has roots in an elementary transformation of Riccati's equation (5). Three earlier mathematicians studied special cases of Bessel's equation (6). In 1732 Daniel Bernoulli studied the problem of a suspended heavy flexible chain. He obtained a differential equation that can be transformed into the same form as that used by Bessel. In 1764 Leonhard Euler studied the vibration of a stretched circular membrane and derived a differential equation essentially the same as Bessel's equation. In 1770 Joseph-Louis Lagrange derived an infinite series solution to the problem of the elliptic motion of a planet. His series coefficients are related to Bessel's later solution to the same problem. Various particular cases were solved by Bernoulli, Euler, and Lagrange, but it was Bessel who arrived at a systematic solution and the subsequent Bessel functions.

Bessel functions arise as a solution to the following differential equation

$$x\frac{d}{dx}\left(x\frac{dy}{dx}\right) + (x^2 - \nu^2)y = 0 \tag{1}$$

Equation (1), by applying the chain rule, may also be expressed as

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 - \nu^2)y = 0 \tag{2}$$

Solutions to the preceding differential equation can be in one of three forms: Bessel functions of the first, second, or third kind and order ν. Bessel functions (7) of the first kind are denoted $J_{\pm\nu}(x)$. Bessel functions of the second kind are called Weber (Heinrich Weber) or Neumann (Carl Neumann) functions and are denoted $Y_\nu(x)$. They are sometimes also labeled $N_\nu(x)$. Bessel functions of the third kind are denoted $H_\nu^{(1)}(x)$, $H_\nu^{(2)}(x)$ and are called Hankel functions (Hermann Hankel). $H_\nu^{(1)}(x)$ is called the Hankel function of the first kind and order ν where $H_\nu^{(2)}(x)$ is the Hankel function of the second kind. Bessel functions of the second and third kind are linear functions of the Bessel function of the first kind. A variation of Eq. (2) will yield what is called the modified Bessel functions of the first $[I_\nu(x)]$ and second $[K_\nu(x)]$kind. The modified Bessel functions will be discussed later.

The solution of Eq. (2) for noninteger orders can be expressed as a linear combination of Bessel functions of the first kind with positive and negative orders. It is given as

$$y(x) = AJ_\nu(x) + BJ_{-\nu}(x) \tag{3}$$

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

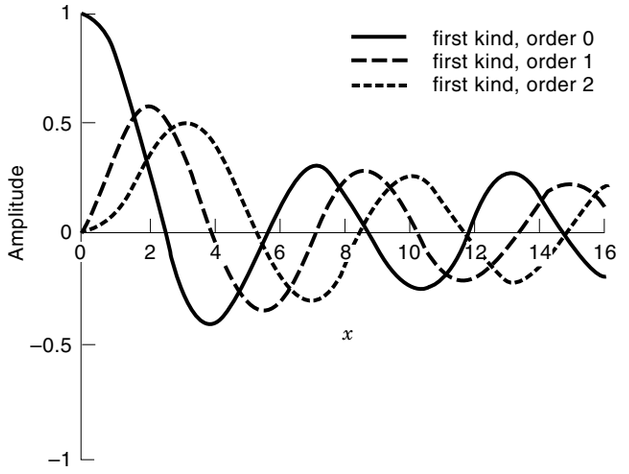


Figure 1. Plot of three typical Bessel functions of the first kind, orders 0, 1, and 2.

where A and B are arbitrary constants to be found by applying boundary conditions. Figure 1 shows a plot of Bessel functions of the first kind of orders 0, 1, and 2.

By allowing $A = \text{cotan}(\nu\pi)$ and $B = -\csc(\nu\pi)$ and assuming ν is a noninteger, we arrive at the Weber function solution to Bessel's differential equation

$$Y_\nu(x) = \frac{J_\nu(x)\cos(\nu\pi) - J_{-\nu}(x)}{\sin(\nu\pi)} \tag{4}$$

Figure 2 shows a plot of the Weber function of orders 0, 1, and 2.

We can also find a linear combination of the Bessel functions of the first and second kind to derive the complex Hankel functions. The Hankel function of the first kind is given as

$$\begin{aligned} H_\nu^{(1)}(x) &= J_\nu(x) + jY_\nu(x) \\ &= j\csc(\nu\pi)[e^{-j\nu\pi}J_\nu(x) - J_{-\nu}(x)] \end{aligned} \tag{5}$$

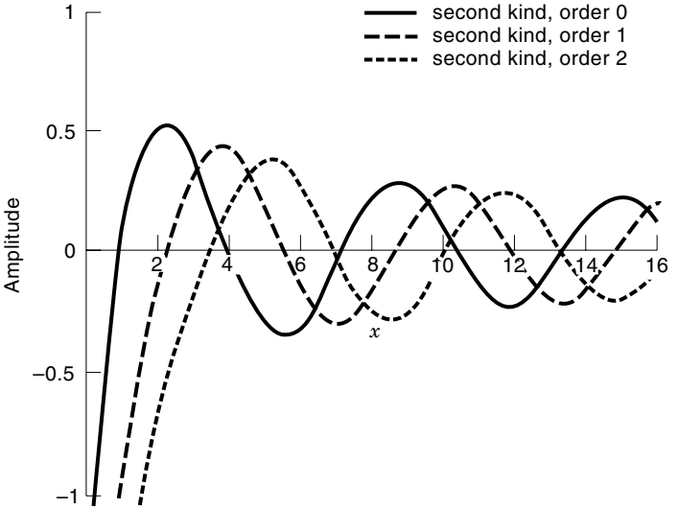


Figure 2. Plot of three typical Weber functions, orders 0, 1, and 2.

The Hankel function of the second kind is given as

$$\begin{aligned} H_\nu^{(2)}(x) &= J_\nu(x) + jY_\nu(x) \\ &= j\csc(\nu\pi)[J_{-\nu}(x) - e^{j\nu\pi}J_\nu(x)] \end{aligned} \tag{6}$$

It can be seen that $H_\nu^{(2)}(x)$ is the conjugate of $H_\nu^{(1)}(x)$.

Ascending Series Solution

Bessel's equation, Eq. (2), can be solved by using the method of Frobenius (Georg Frobenius). The method of Frobenius is the attempt to find nontrivial solutions to Eq. (2), which take the form of an infinite power series in x multiplied by x to some power ν. As a consequence, the Bessel functions of the first kind and order ν can then be expressed as

$$\begin{aligned} J_\nu(x) &= \sum_{m=0}^{\infty}(-1)^m \frac{1}{m!(m+\nu)!}\left(\frac{x}{2}\right)^{2m+\nu} \\ &= \sum_{m=0}^{\infty}(-1)^m \frac{1}{m!\Gamma(m+\nu+1)}\left(\frac{x}{2}\right)^{2m+\nu} \end{aligned} \tag{7}$$

where the Gamma function $\Gamma(m + \nu + 1)$ has been used to replace the factorial $(m + \nu)!$.

The series terms in Eq. (7) have alternating signs. Looking at the first few terms in the series we have

$$\begin{aligned} J_\nu(x) = \frac{1}{\nu!}\left(\frac{x}{2}\right)^\nu \Bigg[1 &- \left(\frac{x}{2}\right)^2 \frac{1}{(1+\nu)} \\ &+ \left(\frac{x}{2}\right)^4 \frac{1}{2(2+\nu)(1+\nu)} - \cdots\Bigg] \end{aligned} \tag{8}$$

In the case of $\nu = 0$, we have the ascending series for the Bessel function of the first kind and order 0 given as

$$J_0(x) = 1 - \left(\frac{x}{2}\right)^2 + \frac{1}{(2!)^2}\left(\frac{x}{2}\right)^4 - \frac{1}{(3!)^2}\left(\frac{x}{2}\right)^6 + \cdots \tag{9}$$

To obtain the series solution for Bessel functions of negative order, merely substitute $-\nu$ for ν to get

$$J_{-\nu}(x) = \sum_{m=0}^{\infty}(-1)^m \frac{1}{m!\Gamma(m-\nu+1)}\left(\frac{x}{2}\right)^{2m-\nu} \tag{10}$$

If ν is an integer n, then the pair of Bessel functions for positive and negative orders is

$$J_n(x) = \sum_{m=0}^{\infty}(-1)^m \frac{1}{m!\Gamma(m+n+1)}\left(\frac{x}{2}\right)^{2m+n} \tag{11}$$

$$J_{-n}(x) = \sum_{m=0}^{\infty}(-1)^m \frac{1}{m!\Gamma(m-n+1)}\left(\frac{x}{2}\right)^{2m-n} \tag{12}$$

However, in the negative order series [Eq. (12)] the Gamma function $\Gamma(m - n + 1)$ is ∞ when $m < n$. In this case, all the terms in the series are zero for $m < n$. The series can then be rewritten as

$$J_{-n}(x) = \sum_{m=n}^{\infty}(-1)^m \frac{1}{m!\Gamma(m-n+1)}\left(\frac{x}{2}\right)^{2m-n} \tag{13}$$

By letting $m' = m - n$ the series can be reexpressed as

$$J_{-n}(x) = \sum_{m'=0}^{\infty} (-1)^{m'+n} \frac{1}{m'!\Gamma(m'+n+1)} \left(\frac{x}{2}\right)^{2m'+n} \tag{14}$$

In the integer order case by comparing Eqs. (11) and (14), it can be shown that

$$J_{-n}(x) = (-1)^n J_n(x) \tag{15}$$

The same procedure can be performed for the Weber function to show that

$$Y_{-n}(x) = (-1)^n Y_n(x) \tag{16}$$

Integral Solutions

The Bessel function solution can not only be defined in terms of the ascending power series above, but it also can be expressed in several integral forms. An extensive list of integral forms can be found in Refs. 7 and 8. When the order ν is not necessarily an integer, then the Bessel function can be expressed as Poisson's integral

$$J_\nu(x) = \left(\frac{x}{2}\right)^\nu \frac{1}{\sqrt{\pi}\Gamma(\nu+1/2)} \int_0^\pi \cos(z\cos\beta)\sin^{2\nu}(\beta)\,d\beta \tag{17}$$

Equation (17) is valid when letting $\cos\beta = z$, Eq. (17) can be written as

$$J_\nu(x) = 2\left(\frac{x}{2}\right)^\nu \frac{1}{\sqrt{\pi}\Gamma(\nu+1/2)} \int_0^1 (1-z^2)^{\nu-1/2}\cos(zx)\,dz \tag{18}$$

In the case where ν is an integer, several more integral representations can used. For $\nu = 2m$ = even

$$J_{2m}(x) = \frac{2}{\pi}\int_0^\pi \cos(x\sin\beta)\cos(2m\beta)\,d\beta \qquad m > 0 \tag{19}$$

and for $\nu = 2m + 1$ = odd

$$J_{2m+1}(x) = \frac{2}{\pi}\int_0^\pi \sin(x\sin\beta)\sin[(2m+1)\beta]\,d\beta \qquad m > 0 \tag{20}$$

For an arbitrary integer n

$$J_n(x) = \frac{1}{\pi}\int_0^\pi \cos(x\sin\beta - n\beta)\,d\beta \tag{21}$$

Integrals Involving Bessel Functions

Many useful integrals involving Bessel functions may be found in Refs. 8 and 9. Several indefinite integrals follow. (A constant of integration may be necessary under certain conditions.)

$$\int J_\nu(x)\,dx = 2\sum_k J_{\nu+2k+1}(x) \tag{22}$$

$$\int x^{\nu+1}J_\nu(\alpha x)\,dx = \frac{1}{\alpha}x^{\nu+1}J_{\nu+1}(\alpha x) \tag{23}$$

$$\int x^{1-\nu}J_\nu(\alpha x)\,dx = -\frac{1}{\alpha}x^{1-\nu}J_{\nu-1}(\alpha x) \tag{24}$$

$$\int x^m J_n(x)\,dx = x^m J_{n+1}(x) - (m-n-1)\int x^{m-1}J_{n+1}(x)\,dx \tag{25}$$

$$\int x^m J_n(x)\,dx = -x^m J_{n-1}(x) + (m+n-1)\int x^{m-1}J_{n-1}(x)\,dx \tag{26}$$

Several definite integrals involving Bessel functions are given as

$$\int_0^\infty J_\nu(\alpha x)\,dx = \frac{1}{\alpha} \qquad [\mathrm{Re}\nu > -1,\ \alpha > 0] \tag{27}$$

$$\int_0^a J_0(x)\,dx = aJ_0(a) + \frac{\pi a}{2}[J_1(a)\mathbf{H}_0(a) - J_0(a)\mathbf{H}_1(a)] \qquad [a > 0] \tag{28}$$

$$\int_a^\infty J_0(x)\,dx = 1 - aJ_0(a) + \frac{\pi a}{2}[J_0(a)\mathbf{H}_1(a) - J_1(a)\mathbf{H}_0(a)][a > 0] \tag{29}$$

where $\mathbf{H}_0(a)$ and $\mathbf{H}_1(a)$ are the Struve functions defined by

$$\mathbf{H}_\nu(a) = \sum_{m=0}^\infty (-1)^m \frac{\left(\frac{a}{2}\right)^{2m+\nu+1}}{\Gamma(m+3/2)\Gamma(\nu+m+3/2)} \tag{30}$$

$$\int_0^a J_1(x)\,dx = 1 - J_0(a) \qquad [a > 0] \tag{31}$$

$$\int_a^\infty J_1(x)\,dx = J_0(a) \qquad [a > 0] \tag{32}$$

$$\left.\begin{aligned} \int_0^\infty J_\nu(\alpha x)J_{\nu-1}(\beta x)\,dx &= \frac{\beta^{\nu-1}}{\alpha^\nu} \quad [\beta < \alpha] \\ &= \frac{1}{2\beta} \quad [\beta = \alpha] \\ &= 0 \quad [\beta > \alpha] \end{aligned}\right\} \qquad [\mathrm{Re}\nu > 0] \tag{33}$$

$$\int_0^a J_\nu(x)J_{\nu+1}(x)\,dx = \sum_{n=0}^\infty [J_{\nu+n+1}(a)]^2 \quad [\mathrm{Re}(\nu) > -1] \tag{34}$$

Recursion Relationships for $J_n(x)$ and $Y_n(x)$

By taking a derivative with respect to x (see Ref. 6 or 7) of Eq. (11), it can be shown that

$$xJ_n'(x) = nJ_n(x) + xJ_{n+1}(x) \tag{35}$$

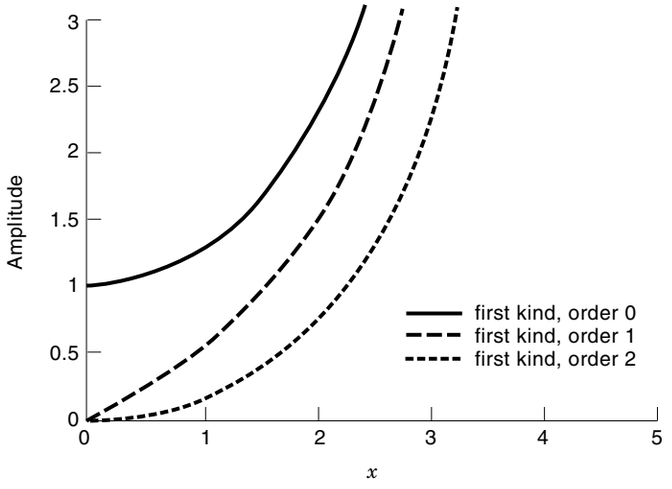


Figure 3. Modified Bessel functions of the first kind, orders 0, 1, and 2.

In a similar manner, we can also find that

$$xJ_n'(x) = nJ_n(x) - xJ_{n-1}(x) \tag{36}$$

By adding Eqs. (35) and (36) and normalizing by x, we can find that

$$J_n'(x) = \frac{1}{2}[J_{n-1}(x) - J_{n+1}(x)] \tag{37}$$

By subtracting Eq. (36) from Eq. (35), we get a recursion relationship for the Bessel function of the first kind and order $n + 1$ based upon orders n and $n - 1$.

$$J_{n+1}(x) = \frac{2n}{x} J_n(x) - J_{n-1}(x) \tag{38}$$

Modified Bessel Functions

A similar differential equation to that given in Eq. (2) can be derived by replacing x with the imaginary variable ix. We arrive at a variation on Bessel's differential equation given as

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} - (x^2 + \nu^2)y = 0 \tag{39}$$

Equation (39) differs from Eq. (2) only in the sign of x^2 in parenthesis. The solution of Eq. (39) is defined as the modified Bessel function of the first kind and is given as

$$I_\nu(x) = \sum_{m=0}^{\infty} \frac{(x/2)^{2m+\nu}}{m!(m+\nu)!} = \sum_{m=0}^{\infty} \frac{(x/2)^{2m+\nu}}{m!\Gamma(m+\nu+1)} = i^{-\nu}J_\nu(ix) \tag{40}$$

The symbol $I_\nu(x)$ was chosen because $I_\nu(x)$ in Eq. (40) is related to the Bessel function $J_\nu(ix)$, which has an "imaginary" argument. One may note that the terms in the series of Eq. (40) are all positive, whereas the terms in Eq. (7) have alternating signs. The solution $I_{-\nu}(x)$ for $-\nu$ is linearly independent of $I_\nu(x)$ except when ν is an integer. When ν is equal to the integer n then $I_{-n}(x) = I_n(x)$. Figure 3 shows a graph of the modified Bessel functions of the first kind and orders 0, 1, and 2. We can define a second valid solution to Eq. (39) as a linear combination of the modified Bessel function of the first kind. The second solution is referred to as the modified Bessel function of the second kind $K_\nu(x)$ and is given as

$$K_\nu(x) = \frac{\pi}{2}\frac{I_{-\nu}(x) - I_\nu(x)}{\sin(\nu\pi)} \tag{41}$$

In the case where ν equals an integer n, the modified Bessel function $K_n(x)$ is found by

$$K_n(x) = \lim_{\nu \to n} K_\nu(x) \tag{42}$$

Figure 4 shows a graph of the modified Bessel functions of the second kind and orders 0, 1, and 2.

Integral Form of the Modified Bessel Function of the First Kind

In addition to the ascending series expression of Eq. (40), the modified Bessel function may be expressed in integral form. Several integral forms follow and are valid for $\mathrm{Re}(\nu) > 1/2$.

$$\begin{aligned} I_\nu(x) &= \frac{\left(\frac{x}{2}\right)^\nu}{\Gamma(\nu+1/2)\Gamma(1/2)} \int_0^\pi \cosh(x\cos\beta)\sin^{2\nu}(\beta)\,d\beta \\ &= \frac{\left(\frac{x}{2}\right)^\nu}{\Gamma(\nu+1/2)\Gamma(1/2)} \int_0^\pi e^{\pm x\cos\beta}\sin^{2\nu}(\beta)\,d\beta \\ &= \frac{\left(\frac{x}{2}\right)^\nu}{\Gamma(\nu+1/2)\Gamma(1/2)} \int_{-1}^1 (1-y^2)^{\nu-1/2}\cosh(xy)\,dy \end{aligned} \tag{43}$$

Approximations to Bessel Functions

The small argument approximation for the Bessel function is (6)

$$J_n(x) \approx \frac{1}{\Gamma(n+1)}\left(\frac{x}{2}\right)^n = \frac{1}{n!}\left(\frac{x}{2}\right)^n; \quad x \to 0 \tag{44}$$

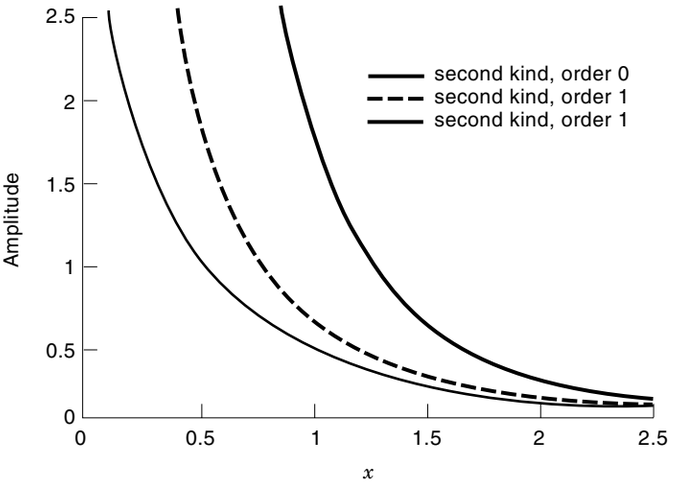


Figure 4. Modified Bessel functions of the second kind, orders 0, 1, and 2.

In the case of the $J_0(x)$ and the $J_1(x)$ Bessel functions,

$$J_0(x) = 1; \qquad J_1(x) = \frac{x}{2} \tag{45}$$

The large argument approximation is given as

$$J_\alpha(x) \approx \left(\frac{2}{\pi x}\right)^{1/2} \cos(x - \alpha\pi/2 - \pi/4); \quad x \to \infty \tag{46}$$

In the case of the $J_0(x)$ and the $J_1(x)$ Bessel functions we have

$$\begin{aligned} J_0(x) &= \left(\frac{2}{\pi x}\right)^{1/2} \cos(x - \pi/4); \\ J_1(x) &= \left(\frac{2}{\pi x}\right)^{1/2} \cos(x - 3\pi/4) \end{aligned} \tag{47}$$

The small argument approximation is only reasonably accurate, for orders 0 and 1, when $x < 0.5$. The large argument approximation, for orders 0 and 1, is only reasonably accurate for $x > 2.5$.

A 12th-order polynomial approximation is available in Abramowitz and Stegun (7), which is valid for $|x| \leq 3$. In the case of the 0th-order Bessel function, the approximation is given as

$$\begin{aligned} J_0(x) = 1 &- 2.2499997\left(\frac{x}{3}\right)^2 + 1.2656208\left(\frac{x}{3}\right)^4 \\ &- 0.3163866\left(\frac{x}{3}\right)^6 + 0.0444479\left(\frac{x}{3}\right)^8 \\ &- 0.0039444\left(\frac{x}{3}\right)^{10} + 0.0002100\left(\frac{x}{3}\right)^{12} + \epsilon \\ &|\epsilon| < 5 \times 10^{-8} \end{aligned} \tag{48}$$

In the case of the first-order Bessel function the approximation is given as

$$\begin{aligned} \frac{1}{x} J_1(x) = 0.5 &- 0.56249985\left(\frac{x}{3}\right)^2 + 0.21093573\left(\frac{x}{3}\right)^4 \\ &- 0.03954289\left(\frac{x}{3}\right)^6 + 0.00443319\left(\frac{x}{3}\right)^8 \\ &- 0.00031761\left(\frac{x}{3}\right)^{10} + 0.00001109\left(\frac{x}{3}\right)^{12} + \epsilon \\ &|\epsilon| < 1.3 \times 10^{-8} \end{aligned} \tag{49}$$

The polynomial requires an eight decimal place accuracy in the coefficients. Several other polynomial and rational approximations can be found in Luke (10–12).

A new approximating function can be developed that is simpler than Eqs. (48) and (49) and useful over the range $|x| \leq 5$. This function is adequate to replace the small argument approximation and bridges the gap to the large argument approximation of Eq. (46).

DERIVATION OF A NEW BESSEL FUNCTION APPROXIMATION

In studying the general problem of TM scattering from conducting strip gratings by using conformal mapping methods an integral was discovered with no previously known solution (13). The new integral is

$$f_{2n}(k) = \frac{2}{\pi} \int_0^\delta \frac{\cos(x)}{\sqrt{k^2 - \sin^2(x)}} \cos(2nx)\, dx \tag{50}$$

where

$$k = \sin\delta; \quad 0 \leq \delta \leq \pi/2$$

[Note the similarities between Eqs. (50) and (17).] Equation (50) can be reduced to a form identical to Eq. (17) by allowing δ to be vanishingly small. This application will be made after the function $f_{2n}(k)$ has been evaluated.

Several steps are undertaken in finding the solution for Eq. (50). By letting $\sin x = k \sin\alpha$, Eq. (50) can be reduced to the form

$$f_{2n}(k) = \frac{2}{\pi} \int_0^{\pi/2} \cos\{2n \arcsin[k \sin(\alpha)]\}\, d\alpha \tag{51}$$

Since $\arcsin(k \sin\alpha) = \pi/2 - \arccos(k \sin\alpha)$, $\cos(n\pi + \phi) = (-1)^n \cos(\phi)$, and using the definition of a Chebyshev polynomial (Eq. 22.3.15 in Ref. 7) we get

$$f_{2n}(k) = \frac{2}{\pi} \int_0^{\pi/2} (-1)^n T_{2n}(k \sin\alpha)\, d\alpha \tag{52}$$

with

$$T_{2n}(k \sin\alpha) = \text{Chebyshev polynomial of order } 2n$$

The Chebyshev polynomial of order $2n$ can be expressed as a finite sum (8) and is alternatively defined as

$$T_{2n}(x) = n \sum_{m=0}^{n} (-1)^m \frac{(2n - m - 1)!}{m!(2n - 2m)!} (2x)^{2n-2m} \tag{53}$$

by substituting Eq. (53) into Eq. (52), we get

$$\begin{aligned} f_{2n}(k) = \frac{2}{\pi}(-1)^n \Bigg[n \sum_{m=0}^{n} (-1)^m \frac{(2n - m - 1)!}{m!(2n - 2m)!} (2k)^{2n-2m} \\ \times \int_0^{\pi/2} (\sin\alpha)^{2n-2m}\, d\alpha \Bigg] \end{aligned} \tag{54}$$

However, the integral imbedded in Eq. (54) is $\pi/2$ when $2n = 2m$ and in general is given by

$$\int_0^{\pi/2} (\sin\alpha)^{2n-2m}\, d\alpha = \frac{\pi}{2} \frac{(2n - 2m - 1)!!}{(2n - 2m)!!}, \qquad 2n > 2m \tag{55}$$

where

$$(2n - 2m - 1)!! = 1 \cdot 3 \cdot 5 \cdots (2n - 2m - 1)$$

and

$$(2n - 2m)!! = 2 \cdot 4 \cdot 6 \cdots (2n - 2m)$$

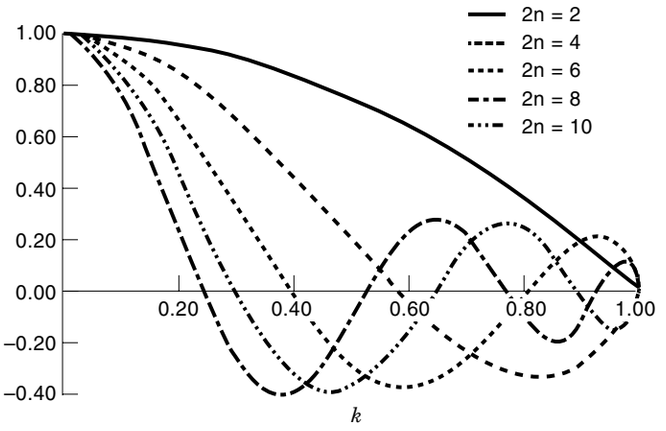


Figure 5. Plot of Bessel approximating function $f_{2n}(k)$.

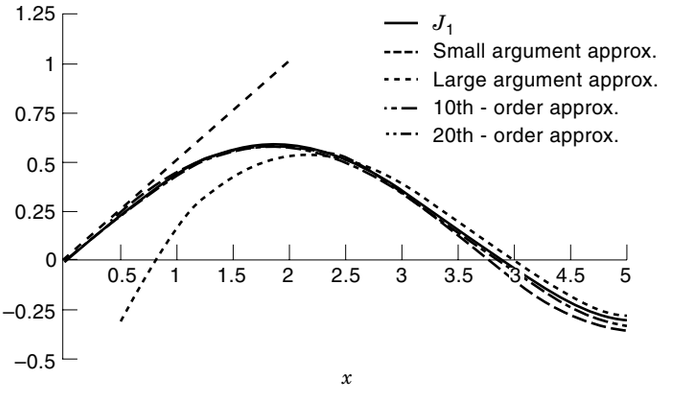


Figure 7. Comparison among $J_1(x)$, classic approximations, and the new approximations.

Upon substitution of Eq. (55) into Eq. (54) the final solution is given as

$$f_{2n}(k) = \sum_{m=0}^{n} b_m k^{2n-2m} \tag{56}$$

where the coefficients to the series are given as

$$b_m = n(-1)^{m+n} \frac{(2n-m-1)!2^{2n-2m}}{m!((2n-2m)!!)^2}$$

The solution in Eq. (56) is a closed form expression yielding an even-order polynomial of degree $2n$. The solutions $f_{2n}(k)$ for $2n = 0, \ldots, 10$ follow and are plotted in Fig. 5.

$$\begin{aligned} f_0(k) &= 1 \\ f_2(k) &= 1 - k^2 \\ f_4(k) &= 1 - 4k^2 + 3k^4 \\ f_6(k) &= 1 - 9k^2 + 18k^4 - 10k^6 \\ f_8(k) &= 1 - 16k^2 + 60k^4 - 80k^6 + 35k^8 \\ f_{10}(k) &= 1 - 25k^2 + 150k^4 - 350k^6 + 350k^8 - 126k^{10} \end{aligned} \tag{57}$$

The function $f_{2n}(k)$ is only defined over the range $0 \le k \le 1$, and the coefficients are integers that always sum to 0 (except when $n = 0$). The new approximation has n maxima and minima over its domain.

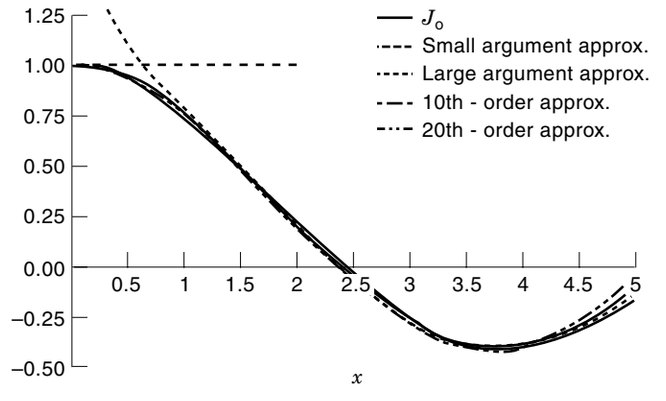


Figure 6. Comparison among $J_0(x)$, classic approximations, and the new approximations.

Bessel Function Approximation

By allowing δ to approach 0 in Eq. (50) and by manipulating the variables, it can easily be shown that

$$J_0(x) \approx f_{2n}\left(\frac{x}{2n}\right); \qquad n \ge 1 \tag{58}$$

Using the identity $J_1(x) = -J_0'(x)$ given in Eq. (36), we have

$$J_1(x) \approx -\frac{d}{dx} f_{2n}\left(\frac{x}{2n}\right) \tag{59}$$

The approximations in Eqs. (58) and (59) are appropriate for any x as long as $x/2n \le 1$. The accuracy increases as $x/2n$ approaches 0. Therefore for small values of x, small-order polynomials are sufficient to approximate $J_0(x)$ and $J_1(x)$. Figure 6 compares the exact solution for $J_0(x)$, classic asymptotic solutions, and the polynomial approximation of Eq. (58) when $2n = 10$ and 20. Figure 7 compares the exact solution for $J_1(x)$, classic asymptotic solutions, and the polynomial approximation of Eq. (59) when $2n = 10$ and 20. It can be seen that the higher-order approximation is understandably better. However, the tenth-order polynomial is quite accurate for $x < 3.5$ in the $J_0(x)$ case and is reasonably accurate for $x < 3$ in the $J_1(x)$ case. If smaller arguments are anticipated, then lower-order polynomials seen in Eq. (57) are sufficient.

The polynomial approximations of Eqs. (58) and (59) are much simpler to express than the polynomials of Eqs. (48) and (49). They are also accurate over a greater range of x for the tenth- and higher-order polynomials.

BIBLIOGRAPHY

1. W. Fricke, Friedrich Wilhelm Bessel (1784–1846), *Astrophys. Space Sci.,* **110** (1): 11–19, 1985.
2. J. Daintith, S. Mitchell, and E. Tootill, *Biographical Encyclopedia of Scientists,* Vol. 1, Facts on File, Inc., 1981.
3. I. Asimov, *Asimov's Biographical Encyclopedia of Science and Technology,* New York: Doubleday, 1982.

4. J. O'Conner and E. Robertson, MacTutor History of Mathematics Archive. University of St. Andrews, St. Andrews, Scotland [Online]. Available www: http://www-groups.dcs.st-and.ac.uk/~history/index.html
5. G. N. Watson, *A Treatise on the Theory of Bessel Functions,* 2nd ed., New York: Macmillan, 1948.
6. N. McLachlan, *Bessel Functions for Engineers,* 2nd ed., Oxford: Clarendon Press, 1955.
7. M. Abramowitz and I. Stegun (eds.), *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables,* National Bureau of Standards, Applied Mathematics Series 55, June, 1964.
8. I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products,* Academic Press, 1980.
9. C. Balanis, *Advanced Engineering Electromagnetics,* New York: Wiley, 1989.
10. Y. Luke, *Mathematical Functions and Their Approximations,* New York: Academic Press, 1975.
11. Y. Luke, *The Special Functions and Their Approximations,* Vol. II, New York: Academic Press, 1969.
12. Y. Luke, *Algorithms for the Computation of Mathematical Functions,* New York: Academic Press, 1977.
13. F. B. Gross and W. Brown, New frequency dependent edge mode current density approximations for TM scattering from a conducting strip grating, *IEEE Trans. Antennas Propag.,* **AP-41**: 1302–1307, 1993.

FRANK B. GROSS
Florida State University

BETA TUNGSTEN SUPERCONDUCTORS, METALLURGY.

See SUPERCONDUCTORS, METALLURGY OF BETA TUNGSTEN.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Boundary-Value Problems
Standard Article

Benjamin Beker[1], George Cokkinides[2], Myung Jin Kong[3]
[1]University of South Carolina, Columbia, SC
[2]University of South Carolina, Columbia, SC
[3]University of South Carolina, Columbia, SC

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2403
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (297K)

Abstract

The sections in this article are

Brief History of Finite Differencing in Electromagnetics

Engineering Basics of Finite Differencing

Governing Equations of Electrostatics

Advanced Topics

Sample Numerical Results

Summary

Acknowledgments

| | |

Copyright © 1999-2008 All Rights Reserved.

BOUNDARY-VALUE PROBLEMS

Many problems in electrical engineering require solution of integral or differential equations which describe physical phenomena. The choice whether integral or differential equations are used to formulate and solve specific problems depends on many factors, whose discussion is beyond the scope of this article. This article strictly concentrates on the use of the finite-difference method (FDM) in the numerical analysis of boundary-value problems associated with primarily static and to some extent quasistatic electromagnetic (EM) fields. Although all examples presented herein deal with EM-related engineering applications, some numerical aspects of FDM are also covered. Since there is a wealth of literature in numerical and applied mathematics about the FDM, little will be said about the theoretical aspects of finite differencing. Such issues as the proof of existence or convergence of the numerical solution will not be covered, while the appropriate references will be provided to the interested reader. Instead, the coverage of FDM will deal with the details about implementation of numerical algorithms, compact storage schemes for large sparse matrices, the use of open boundary truncation, and efficient handling of inhomogeneous and anisotropic materials.

The emphasis will be placed on the applications of FDM to three-dimensional boundary-value problems involving objects with arbitrary geometrical shapes that are composed of complex dielectric materials. Examples of such problems include modeling of discrete passive electronic components, semiconductor devices and their packages, and cross-talk in multiconductor transmission lines. When the wavelength of operation is larger than the largest geometrical dimensions of the object that is to be modeled, static or quasistatic formulation of the problem is appropriate. In electrostatics, for example, this implies that the differential equation governing the physics (voltage distribution) is of Laplace type for the source-free environment and of Poisson type in regions containing sources. The solution of such second-order partial differential equations can be readily obtained using the FDM.

Although the application of FDM to homogeneous materials is simple, complexities arise as soon as inhomogeneity and anisotropy are introduced. The following discussion will provide essential details on how to overcome any potential difficulties in adapting FDM to boundary-value problems involving such materials. The analytical presentation will be supplemented with abundant illustrations that demonstrate how to implement the theory in practice. Several examples will also be provided to show the complexity of problems that can be solved by using FDM.

BRIEF HISTORY OF FINITE DIFFERENCING IN ELECTROMAGNETICS

The utility of the numerical solution to partial differential equations (or PDEs) utilizing finite difference approximation to partial derivatives was recognized early (1). Improvements to the initial iterative solution methods, discussed in Ref. 1, by using relaxation were subsequently introduced (2,3). However, before digital computers became available, the applications of the FDM to the solution of practical boundary-value problems was a tedious and often impractical task. This was especially true if high level of accuracy were required.

With the advent of digital computers, numerical solution of PDEs became practical. They were soon applied to various problems in electrostatics and quasistatics such as in the analysis of microstrip transmission lines (4), among many others. The FDM found quick acceptance in the solution of boundary-value problems within regions of finite extent, and efforts were initiated to extend their applicability to open region problems as well (5).

As a point of departure, it is interesting to note that some of the earliest attempts to obtain the solution to practical boundary-value problems in electrostatics involved experimental methods. They included the electrolytic tank approach and resistance network analog technique (6), among others, to simulate the finite difference approximations to PDEs.

Finally, it should be mentioned that in addition to the application of FDM to static and quasistatic problems, the FDM was also adapted for use in the solution of dynamic, full-wave EM problems in time and frequency domain. Most notably, the use of finite differencing was proposed for the solution to Maxwell's curl equations in the time domain (7). Since then, a tremendous amount of work on the finite-difference time-domain (or FD-TD) approach was carried out in diverse areas of electromagnetics. This includes antennas and radiation, scattering, microwave integrated circuits, and optics. The interested reader can consult the authoritative work in Ref. 8, as well as other articles on eigenvalue and related problems in this encyclopedia, for further details and additional references.

ENGINEERING BASICS OF FINITE DIFFERENCING

It is best to introduce the FDM for the solution of engineering problems, which deal with static and quasistatic electromagnetic fields, by way of example. Today, just about every ele-

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

mentary text in electromagnetics—as well as newer, more numerically focused introductory EM textbooks—will have a discussion on FDM (e.g., pp. 241–246 of Ref. 9 and Section 4.4 of Ref. 10) and its use in electrostatics, magnetostatics, waveguides, and resonant cavities. Regardless, it will be beneficial to briefly go over the basics of electrostatics for the sake of completeness and to provide a starting point for generalizing FDM for practical use.

GOVERNING EQUATIONS OF ELECTROSTATICS

The analysis of electromagnetic phenomena has its roots in the experimental observations made by Michael Faraday. These observations were cast into mathematical form by James Clerk Maxwell in 1873 and verified experimentally by Heinrich Hertz 25 years later. When reduced to electrostatics, they state that the electric field at every point in space within a homogeneous medium obeys the following differential equations (9):

$$\nabla \times \vec{E} = 0 \tag{1}$$

$$\nabla \cdot \epsilon_0 \epsilon_r \vec{E} = \rho_v \tag{2}$$

where ρ_v is the volumetric charge density, ϵ_0 ($\approx 8.854 \cdot 10^{-12}$ Farads/m) is the permittivity of freespace, and ϵ_r is the relative dielectric constant. The use of the vector identity $\nabla \times \nabla\phi \equiv 0$ in Eq. (1) allows for the electric field intensity, $\vec{E}$ (volts/m), to be expressed in terms of the scalar potential $\vec{E} = -\nabla\phi$. When this is substituted for the electric field in Eq. (2), a second-order PDE for the potential ϕ is obtained:

$$\nabla \cdot (\epsilon_r \nabla\phi) = -\rho_v/\epsilon_0 \tag{3}$$

which is known as the Poisson equation. If the region of space, where the solution for the potential is sought, is source-free and the dielectric is homogeneous (i.e., ϵ_r is constant everywhere), Eq. (3) reduces to the Laplace equation $\nabla^2\phi = 0$.

The solution to the Laplace equation can be obtained in several ways. Depending on the geometry of the problem, the solution can be found analytically or numerically using integral or differential equations. In either case, the goal is to determine the electric field in space due to the presence of charged conductors, given that the voltage on their surface is known. For example, if the boundaries of the charged conductors are simple shapes, such as a rectangular box, circular cylinder, or a sphere, then the boundary conditions (constant voltage on the surface) can be easily enforced and the solution can be obtained analytically. On the other hand, when the charged object has an irregular shape, the Laplace equation cannot be solved analytically and numerical methods must be used instead.

The choice as to whether integral or differential equation formulation is used to determine the potential heavily depends on the geometry of the boundary-value problem. For example, if the charged object is embedded within homogeneous medium of infinite extent, integral equations are the preferable choice. They embody the boundary conditions on the potential ϕ at infinity and reduce the numerical effort to finding the charge density on the surface of the conductor (see Sections 5.2 or 4.3 of Ref. 9 or 10 for further details).

On the other hand, when the boundary-value problem includes inhomogeneous dielectrics (i.e., ϵ_r varying from point to point in space), the surface integral equation methods are no longer applicable (or are impractical). Instead, such problems can be formulated using volumetric PDE solvers such as the FDM.

It is important to note that similar considerations (to those stated above) also apply to the solution of the Poisson equation. In this case, in addition to the integration over the conductor boundaries, integration over the actual sources (charge density) must also be performed. The presence of the sources has the same effects on PDEs and FDM, as their effects must be taken into account at all points in space where they exist.

Direct Discretization of Governing Equation

To illustrate the utility and limitations of FDM and to introduce two different ways of deriving the numerical algorithm, consider the geometry shown in Fig. 1. Note that for the sake of clarity and simplicity, the initial discussion will be restricted to two dimensions.

The infinitely long, perfectly conducting circular cylinder in Fig. 1 is embedded between two dielectrics. To determine the potential everywhere in space, given that the voltage on the cylinder surface is V_0, the FDM will be used. There are two approaches that might be taken to develop the FDM algorithm. One approach would be to solve Laplace (or Poisson) equations in each region of uniform dielectric and enforce the boundary conditions at the interface between them. The other route would involve development of a general volumetric algorithm, which would be valid at every point in space, including the interface between the dielectrics. This would involve seeking the solution of a single, Laplace-type (or Poisson) differential equation:

$$\nabla \cdot (\epsilon_r(x,y)\nabla\phi) = \epsilon_r(x,y)\left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}\right) + \left(\frac{\partial\epsilon_r}{\partial x}\frac{\partial\phi}{\partial x} + \frac{\partial\epsilon_r}{\partial y}\frac{\partial\phi}{\partial y}\right) = \begin{cases} 0 \\ -\dfrac{\rho(x,y)}{\epsilon_0} \end{cases} \tag{4}$$

which is valid everywhere, except for the surface of the conductor.

The numerical approach to solving Eq. (4) starts with the approximation to partial derivatives using finite differences.

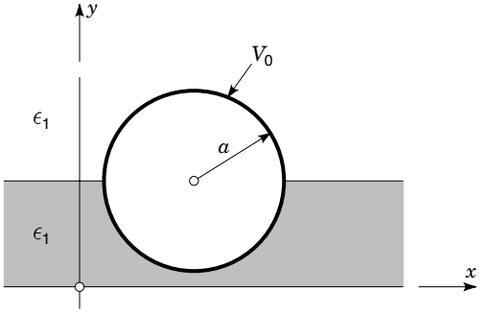


Figure 1. Charged circular cylinder embedded between two different dielectrics.

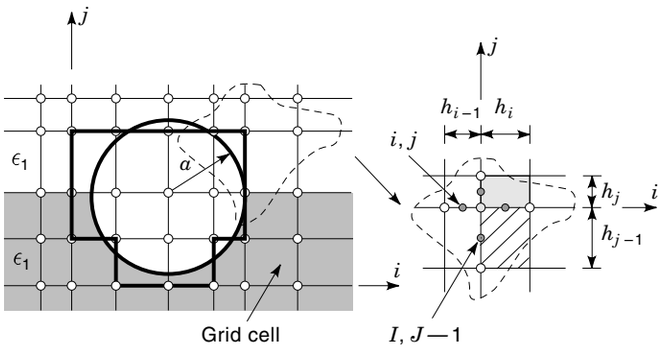


Figure 2. "Staircase" approximation to boundary of circular cylinder and notation for grid dimensions.

This requires some form of discretization for the space (area or volume in two or three dimensions) where the potential is to be computed. The numerical solution of the PDE will lead to the values of the potential at a finite number of points within the discretized space. Figure 2 shows one possible discretization scheme for the cylinder in Fig. 1 and its surroundings. The points form a two-dimensional (2-D) grid and they need not be uniformly spaced. Note that the grid points, where the potential is to be computed, have to be defined all along the grid lines to allow for properly approximating the derivatives in Eq. (4). In other words, the grid lines cannot abruptly terminate or become discontinuous within the grid.

Using finite differences, the first-order derivative at any point in the grid can be approximated as follows:

$$\left(\frac{\partial\phi}{\partial x}\right)_{i,j} \approx \frac{(\phi_{I_1,J} - \phi_{I-1,J})}{(h_i + h_{i-1})/2} = \left[\left(\frac{\phi_{i+1,j} + \phi_{i,j}}{2}\right) - \left(\frac{\phi_{i,j} + \phi_{i-1,j}}{2}\right)\right] \frac{1}{(h_i + h_{i-1})/2} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{h_i + h_{i-1}} \tag{5}$$

with the help of intermediate points I, J (black circles in Fig. 2). The approximation for the second derivatives can be obtained in a similar manner and is given by

$$\left[\frac{\partial}{\partial x}\left(\frac{\partial\phi}{\partial x}\right)\right]_{i,j} \approx \frac{\left(\frac{\partial\phi}{\partial x}\right)_{I+1,J} - \left(\frac{\partial\phi}{\partial x}\right)_{I-1,J}}{(h_i + h_{i-1})/2} = \left[\left(\frac{\phi_{i+1,j} - \phi_{i,j}}{h_i}\right) - \left(\frac{\phi_{i,j} - \phi_{i-1,j}}{h_{i-1}}\right)\right] \times \frac{1}{(h_i + h_{i-1})/2} \tag{6}$$

Once all derivatives in Eq. (4) are replaced with their respective finite-difference approximations and all similar terms are grouped together, the discrete version of Eq. (4) takes on the following form:

$$\phi_{i,j} \approx \frac{1}{Y_{i,j}}(Y_{i+1}\phi_{i+1,j} + Y_{i-1}\phi_{i-1,j} + Y_{j+1}\phi_{i,j+1} + Y_{j-1}\phi_{i,j-1}) + \begin{cases} 0 \\ \dfrac{(\rho_{i,j} + \rho_{i-1,j} + \rho_{i,j-1} + \rho_{i-1,j-1})}{\epsilon_0} \end{cases} \tag{7}$$

In the above equation, the Y factors contain the material parameters and distances between various adjacent grid points. They are expressed below in a compact form:

$$Y_{i\pm1} = \frac{2}{(h_i + h_{i-1})^2}\left\{(\epsilon_{i,j-1} + \epsilon_{i,j})\begin{pmatrix}\dfrac{3h_i + h_{i-1}}{h_i} \\ \dfrac{h_i - h_{i-1}}{h_{i-1}}\end{pmatrix} + (\epsilon_{i-1,j-1} + \epsilon_{i-1,j})\begin{pmatrix}\dfrac{h_{i-1} - h_i}{h_i} \\ \dfrac{h_i + 3h_{i-1}}{h_{i-1}}\end{pmatrix}\right\} \tag{8}$$

$$Y_{j\pm1} = \frac{2}{(h_j + h_{j-1})^2}\left\{(\epsilon_{i-1,j} + \epsilon_{i,j})\begin{pmatrix}\dfrac{3h_j + h_{j-1}}{h_j} \\ \dfrac{h_j - h_{j-1}}{h_{j-1}}\end{pmatrix} + (\epsilon_{i-1,j-1} + \epsilon_{i,j-1})\begin{pmatrix}\dfrac{h_{j-1} - h_j}{h_j} \\ \dfrac{h_j + 3h_{j-1}}{h_{j-1}}\end{pmatrix}\right\} \tag{9}$$

$$Y_{i,j} = Y_{i+1,j} + Y_{i-1,j} + Y_{i,j+1} + Y_{i,j-1} \tag{10}$$

It is important to add that in deriving the above equations, a particular convention for associating the medium parameters to individual grid cells was employed. Specifically, it was assumed that the medium parameter values of the entire grid cell were associated with (or assigned to) the lower left corner of that cell. For example, $\epsilon_{i,j}$ is assumed to be constant over the shaded grid cell area shown in Fig. 2, while $\epsilon_{i,j-1}$ is constant over the hatched area, which is directly below.

Observe what are the consequences of converting the continuous PDE given in Eq. (4) to its approximate discrete form stated in Eq. (7). First, the boundary-value problem over the continuous space, shown in Fig. 1, was "mapped" onto a discrete grid (see Fig. 2). Clearly, if the number of grid points increases, then spacing between them will become smaller. This provides a better approximation to the actual continuous problem. In fact, in the limit as the number of grid point reaches infinity, the discrete and continuous problems become identical.

In addition to illustrating the "mapping" of a continuous problem to its discrete analog, Fig. 2 also clearly demonstrates one of FDM's undesirable artifacts. Note that objects with smooth surfaces are replaced with a "staircased" approximation. Obviously, this approximation can be improved by reducing the discretization grid spacing. However, this will increase the number of points where the potential has to be calculated, thus increasing the computational complexity of the problem. One way to overcome this is to use a nonuniform discretization, as depicted in Fig. 2. Specifically, finer discretization can be used in the region near the smooth surface of the cylinder to better approximate its shape, followed by gradually increasing the grid point spacing between the cylinder and grid truncation boundary.

At this point, it is also appropriate to add that other, more rigorous methods have been proposed for incorporating curved surfaces into the finite-difference type of algorithms. They are based on special-purpose differencing schemes,

which are derived by recasting the same PDEs into their equivalent integral forms. They exploit the surface or contour integration and are used to replace the regular differencing algorithm on the curved surfaces or contours of smooth objects. This approach was already implemented for the solution of dynamic full-wave problems (11) and could be adapted to electrostatic boundary-value problems as well.

Finally, Eq. (7) also shows that the potential at any point in space, which is source-free, is a weighted average of the potentials at the neighboring points only. This is typical of PDEs, because they only represent physical phenomena locally—that is, in the immediate vicinity of the point of interest. As will be shown later, one way to "propagate" the local information through the grid is to use an iterative scheme. In this scheme, the known potential, such as V_0 at the surface of the conducting cylinder in Fig. 1, is carried throughout the discretized space by stepping through all the points in the grid. The iterations are continued until the change in the potential within the grid is very small.

"Indirect" Discretization of Governing Equation

As shown in the previous section, appropriate finite-difference approximations were required for the first- and second-order derivatives in order to convert Eq. (4) to a discrete form. Intermediate points (I, J) were used midway between the regular grid nodes for obtaining average values of the potential, its first derivatives, and dielectric constants to facilitate the derivation of the final update equation for the potential. This can be avoided and an alternative, but equally valid finite differencing scheme to that given in Eq. (7), which for the sake of brevity is restricted to the Laplace equation only, can be obtained. The first step is to simplify Eq. (4) by recasting it into an integrodifferential form. To achieve this, Eq. (4) should be integrated over a volume, which completely encloses any one of the grid nodes. This will be referred to as the volume of the unit cell (V_U), which is bounded by surface S_U (see Fig. 3). Stoke's theorem is applied to replace the volume integration by a surface integral:

$$\begin{aligned}\int_{V_U} \nabla \cdot (\epsilon_r(x,y)\nabla\phi)\, dv &= \oint_{S_U} (\epsilon_r(x,y)\nabla\phi) \cdot \hat{n}\, ds \\ &= \oint_{S_U} \epsilon_r(x,y)\frac{\partial\phi}{\partial n} ds = 0\end{aligned} \tag{11}$$

where $\hat{n}$ is the unit vector, normal to S_U and pointing out of it.

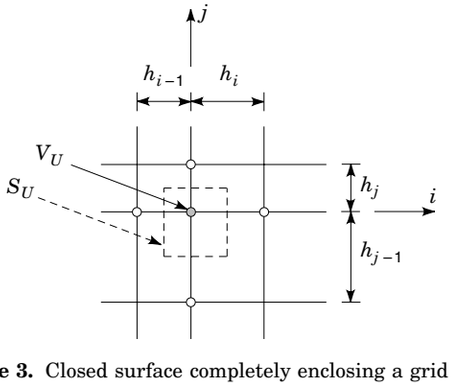


Figure 3. Closed surface completely enclosing a grid node.

To illustrate this "indirect" discretization procedure in 2-D, consider surface $S_{Ui,j}$ (that is, just a contour) shown in Fig. 3, which completely encloses grid point i, j. The integral in Eq. (11) reduces to four terms, each corresponding to one of the faces of $S_{Ui,j}$. For example, the integral over the right edge (or face) of $S_{Ui,j}$ can be approximated as

$$\frac{\phi_{i+1,j} - \phi_{i,j}}{h_i}\left(\frac{h_{j-1}}{2}\epsilon_{i,j-1} + \frac{h_j}{2}\epsilon_{i,j}\right) \tag{12}$$

When the remaining integrals are evaluated and the like terms are grouped together, the final form of the "indirect" FDM algorithm is obtained. This algorithm is identical in form to that given earlier in Eq. (7). However, the weighting Y factors are different from those appearing in Eqs. (8) and (9). Their complete expressions are given by

$$Y_{\left\{\begin{smallmatrix} i+1 \\ i-1 \end{smallmatrix}\right\}} = \left(h_{j-1}\epsilon_{\left\{\begin{smallmatrix} i,j-1 \\ i-1,j-1 \end{smallmatrix}\right\}} + h_j\epsilon_{\left\{\begin{smallmatrix} i,j \\ i-1,j \end{smallmatrix}\right\}}\right)\frac{1}{2h_{\left\{\begin{smallmatrix} i \\ i-1 \end{smallmatrix}\right\}}} \tag{13}$$

with $Y_{i,j}$ being the sum of all other Y's, the same as before [see Eq. (10)].

It should be added that this approach has been suggested several times in the literature—for example, in Refs. 12 and 13. Therein, Eq. (11) was specifically used to enforce the boundary conditions at the interface between different dielectrics only in order to "connect" FDM algorithms based on the Laplace equation for the homogeneous regions. However, there is no reason not to use Eq. (11) at every point in space, especially if the boundary-value problem involves inhomogeneous media. This form of FDM scheme is completely analogous to that presented in the previous section. In fact, it is a little simpler to derive and involves a fewer number of arithmetic operations.

Numerical Implementation in Two-Dimensions

There are several important numerical issues that must be addressed prior to implementing FDM on the computer. Such questions as how to terminate the grid away from the region of interest and which form of the FDM algorithm to choose must be answered first. The following discussion provides some simple answers, postponing the more detailed treatment until later.

Simplistic Grid Boundary Truncation. Clearly, since even today's computers do not have infinite resources, the computational volume (or space) must somehow be terminated (see Fig. 4). The simplest approach is to place the truncation boundary far away from the region of interest and to set the potential on it equal to zero. This approach is valid as long as the truncation boundary is placed far enough not to interact with the charged objects within, as for example the "staircased" cylinder shown in Fig. 4. The downside of this approach is that it leads to large computational volumes, thereby requiring unnecessarily high computer resources. This problem can be partly overcome by using a nonuniform grid, with progressively increasing spacing from the cylinder toward the truncation boundary. It should be added that there are other ways to simulate the open-boundary conditions, which is an advanced topic and will be discussed later.

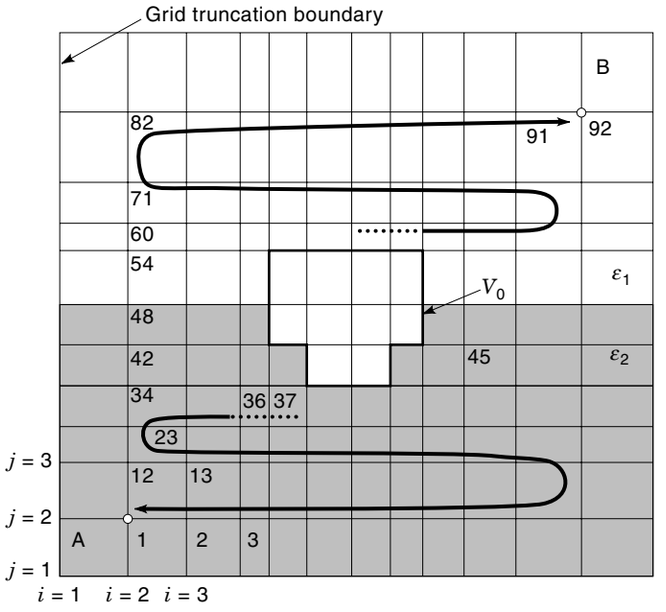


Figure 4. Complete discretized geometry and computational space for cylinder in Fig. 2.

Iteration-Based Algorithm. Several iteration methods can be applied to solve Eq. (7), each leading to different convergence rates (i.e., how fast an acceptable solution is obtained). A complete discussion of this topic, as well as of the accuracy of the numerical approximations in FDM, appears in Ref. 14 and will not be repeated here. The interested reader may also find Ref. 15 quite useful, because it covers such topics in more rigorous detail and includes a comprehensive discussion on the proof of the existence of the finite-differencing solution to PDEs. However, for the sake of brevity, this article deals with the most popular and widely used approach, which is called successive overrelaxation (SOR) (see, e.g., Ref. 14).

SOR is based on Eq. (7), which is rearranged as

$$\begin{aligned}\phi_{i,j}^{p+1} &\approx \phi_{i,j}^{p} + \frac{\Omega}{Y_{i,j}}(Y_{i+1}\phi_{i+1,j}^{p} + Y_{i-1}\phi_{i-1,j}^{p+1} + Y_{j+1}\phi_{i,j+1}^{p} \\ &\quad + Y_{j-1}\phi_{i,j-1}^{p+1} - Y_{i,j}\phi_{i,j}^{p}) \\ &= (1-\Omega)\phi_{i,j}^{p} + \frac{\Omega}{Y_{i,j}}(Y_{i+1}\phi_{i+1,j}^{p} + Y_{i-1}\phi_{i-1,j}^{p+1} \\ &\quad + Y_{j+1}\phi_{i,j+1}^{p} + Y_{j-1}\phi_{i,j-1}^{p+1})\end{aligned} \tag{14}$$

In the above equation, superscripts p and $p+1$ denote the present and previous iteration steps and Ω is the so-called overrelaxation factor, whose value can vary from 1 to 2. Note that Ω accelerates the change in the potential from one iteration to the next at any point in the grid. It can be a constant throughout the entire relaxation (solution) process or be varying according to some heuristic scheme. For example, it was found that the overall rate of convergence is improved by setting Ω near 1.8 at the start of the iteration process and gradually reducing it to 1.2 with the numbers of iterations.

At this point, the only remaining task is to define the appropriate criteria for terminating the FDM algorithm. Although there are rigorous ways of selecting the termination criterion, the following approach, which seems to work quite well, is presented instead. As stated below, it is based on calculating the change in the potential between successive iterations at every point in the grid and comparing the maximum value to the (user-selectable) error criterion:

$$\mathrm{ERR}_{\max} = \sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\max(|\phi_{i,j}^{p+1} - \phi_{i,j}^{p}|) \tag{15}$$

Regardless of it being simple, the redeeming feature of this approach is that the error is computed globally within the grid, rather than within a particular single grid node. The danger in monitoring the convergence of the algorithm at a single node may lead to premature termination or to an unnecessarily prolonged execution.

Now that an error on which the algorithm termination criterion is based has been defined, the iteration process can be initiated. Note that there are several ways to "march" through the grid. Specifically, the updating of the potential may be started from point A, as shown in Fig. 4, and end at point B, or vice versa. If the algorithm works in this manner, the solution will tend to be artifically "biased" toward one region of the grid, with the potential being "more converged" in regions where the iteration starts. The obvious way to avoid this is to change the direction of the "marching" process after every few iterations. As a result, the potential will be updated throughout the grid uniformly and will converge at the same rate. Note from Fig. 4 that the potential only needs to be computed at the internal points of the grid, since the potential at the outer boundary is (for now) assumed to be zero. Moreover, it should also be evident that the potential at the surface of the cylinder, as well as inside it, is known (V_0) and need not be updated during the iteration process.

Matrix-Based Algorithm. Implementation of the finite-difference algorithms is not restricted to relaxation techniques only. The solution of Eq. (4) for the electrostatic potential can also be obtained using matrix methods. To illustrate this, the FDM approximation to Eq. (4)—namely, Eq. (7)—will be rewritten as

$$\begin{aligned}(Y_{i+1}\phi_{i+1,j} + Y_{i-1}\phi_{i-1,j} + Y_{j+1}\phi_{i,j+1} + Y_{j-1}\phi_{i,j-1}) \\ - Y_{i,j}\phi_{i,j} \approx 0\end{aligned} \tag{16}$$

The above equation must be enforced at every internal point in the grid, except at the surface of and internal to the conductors, where the potential is known (V_0). For the particular example of the cylinder shown in Fig. 4, these points are numbered 1 through 92. This implies that there are 92 unknowns, which must be determined. To accomplish this, Eq. (16) must be enforced at 92 locations in the grid, leading to a system of 92 linear equations that must be solved simultaneously.

To demonstrate how the equations are set up, consider nodes 1, 36, and 37 in detail. At node 1 (where $i = j = 2$), Eq. (16) reduces to

$$Y_3^i\phi_{12} + Y_3^j\phi_2 - Y_{2,2}\phi_1 = 0 \tag{17}$$

where the fact that the potential at the outer boundary nodes $(i, j) = (1, 2)$ and $(2, 1)$ is zero was taken into account and superscripts i and j on Y's were introduced as a reminder

whether they correspond to $Y_{i\pm1}$ or $Y_{j\pm1}$. In addition, the potential at nodes $(i, j) = (2, 2)$, $(3, 2)$ and $(2, 3)$ was also relabeled as ϕ_1, ϕ_2, and ϕ_{12}, respectively. Similarly, at nodes 36 ($i, j = 4, 5$) and 37 ($i, j = 5, 5$), Eq. (16) becomes

$$Y_5^i\phi_{37} + Y_3^i\phi_{35} + Y_6^j\phi_{44} + Y_4^j\phi_{25} - Y_{4,5}\phi_{36} = 0 \tag{18}$$

and

$$Y_4^i\phi_{36} + Y_4^j\phi_{26} - Y_{5,5}\phi_{37} = -Y_6^iV_0 - Y_6^jV_0 = V_{37} \tag{19}$$

where in Eq. (19) the known quantities (the potentials on the surface of the cylinder at nodes 5, 6 and 6, 5) were moved to the right-hand side.

Similar equations can be obtained at the remaining free grid nodes where the potential is to be determined. Once Eq. (16) has been enforced everywhere within the grid, the resulting set of equations can be combined into the following matrix form:

$$\begin{bmatrix} -Y_{2,2} & Y_3^j & 0 & \cdot & 0 & Y_3^i & 0 & \cdot \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ 0 & 0 & \cdot & \cdot & 0 & 0 & 0 & Y_4^j \\ 0 & 0 & \cdot & \cdot & 0 & 0 & 0 & 0 \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & &

3. Look up k using $k = \text{INDEX}(m)$, thereby identifying the appropriate ϕ_k, given i, j.

The criteria for selecting a particular sparsity coding scheme for the matrix [Y] are (a) the minimization of storage requirements and (b) optimization of matrix operations—in particular, multiplication and LU factoring. One very efficient scheme is based on storing [Y] using four one-dimensional arrays:

1. Real array DIAG(i) = diagonal entry of row i
2. Real array OFFD(i) = ith nonzero off-diagonal entry (scanned by rows)
3. Integer array IROW(i) = index of first nonzero off-diagonal entry of row i
4. Integer array ICOL(i) = column number of ith nonzero off-diagonal entry (scanned by rows)

Assuming a system of N equations, the arrays DIAG, IROW, OFFD, and ICOL have N, $N + 1$, $4N$, and $4N$ entries, respectively. Therefore, the total memory required to store a sparsity coded matrix [Y] is approximately $40N$ bytes (assuming 32-bit storage for both real and integer numbers). On the other hand, $4N^2$ bytes would be needed to store the full form of [Y]. For example, in a system with 1000 equations, the full storage mode requires 4 megabytes, while the sparsity coded matrix occupies only 40 kilobytes of computer memory.

Perhaps the most important feature of sparsity coding is the efficiency with which multiplication and other matrix operations can be performed. This is best illustrated by a sample FORTRAN coded needed to multiply a matrix stored in this mode, by a vector $B(i)$:

```
DO I =1, N
      C(I) = DIAG(I) * B(I)
      DO J = IROW(I), IROW(I + 1) − 1
      C(I) = C(I) + OFFD(J) * B(ICOL(J))
      ENDDO
ENDDO
```

(23)

The above double loop involves $5N$ multiplications and $4N$ additions, without the need of search and compare operations. To perform the same operation using the brute force, full storage approach would require N^2 multiplications and N^2 additions. Thus, for a system of 1000 unknowns, the sparsity-based method is at least 200 times faster than the full storage approach in performing matrix multiplication.

To solve Eq. (20), [Y] can be inverted and the inverse multiplied by [V_0]. However, for sparse systems, complete matrix inversion should be avoided. The reason is that, in most cases, the inverse of a sparse matrix is full, for which the advantages of sparsity coding cannot be exploited. The solution of sparsity coded linear systems is typically obtained by using the LU decomposition, since usually the L and U factors are sparse. Note that the sparsity of the L and U factor matrices can be significantly affected by the ordering of the grid nodes (i.e., in which sequence [ϕ] was filled). Several very successful node ordering schemes that are associated with the analysis of electrical networks were reported for the solution of sparse matrix equations (see Ref. 16). Unfortunately, the grid node connectivity in typical FDM problems is such that the L and U factor matrices are considerably fuller than the original matrix [Y], even if the nodes are optimally ordered. Therefore, direct solution techniques are not as attractive for use in FDM as they are for large network problems.

Unlike the direct solution methods, there are iterative techniques for solving matrix equations, which do not require LU factoring. One of them is the Conjugate Gradient Method (17–19).

This method uses a sequence of matrix/vector multiplications, which can be performed very efficiently using the sparsity coding scheme described above.

Convergence. Given the FDM equations in matrix form, either direct (16) or iterative methods (17–19) such as the Conjugate Gradient Method (CGM) can be readily applied. It is important to point out that CGM-type algorithms are considerably faster than direct solution, provided that a good initial guess is used. One simple approach is to assume that the potential is zero everywhere, but on the surface of the conductor, and let this be an initial guess to start the CGM algorithm. For this initial guess, the convergence is very poor and the solution takes a long time. To improve the initial guess, several iterations of the SOR-based FDM algorithm can be performed to calculate the potential everywhere within the grid. It was found that for many practical problems, 10 to 15 iterations provide a very good initial guess for CGM.

From the performance point of view, the speed of CGM was most noticeable when compared to the SOR-based algorithm. In many problems, CGM was found to be an order of magnitude faster than SOR. In all fairness to SOR, its implementation, as described above, can be improved considerably by using the so-called multigrid/multilevel acceleration (20–21). The idea behind this method is to perform the iterations over coarse and fine grids alternatively, where the coarse grid points also coincide with and are a part of the fine grid. This means that iterations are first performed over a coarse grid, then interpolated to the fine grid and iterated over the fine grid. More complex multigrid schemes involve several layers of grids with different levels of discretization, with the iterations being performed interchangeably on all grids.

Finally, it should be pointed out that theoretical aspects of convergence for algorithms discussed thus far are well-documented and are outside the main scope of this article. The interested reader should consult Refs. 15, 18, or 19 for detailed mathematical treatment and assessment of convergence.

ADVANCED TOPICS

Open Boundary Truncation

If the electrostatic boundary-value problem consists of charged conductors in a region of infinite extent, then the simplest approach to truncate the computational (or FDM) boundary is with an equipotential wall of zero voltage. This has the advantage of being easy to implement, but leads to erroneous solution, especially if the truncation boundary is too close to the charged conductors. On the other hand, placing it too far from the region of interest may result in unacceptably large computational volume, which will require large computational resources.

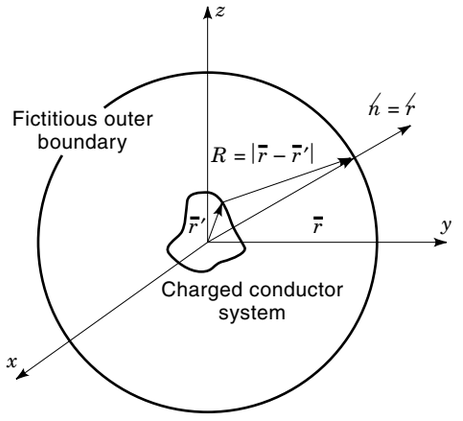


Figure 5. Virtual surface used for boundary truncation.

Although some early attempts to overcome such difficulties (5) provided the initial groundwork, rigorous absorbing boundary truncation operators were recently introduced (22–24) for dynamic problems, which can be modified for electrostatics. They are based on deriving mathematical operators that help simulate the behavior of the potential on a virtual boundary truncation surface, which is placed close to charged conductors (see Fig. 5). In essence, these operators provide the means for numerically approximating the proper behavior of the potential at infinity within a computational volume of finite extent.

Such absorbing boundary conditions (ABCs) are based on the fact that the potential due to any 3-D charge distribution is inversely proportional to the distance measured from it. Consider an arbitrary collection of charged conductors shown in Fig. 5. Although it is located in free unbounded space, a fictitious surface will be placed around it, totally enclosing all conductors. If this surface is far away from the charged conductor system—namely, if $\vec{r}$ is much greater that $\vec{r}'$—then the dominant radial variation of potential, ϕ, will be given by

$$\frac{1}{|\vec{r} - \vec{r}'|} \to \frac{1}{r} \tag{24}$$

If the fictitious boundary is moved closer toward the conductor assembly, then the potential will also include additional terms with higher inverse powers of r. These terms will contribute to the magnitude of the potential more significantly than those with lower inverse powers of r, as r becomes small.

The absorbing boundary conditions emphasize the effect of leading (dominant) radial terms on the magnitude of the potential evaluated on the fictitious (boundary truncation) surface. The ABCs provide the proper analytic means to annihilate the nonessential terms, instead of simply neglecting their contribution. Numerically, this can be achieved by using the so-called absorbing boundary truncation operators.

In general, absorbing boundary operators can be of any order. For example, as shown in Ref. 23, the first- and second-order operators in 3-D have the following forms:

$$B_1 u = \frac{\partial u}{\partial r} + \frac{u}{r} = O\left(\frac{1}{r^3}\right) \to 0 \qquad \text{as } r \to \infty \tag{25}$$

$$B_2 u = \left(\frac{\partial}{\partial r} + \frac{3}{r}\right)\left(\frac{\partial u}{\partial r} + \frac{u}{r}\right) = O\left(\frac{1}{r^5}\right) \to 0 \qquad \text{as } r \to \infty \tag{26}$$

where u is the scalar electric potential function, ϕ, that satisfies the Laplace equation, and $r = |\vec{r}|$ is the radial distance measured from the coordinate origin (see Fig. 5). Since FDM is based on the iterative solution to the Laplace equation, small increases in the overall lattice (discretized 3-D space whose planes are 2-D grids) size do not slow the algorithm down significantly, nor do they require an excessive amount of additional computer memory. As a result, from a practical standpoint, the fictitious boundary truncation surface need not be placed too close to the region of interest, therefore not requiring the use of high-order ABC operators in order to simulate proper behavior of the potential at lattice boundaries accurately. Consequently, in practice, it is sufficient to use the first-order operator, B_1, to model open boundaries. Previous numerical studies suggest that this choice is indeed adequate for many engineering problems (25).

To be useful for geometries that mostly conform to rectangular coordinates, the absorption operator B_1, when expressed in Cartesian coordinates, takes on the following form:

$$\frac{\partial u}{\partial x} \approx \mp\left(\frac{u}{x} + \frac{y}{x}\frac{\partial u}{\partial y} + \frac{z}{x}\frac{\partial u}{\partial z}\right) \tag{27}$$

$$\frac{\partial u}{\partial y} \approx \mp\left(\frac{u}{y} + \frac{x}{y}\frac{\partial u}{\partial x} + \frac{z}{y}\frac{\partial u}{\partial z}\right) \tag{28}$$

$$\frac{\partial u}{\partial z} \approx \mp\left(\frac{u}{z} + \frac{x}{z}\frac{\partial u}{\partial x} + \frac{y}{z}\frac{\partial u}{\partial y}\right) \tag{29}$$

where $\mp$ signs correspond to the outward pointing unit normal vectors $\hat{n} = \pm(\hat{x}, \hat{y}, \hat{z})$ for operators in Eqs. (27), (28), and (29), respectively.

The finite-difference approximations to the above equations which have been employed in the open boundary FDM algorithm are given by

$$u_{i+1,j,k} = u_{i-1,j,k} - \frac{(h_i + h_{i-1})}{(x_{i,j,k} - x_{\text{ref}})} \left[u_{i,j,k} + \frac{(y_{i,j,k} - y_{\text{ref}})(u_{i,j+1,k} - u_{i,j-1,k})}{(h_j + h_{j-1})} + \frac{(z_{i,j,k} - z_{\text{ref}})(u_{i,j,k+1} - u_{i,j,k-1})}{(h_k + h_{k-1})}\right] \tag{30}$$

$$u_{i,j+1,k} = u_{i,j-1,k} - \frac{(h_j + h_{j-1})}{(y_{i,j,k} - y_{\text{ref}})} \left[u_{i,j,k} + \frac{(x_{i,j,k} - x_{\text{ref}})(u_{i+1,j,k} - u_{i-1,j,k})}{(h_i + h_{i-1})} + \frac{(z_{i,j,k} - z_{\text{ref}})(u_{i,j,k+1} - u_{i,j,k-1})}{(h_k + h_{k-1})}\right] \tag{31}$$

$$u_{i,j,k+1} = u_{i,j,k-1} - \frac{(h_k + h_{k-1})}{(z_{i,j,k} - z_{\text{ref}})} \left[u_{i,j,k} + \frac{(x_{i,j,k} - x_{\text{ref}})(u_{i+1,j,k} - u_{i-1,j,k})}{(h_i + h_{i-1})} + \frac{(y_{i,j,k} - y_{\text{ref}})(u_{i,j+1,k} - u_{i,j-1,k})}{(h_j + h_{j-1})}\right] \tag{32}$$

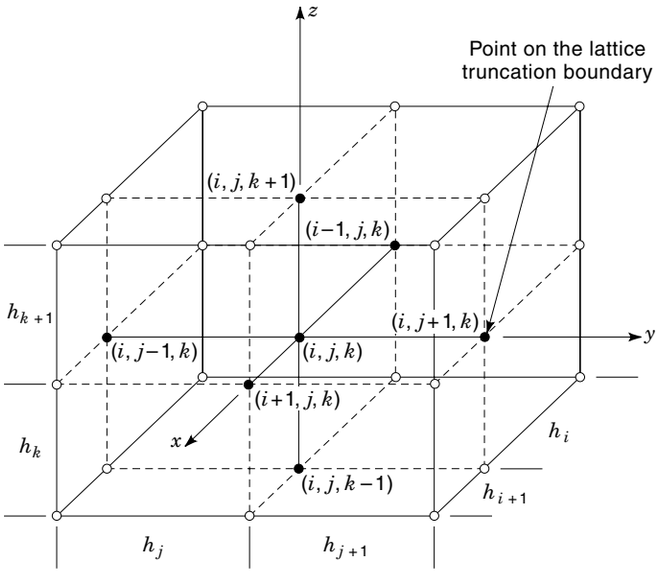


Figure 6. Detail of FDM lattice near the boundary truncation surface.

where $(x, y, z)_{\text{ref}}$ are the x, y, and z components of a vector pointing (referring) to the geometric center of the charged conductor assembly, with other quantities that appear in Eqs. (30) through (32) shown in Fig. 6. It is important to add that the $(x, y, z)_{i,j,k} - (x, y, z)_{\text{ref}}$ terms are the x, y, and z components of a vector from the truncation boundary to the geometrical center of the charged conductor system.

Notice that Fig. 6 graphically illustrates the FDM implementation of the open boundary truncation on lattice faces aligned along the xz plane. On this plane, the normal is in the y direction, for which Eq. (31) is the FDM equivalent of the first-order absorbing boundary operator in Cartesian coordinates. Similarly, Eqs. (32) and (30) are used to simulate the open boundary on xy and yz faces of the lattice, respectively.

When reduced to two dimensions, the absorption operator, B_1, in Cartesian coordinates, takes on the form given below:

$$\frac{\partial u}{\partial x} \approx \mp\left(\frac{1}{x} + \frac{y}{x}\frac{\partial u}{\partial y}\right) \tag{33}$$

$$\frac{\partial u}{\partial y} \approx \mp\left(\frac{1}{y} + \frac{\partial u}{\partial x}\right) \tag{34}$$

The discrete versions of the above equations can be written as

$$u_{i+1,j} = \mp\left\{u_{i-1,j} - \frac{(h_i + h_{i-1})}{(x_{i,j} - x_{\text{ref}})}\left[u_{i,j} + \frac{(y_{i,j} - y_{\text{ref}})(u_{i,j+1} - u_{i,j-1})}{(h_j + h_{j-1})}\right]\right\} \tag{35a}$$

$$u_{i,j+1} = \mp\left\{u_{i,j-1} - \frac{(h_j + h_{j-1})}{(y_{i,j} - y_{\text{ref}})}\left[u_{i,j} + \frac{(x_{i,j} - x_{\text{ref}})(u_{i+1,j} - u_{i-1,j})}{(h_i + h_{i-1})}\right]\right\} \tag{35b}$$

where, as before, $\mp$ correspond to the outward pointing unit normal vectors $\hat{n} = \pm(\hat{x}, \hat{y})$ for operators in Eqs. (33) and (34) or (35a) and (35b), respectively. The points $x_{i,j}$ and $y_{i,j}$ denote those points in the grid that are located one cell away from the truncation boundary, while x_{ref} and y_{ref} correspond to the center of the cylinder in Figs. 2 and 4.

Finally, another approach to open boundary truncation, which is worth mentioning, involves the regular finite-difference scheme supplemented by the use of electrostatic surface equivalence (26). A virtual surface S_v is defined near the actual grid truncation boundary. The electrostatic potential due to charged objects, enclosed within S_v, is computed using the regular FDM algorithm. Subsequently, it is used to calculate the surface charge density and surface magnetic current, which are proportional to the normal and tangential components of the electric field on S_v.

Once the equivalent sources are known, the potential between the virtual surface and the grid truncation boundary can be readily calculated (for details see Ref. 26). This procedure is repeated every iteration, and since the potential on the virtual surface is estimated correctly, it produces a physical value of the potential on the truncation boundary. As demonstrated in Ref. 26, this approach leads to very accurate results in boundary-value problems with charged conductors embedded in open regions.

It is vastly superior to simply using the grounded conductor to terminate the computational space.

Inclusion of Dielectric Anisotropy

Network Analog Approach. Many materials such as printed circuit board and microwave circuit substrates, which are commonly used in electrical engineering exhibit anisotropic behavior. The electrical properties of these materials vary with direction and have to be described by a tensor instead of a single scalar quantity. This section will examine how the anisotropy affects the FDM and how the algorithnm must be changed to accommodate the solution of 3-D problems containing such materials. The theoretical development presented below is a generalization of that available in Ref. 27 and is restricted to linear anisotropic dielectrics only.

In an attempt to provide a more intuitive interpretation to the abstract nature of the FDM algorithm, an equivalent circuit model will be used for linear inhomogeneous, anisotropic regions. This approach is called *resistance network analog* (6). It was initially proposed for approximating the solution of the Laplace equation in two dimensions experimentally, with a network of physical resistors whose values could be adjusted to correspond to the weighting factors [e.g., the Y's in Eq. (7)] that appear in the FDM algorithm. Since its introduction, the resistance network approach has been implemented numerically in the analysis of (a) homogeneous dielectrics in 3-D (28) and (b) simple biaxial anisotropic materials (described by diagonal permitivitty tensors) in 2-D (29).

Since the resistance network analog gives a physical intepretation to FDM, the discretized versions of the Laplace equations for anisotropic media will be recast into this form. As the details of FDM were described earlier, only the key steps in developing the two-dimensional model are summarized below. Moreover, for the sake of brevity, the discussion of the three-dimensional case will be limited to the final equations and their pictorial interpretation.

The Laplace equation for boundary-value problems involving inhomogeneous and anisotropic dielectrics in three-dimensions is given by

$$\nabla \cdot (\epsilon_0[\epsilon_r(x,y,z)] \cdot \nabla\phi(x,y,z)) = 0 \tag{36}$$

In the above equation, $[\epsilon_r]$ stands for the relative dielectric tensor and is defined as

$$[\epsilon] = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} & \epsilon_{xz} \\ \epsilon_{yx} & \epsilon_{yy} & \epsilon_{yz} \\ \epsilon_{zx} & \epsilon_{zy} & \epsilon_{zz} \end{bmatrix} \tag{37}$$

Since the material properties need not be homogeneous in the region of interest, the elements of $[\epsilon_r]$ are assumed to be functions of position. The dielectric is assumed to occupy only part of the modeling (computational) space, and its properties may vary from point to point. When Eq. (37) is substituted into Eq. (36) and rewritten in a matrix form as

$$\left[\frac{\partial}{\partial x}\frac{\partial}{\partial y}\frac{\partial}{\partial z}\right] \cdot \begin{bmatrix} \epsilon_{xx}\dfrac{\partial\phi}{\partial x} + \epsilon_{xy}\dfrac{\partial\phi}{\partial y} + \epsilon_{xz}\dfrac{\partial\phi}{\partial z} \\ \epsilon_{yx}\dfrac{\partial\phi}{\partial x} + \epsilon_{yy}\dfrac{\partial\phi}{\partial y} + \epsilon_{yz}\dfrac{\partial\phi}{\partial z} \\ \epsilon_{zx}\dfrac{\partial\phi}{\partial x} + \epsilon_{zy}\dfrac{\partial\phi}{\partial y} + \epsilon_{zz}\dfrac{\partial\phi}{\partial z} \end{bmatrix} = 0 \tag{38}$$

it provides the starting point for developing the corresponding FDM algorithm.

After eliminating the z-dependent terms and fully expanding the above equation by following the notation used throughout this paper, the finite-difference approximation for the potential at every nodal point in a 2-D grid is given by

$$\phi_{i,j}^{p+1} = (1-\Omega)\phi_{i,j}^{p} + \frac{\Omega}{Y_{i,j}} \times \left\{ \begin{array}{l} (\phi_{i+1,j}^{p} Y_{i+1}\phi_{i-1,j}^{p+1} Y_{i-1}) \\ +(\phi_{i,j+1}^{p} Y_{j+1} + \phi_{i,j-1}^{p+1} Y_{j-1}) \\ +(\phi_{i+1,j+1}^{p} Y_{i+1,j+1} + \phi_{i-1,j-1}^{p+1} Y_{i-1,j-1}) \\ -(\phi_{i-1,j+1}^{p} Y_{i-1,j+1} + \phi_{i+1,j-1}^{p+1} Y_{i+1,j-1}) \end{array} \right\} \tag{39}$$

where

$$Y_{i\pm1} = \begin{pmatrix} (\epsilon_{i,j-1}^{yy} + \epsilon_{i,j}^{yy}) \\ (\epsilon_{i-1,j-1}^{yy} + \epsilon_{i-1,j}^{yy}) \end{pmatrix} \left(\frac{1}{h_i} + \frac{2}{h_i + h_{i-1}}\right) + \left(\frac{2}{h_i + h_{i-1}}\right)\left(\frac{2}{h_j + h_{j-1}}\right)\left[\left(\epsilon_{i,j}^{zy} + \epsilon_{i-1,j}^{zy}\right) - \left(\epsilon_{i-1,j-1}^{zy} + \epsilon_{i,j-1}^{zy}\right)\right] \tag{40}$$

$$Y_{j\pm1} = \begin{pmatrix} (\epsilon_{i-1,j}^{zz} + \epsilon_{i,j}^{zz}) \\ (e_{i-1,j-1}^{zz} + \epsilon_{i,j-1}^{zz}) \end{pmatrix} \left(\frac{1}{h_j} + \frac{2}{h_j + h_{j-1}}\right) + \left(\frac{2}{h_i + h_{i-1}}\right)\left(\frac{2}{h_j + h_{j-1}}\right)\left[\left(\epsilon_{i,j-1}^{yz} + \epsilon_{i,j}^{yz}\right) - \left(\epsilon_{i-1,j-1}^{yz} + \epsilon_{i-1,j}^{yz}\right)\right] \tag{41}$$

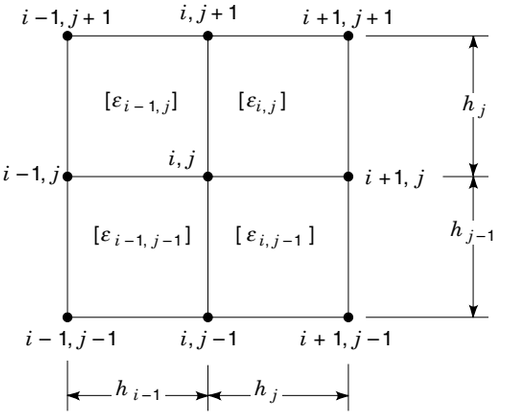


Figure 7. Detail with FDM cell for anisotropic medium in two dimensions.

$$Y_{i\pm1,j\pm1} = Y_{i\pm1,j\mp1} = \left(\frac{2}{h_i + h_{i-1}}\right)\left(\frac{2}{h_j + h_{j-1}}\right) \left[\epsilon_{i,j}^{yz} + \epsilon_{i-1,j}^{yz} + \epsilon_{i-1,j-1}^{yz} + \epsilon_{i,j-1}^{yz} + \epsilon_{i,j}^{zy} + \epsilon_{i-1,j}^{zy} + \epsilon_{i-1,j-1}^{zy} + \epsilon_{i,j-1}^{zy}\right] \tag{42}$$

$$Y_{i,j} = Y_{i+1} + Y_{i-1} + Y_{j+1} + Y_{j-1} \tag{43}$$

Note that unlike the treatment of isotropic dielectrics, the permittivity of each cell is now described by a tensor (see Fig. 7). In addition, the presence of the anisotropy is responsible for added coupling between the voltage $\phi_{i,j}$ and voltages $\phi_{i\pm1,j\pm1}$ (actually all four combinations of the subscripts).

The symbols Y in Eq. (39) can be interpreted as admittances representing the "electrical link" between the grid point voltages. The resulting equivalent network for Eq. (39) can thus be represented pictorially as shown in Fig. 8.

Similarly, after fully expanding Eq. (36) in three dimensions, the following finite-difference approximation for the potential at every nodal point in a 3-D lattice can be obtained:

$$\phi_{i,j,k}^{p+I} = (1-\Omega)\phi_{i,j,k}^{p} + \frac{\Omega\phi_{\text{new}}}{Y_{i,j,k}} \tag{44}$$

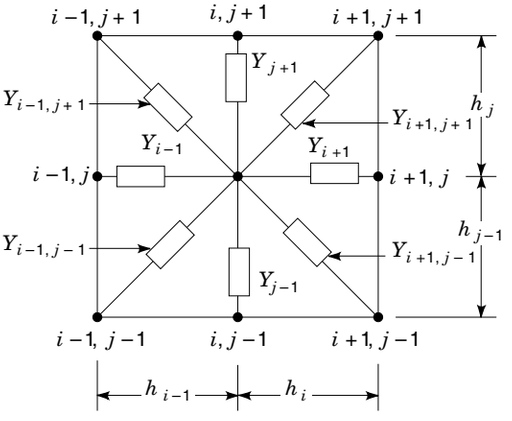


Figure 8. Network analog for 2-D FDM algorithm at grid point i, j for arbitrary anisotropic medium.

where ϕ_{new} is defined as

$$\begin{aligned}\phi_{\text{new}} = {} & \phi^p_{i+1,j,k}\left(Y_{i+1}+Y^A_1\right)+\phi^{p-1}_{i-1,j,k}\left(Y_{i-1}-Y^A_1\right)\\ & +\phi^p_{i,j+1,k}\left(Y_{j+1}+Y^A_2\right)+\phi^{p-1}_{i,j-1,k}\left(Y_{j-1}-Y^A_2\right)\\ & +\phi^p_{i,j,k+1}\left(Y_{k+1}+Y^A_3\right)+\phi^{p-1}_{i,j,k-1}\left(Y_{k-1}-Y^A_3\right)\\ & +Y^A_4\left[\left(\phi^p_{i+1,j+1,k}-\phi^{p-1}_{i-1,j-1,k}\right)-\left(\phi^p_{i-1,j+1,k}+\phi^p_{i+1,j-1,k}\right)\right]\\ & +Y^A_5\left[\left(\phi^p_{i+1,j,k+1}+\phi^{p-1}_{i-1,j,k-1}\right)-\left(\phi^p_{i-1,j,k+1}+\phi^p_{i+1,j,k-1}\right)\right]\\ & +Y^A_6\left[\left(\phi^p_{i,j+1,k+1}+\phi^{p-1}_{i,j-1,k-1}\right)-\left(\phi^p_{i,j+1,k-1}+\phi^p_{i,j-1,k+1}\right)\right]\end{aligned} \tag{45}$$

and

$$Y_{i,j,k} = Y_{i+1}+Y_{i-1}+Y_{j+1}+Y_{j-1}+Y_{k-1}+Y_{k+1}$$

The Y terms appearing in Eqs. (44) and (45) are given by

$$Y_{i-1} = \frac{2}{h_i+h_{i-1}}\left[T_{1,xx}\left(\frac{1}{h_{i-1}}-\frac{2}{h_i+h_{i-1}}\right)+T_{2,xx}\left(\frac{1}{h_{i-1}}+\frac{2}{h_i+h_{i-1}}\right)\right] \tag{46}$$

$$Y_{i+1} = \frac{2}{h_i+h_{i-1}}\left[T_{1,xx}\left(\frac{1}{h_i}+\frac{2}{h_i+h_{i-1}}\right)+T_{2,xx}\left(\frac{1}{h_i}+\frac{2}{h_i+h_{i-1}}\right)\right] \tag{47}$$

$$Y_{j-1} = \frac{2}{h_j+h_{j-1}}\left[T_{1,yy}\left(\frac{1}{h_{j-1}}-\frac{2}{h_j+h_{j-1}}\right)+T_{2,yy}\left(\frac{1}{h_{j-1}}+\frac{2}{h_j+h_{j-1}}\right)\right] \tag{48}$$

$$Y_{j+1} = \frac{2}{h_j+h_{j-1}}\left[T_{1,yy}\left(\frac{1}{h_j}+\frac{2}{h_j+h_{j-1}}\right)+T_{2,yy}\left(\frac{1}{h_j}-\frac{2}{h_j+h_{j-1}}\right)\right] \tag{49}$$

$$Y_{k-1} = \frac{2}{h_k+h_{k-1}}\left[T_{1,zz}\left(\frac{1}{h_{k-1}}-\frac{2}{h_k+h_{k-1}}\right)+T_{2,zz}\left(\frac{1}{h_{k-1}}+\frac{2}{h_k+h_{k-1}}\right)\right] \tag{50}$$

$$Y^A_1 = \left(\frac{2}{h_j+h_{j-1}}\right)\left(\frac{2}{h_i+h_{i-1}}\right)(T_{1,yx}-T_{2,yx})+\left(\frac{2}{h_k+h_{k-1}}\right)\left(\frac{2}{h_i+h_{i-1}}\right)(T_{1,zx}-T_{2,zx}) \tag{51}$$

$$Y^A_2 = \left(\frac{2}{h_j+h_{j-1}}\right)\left(\frac{2}{h_i+h_{i-1}}\right)(T_{1,xy}-T_{2,xy})+\left(\frac{2}{h_k+h_{k-1}}\right)\left(\frac{2}{h_j+h_{j-1}}\right)(T_{1,zy}-T_{2,zy}) \tag{52}$$

$$Y^A_3 = \left(\frac{2}{h_k+h_{k-1}}\right)\left(\frac{2}{h_i+h_{i-1}}\right)(T_{1,xz}-T_{2,xz})+\left(\frac{2}{h_k+h_{k-1}}\right)\left(\frac{2}{h_j+h_{j-1}}\right)(T_{1,yz}-T_{2,yz}) \tag{53}$$

$$Y^A_4 = 2\left(\frac{2}{h_j+h_{j-1}}\right)\left(\frac{2}{h_i+h_{i-1}}\right)\left(\frac{T_{1,xy}+T_{2,xy}}{8}+\frac{T_{1,yx}+T_{2,yx}}{8}\right) \tag{54}$$

$$Y^A_5 = 2\left(\frac{2}{h_k+h_{k-1}}\right)\left(\frac{2}{h_i+h_{i-1}}\right)\left(\frac{T_{1,xz}+T_{2,xz}}{8}+\frac{T_{1,zx}+T_{2,zx}}{8}\right) \tag{55}$$

$$Y^A_6 = 2\left(\frac{2}{h_j+h_{j-1}}\right)\left(\frac{2}{h_k+h_{k-1}}\right)\left(\frac{T_{1,yz}+T_{2,yz}}{8}+\frac{T_{1,zy}+T_{2,zy}}{8}\right) \tag{56}$$

with the T terms having the following forms:

$$T_{1,xx} = \epsilon^{xx}_{i,j-1,k-1}+\epsilon^{xx}_{i,j,k-1}+\epsilon^{xx}_{i,j-1,k}+\epsilon^{xx}_{i,j,k} \tag{57a}$$

$$T_{2,xx} = \epsilon^{xx}_{i-1,j-1,k-1}+\epsilon^{xx}_{i-1,j,k-1}+\epsilon^{xx}_{i-1,j-1,k}+\epsilon^{xx}_{i-1,j,k} \tag{57b}$$

$$T_{1,yy} = \epsilon^{yy}_{i-1,j,k-1}+\epsilon^{yy}_{i,j,k-1}+\epsilon^{yy}_{i-1,j,k}+\epsilon^{yy}_{i,j,k} \tag{58a}$$

$$T_{2,yy} = \epsilon^{yy}_{i-1,j-1,k-1}+\epsilon^{yy}_{i,j-1,k-1}+\epsilon^{yy}_{i-1,j-1,k}+\epsilon^{yy}_{i,j-1,k} \tag{58b}$$

$$T_{1,zz} = \epsilon^{zz}_{i-1,j-1,k}+\epsilon^{zz}_{i,j-1,k}+\epsilon^{zz}_{i-1,j,k}+\epsilon^{zz}_{i,j,k} \tag{59a}$$

$$T_{2,zz} = \epsilon^{zz}_{i-1,j-1,k-1}+\epsilon^{zz}_{i,j-1,k-1}+\epsilon^{zz}_{i-1,j,k-1}+\epsilon^{zz}_{i,j,k-1} \tag{59b}$$

$$T_{1,xy} = \epsilon^{xy}_{i,j-1,k-1}+\epsilon^{xy}_{i,j,k-1}+\epsilon^{xy}_{i,j-1,k}+\epsilon^{xy}_{i,j,k} \tag{60a}$$

$$T_{2,xy} = \epsilon^{xy}_{i-1,j-1,k-1}+\epsilon^{xy}_{i-1,j,k-1}+\epsilon^{xy}_{i-1,j-1,k}+\epsilon^{xy}_{i-1,j,k} \tag{60b}$$

$$T_{1,xz} = \epsilon^{xz}_{i,j-1,k-1}+\epsilon^{xz}_{i,j,k-1}+\epsilon^{xz}_{i,j-1,k}+\epsilon^{xz}_{i,j,k} \tag{61a}$$

$$T_{2,xz} = \epsilon^{xz}_{i-1,j-1,k-1}+\epsilon^{xz}_{i-1,j,k-1}+\epsilon^{xz}_{i-1,j-1,k}+\epsilon^{xz}_{i-1,j,k} \tag{61b}$$

$$T_{1,yx} = \epsilon^{yx}_{i-1,j,k-1}+\epsilon^{yx}_{i,j,k-1}+\epsilon^{yx}_{i-1,j,k}+\epsilon^{yx}_{i,j,k} \tag{62a}$$

$$T_{2,yx} = \epsilon^{yx}_{i-1,j-1,k-1}+\epsilon^{yx}_{i,j,-1,k-1}+\epsilon^{yx}_{i-1,j-1,k}+\epsilon^{yx}_{i,j-1,k} \tag{62b}$$

$$T_{1,yz} = \epsilon^{yz}_{i-1,j,k-1}+\epsilon^{yz}_{i,j,k-1}+\epsilon^{yz}_{i-1,j,k}+\epsilon^{yz}_{i,j,k} \tag{63a}$$

$$T_{2,yz} = \epsilon^{yz}_{i-1,j-1,k-1}+\epsilon^{yz}_{i,j-1,k-1}+\epsilon^{yz}_{i-1,j-1,k}+\epsilon^{yz}_{i,j-1,k} \tag{63b}$$

$$T_{1,zx} = \epsilon^{zx}_{i-1,j-1,k}+\epsilon^{zx}_{i,j-1,k}+\epsilon^{zx}_{i-1,j,k}+\epsilon^{zx}_{i,j,k} \tag{64a}$$

$$T_{2,zx} = \epsilon^{zx}_{i-1,j-1,k-1}+\epsilon^{zx}_{i,j-1,k-1}+\epsilon^{zx}_{i-1,j,k-1}+\epsilon^{zx}_{i,j,k-1} \tag{64b}$$

$$T_{1,zy} = \epsilon^{zy}_{i-1,j-1,k}+\epsilon^{zy}_{i,j-1,k}+\epsilon^{zy}_{i-1,j,k}+\epsilon^{zy}_{i,j,k} \tag{65a}$$

$$T_{2,zy} = \epsilon^{zy}_{i-1,j-1,k-1}+\epsilon^{zy}_{i,j-1,k-1}+\epsilon^{zy}_{i-1,j,k-1}+\epsilon^{zy}_{i,j,k-1} \tag{65b}$$

Note that Eq. (45) has a similar interpretation as its 2-D counterpart Eq. (39). It can also be represented by an equivalent network, whose diagonal terms are shown in Fig. 9. For clarity, the off-diagonal terms, which provide the connections of $\phi_{i,j,k}$ to the voltages at the remaining nodes in Eq. (45), are shown separately in Fig. 10.

Coordinate Transformation Approach. Coordinate transformations can be used to simplify the solution to electrostatic boundary-value problems. Such transformations can reduce the complexity arising from complicated geometry or from the presence of anisotropic materials. In general, these methods utilize coordinate transformation to map complex geometries or material properties into simpler ones, through a specific relationship which links each point in the original and transformed problems, respectively.

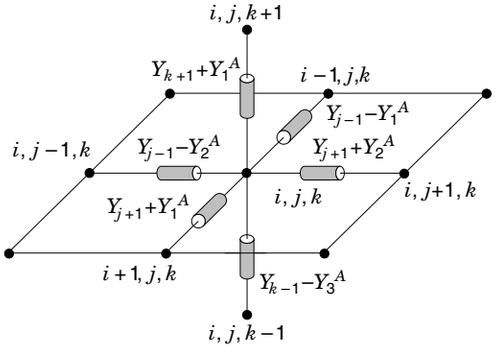


Figure 9. Network analog for 3-D FDM algorithm at grid point i, j for anisotropic dielectric with diagonal permittivity tensor.

One class of coordinate transformations, known as *conformal mapping,* is based on modifying the original complex geometry to one for which an analytic solution is available. This technique requires extensive mathematical expertise in order to identify an appropriate coordinate transformation function. Its applications are limited to a few specific geometrical shapes for which such functions exist. Furthermore, the applications are restricted to two-dimensional problems. Even though this technique can be very powerful, it is usually rather tedious and thus it is considered beyond the scope of this article. The interested reader can refer to Ref. 30, among others, for further details.

The second class of coordinate transformations reduces the complexity of the FDM formulation in problems involving anisotropic materials. As described in the previous section, the discretization of the Laplace equation in anisotropic regions [Eq. (36)] is considerably more complicated than the corresponding procedure for isotropic media [Eq. (7)]. However, it can be shown that a sequence of rotation and scaling transformations can convert any symmetric permittivity tensor into an identity matrix (i.e., free space). As a result, the FDM solution of the Laplace equation in the transformed coordinate system is considerably simplified, since the anisotropic dielectric is eliminated.

To illustrate the concept, this technique will be demonstrated with two-dimensional examples. In 2-D (no z dependence is assumed) the Laplace equation can be written as

$$\nabla \cdot ([\epsilon_r(x, y)]\nabla\phi) = 0 \tag{66}$$

where

$$[\epsilon_r] = \begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} \\ \epsilon_{yx} & \epsilon_{yy} \end{bmatrix} \tag{67}$$

If the principal (crystal or major) axes of the dielectric are aligned with the coordinate system of the geometry, then the off-diagonal terms vanish. Otherwise, $[\epsilon_r]$ is a full symmetric matrix. In this case, any linear coordinate transformation of the form

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = [A] \begin{bmatrix} x \\ y \end{bmatrix} \tag{68}$$

(where $[A]$ is a 2×2 nonsingular matrix of constant coefficients) also transforms the permittivity tensor as follows:

$$[\epsilon'] = [A]^{-1}[\epsilon_r][A] \tag{69}$$

Next, consider the structure shown in Fig. 11(a). It consists of a perfect conductor (metal) embedded in an anisotropic dielectric, all enclosed within a rectangular conducting shell. The field within the rectangular shell must be determined given the potentials on all conductors. In this example, $[\epsilon_r]$ is assumed to be diagonal:

$$[\epsilon_r] = \begin{bmatrix} \epsilon_{xx} & 0 \\ 0 & \epsilon_{yy} \end{bmatrix} \tag{70}$$

By scaling the coordinates with

$$[A] = \begin{bmatrix} 1/\sqrt{\epsilon_{xx}} & 0 \\ 0 & 1/\sqrt{\epsilon_{yy}} \end{bmatrix} \tag{71}$$

the permittivity can be transformed into an identity matrix. The geometry of the structure is deformed as shown in Fig. 11(b), with the corresponding rectangular discretization grid depicted in Fig. 11(c). Note that the locations of the unknown potential variables are marked by white dots, while the conducting boundaries are represented by known potentials and their locations are denoted by black dots. The potential in the transformed boundary-value problem can now be computed by applying the FDM algorithm, which is specialized for free space, since $[\epsilon_r]$ is an identity matrix.

Once the potential is computed everywhere, other quantities of interest, such as the E field and charge, can be calculated next. However, to correctly evaluate the required space derivatives, transformation back to original coordinates is required, as illustrated in Fig. 11(d). Note that in spite of the resulting simplifications, this method is limited to cases where the entire computational space is occupied by a single homogeneous anisotropic dielectric.

In general, when the principal (or major) axes of the permittivity are arbitrarily orientated with respect to the coordinate axes of the geometry, $[\epsilon_r]$ is a full symmetric matrix. In this case, $[\epsilon_r]$ can be diagonalized by an orthonormal coordi-

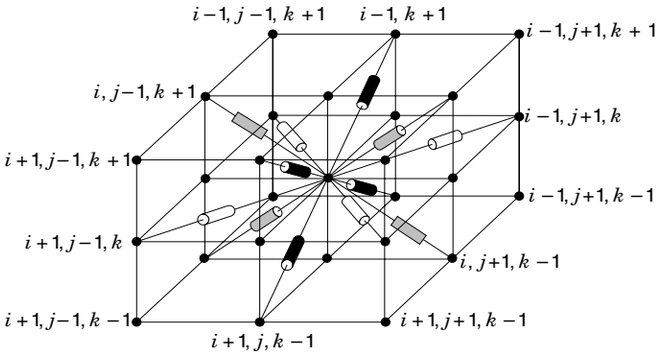


Figure 10. Network analog for 3-D FDM algorithm at grid point i, j for arbitrary anisotropic dielectric.

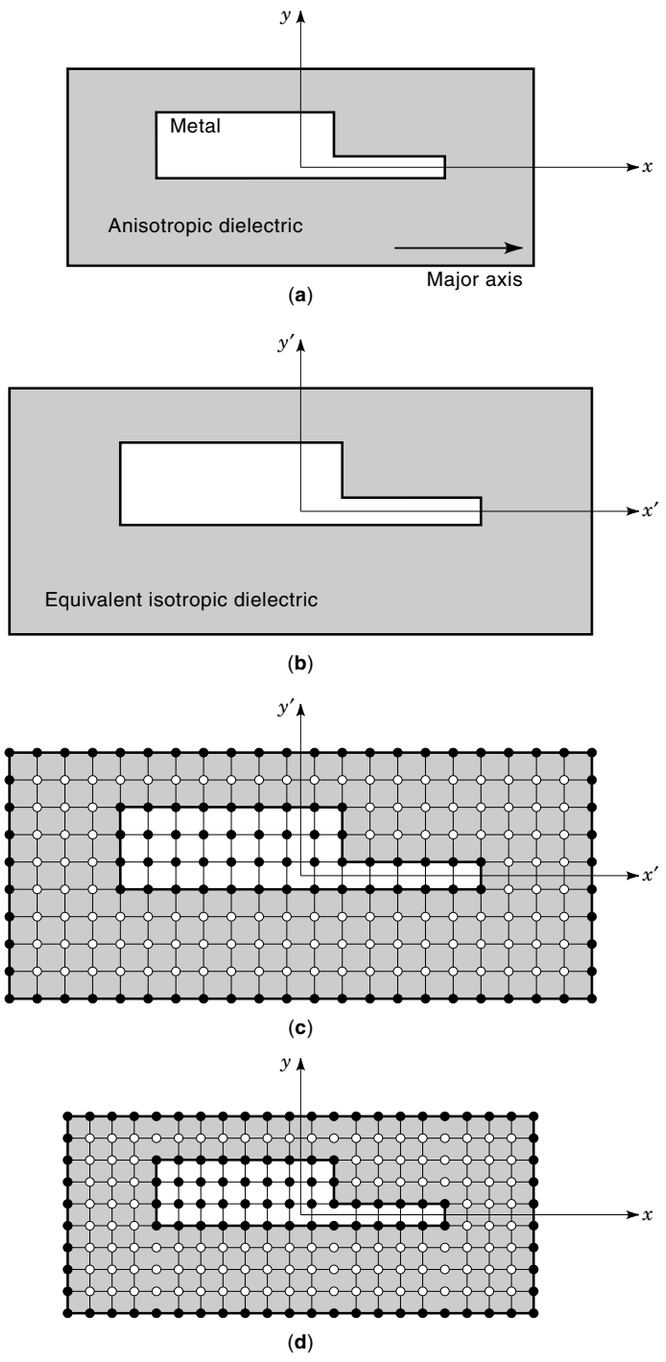


Figure 11. Graphical representation of coordinate transformation for homogeneous anisotropic dielectric with diagonal permittivity tensor.

nate transformation. Specifically, there exists a rotation matrix of the form:

$$[A] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{72}$$

such that the product,

$$[\epsilon'] = [A]^T[\epsilon_r][A] \tag{73}$$

is a diagonal matrix. The angle θ is defined as the angle by which the coordinate system should be rotated to align it with the major axes of the dielectric.

Consider the structure shown in Fig. 12(a), which is enclosed in a metallic shell. However, in this example the nonconducting region of interest includes both free space and an anisotropic dielectric. Furthermore, the major axis of $[\epsilon_r]$ is at 30 degrees with respect to that of the structure. The effect of rotating the coordinates by $\theta = -30$ degrees leads to a geometry shown in Fig. 12(b). In the transformed coordinate system, the major axis of the permittivity is horizontal and $[\epsilon_r]$ is a diagonal matrix. Observe that this transformation does not affect the dielectric properties of the free-space region (or of any other isotropic dielectrics, if present). However, the subsequent scaling operation for transforming the properties of the anisotropic region to free space is not useful. Such transformation also changes the properties of the original free-space region to those exhibiting anisotropic characteristics. Regardless of this limitation, the coordinate rotation alone considerably simplifies the FDM algorithm of Eq. (45) to

$$\begin{aligned} \phi_{\text{new}} = {} & \phi^p_{i+1,j,k}\left(Y_{i+1} + Y^A_1\right) + \phi^{p-1}_{i-1,j,k}\left(Y_{i-1} - Y^A_1\right) \\ & + \phi^p_{i,j+1,k}\left(Y_{j+1} + Y^A_2\right) + \phi^{p-1}_{i,j-1,k}\left(Y_{j-1} - Y^A_2\right) \end{aligned} \tag{74}$$

where all z-dependent (or k) terms have been removed.

Without the rotation, the permittivity is characterized by Eq. (67). Under such conditions, the corresponding FDM update equation includes four additional potential variables, as shown below:

$$\begin{aligned} \phi_{\text{new}} = {} & \phi^p_{i+1,j,k}\left(Y_{i+1} + Y^A_1\right) + \phi^{p-1}_{i-1,j,k}\left(Y_{i-1} - Y^A_1\right) \\ & + \phi^p_{i,j+1,k}\left(Y_{j+1} + Y^A_2\right) + \phi^{p-1}_{i,j-1,k}\left(Y_{j-1} - Y^A_2\right) \\ & + Y^A_4\left[\left(\phi^p_{i+1,j+1,k} - \phi^{p-1}_{i-1,j-1,k}\right)\right. \\ & \left. - \left(\phi^p_{i-1,j+1,k} + \phi^p_{i+1,j-1,k}\right)\right] \end{aligned} \tag{75}$$

The simplification resulting from coordinate rotation in three dimensions is even more significant. In the general case, the full FDM algorithm [Eq. (45)] contains 18 terms, while in the rotated coordinates the new equation has only 6.

Next, a rectangular discretization grid is constructed for the transformed geometry as shown in Fig. 12(c), with the unknown potential represented by white dots and conducting boundaries denoted by black dots. As can be seen, the rotation complicates the assignment (or definition) of the boundary nodes. In general, a finer discretization may be required to approximate the metal boundaries more accurately.

Once the potential field is computed, the transformation back to the original coordinates is performed by applying the inverse rotation $[A]^T$, as illustrated in Fig. 11(d). Note that in the original coordinate system, the grid is rotated and, as such, complicates the computation of electric field. In addition to the required coordinate mapping, this method is also limited to boundary-value problems containing only one type of anisotropic dielectric, though any number of isotropic dielectric regions may be present.

The above examples illustrate that coordinate transformations are beneficial in solving a narrow class of electrostatic problems. Undoubtedly, considerable computational savings can be achieved in the calculation of the potential using FDM.

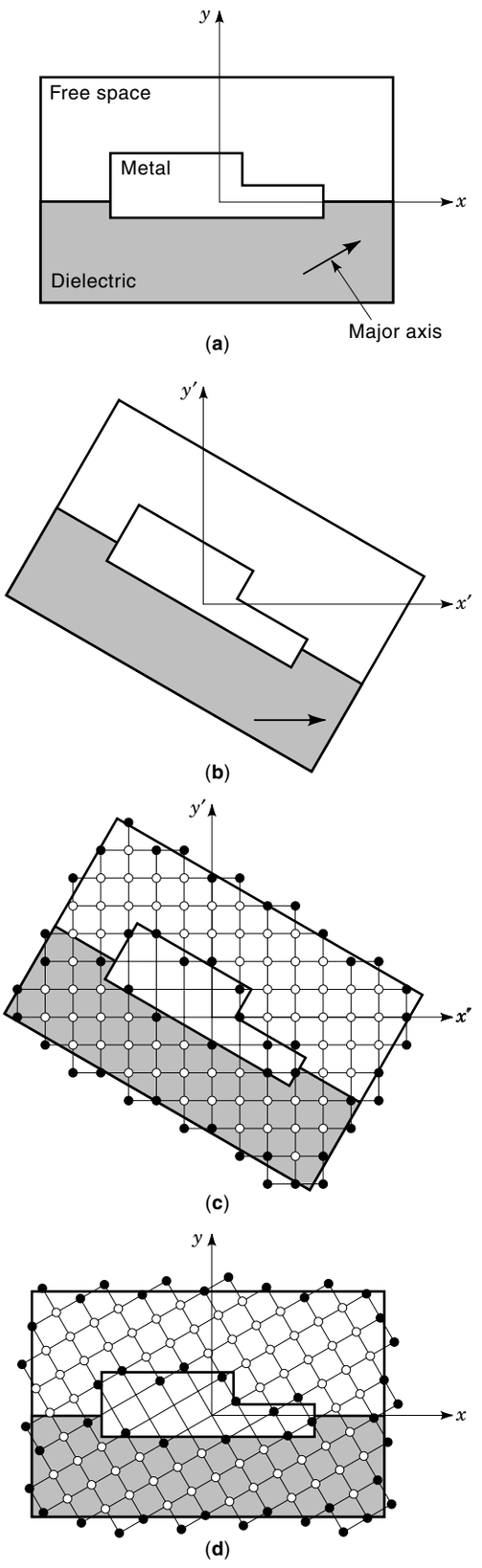


Figure 12. Graphical representation of coordinate transformation for inhomogeneous anisotropic dielectric with diagonal permittivity tensor.

However, the computational overhead associated with the pre- and postprocessing can be significant, since the geometry is usually complicated by such transformations.

SAMPLE NUMERICAL RESULTS

To illustrate the versatility of FDM in solving engineering problems that involve arbitrary geometries and inhomogeneous materials, consider the cross section of a microwave field effect transistor (FET) shown in Fig. 13. Note that this device is composed of many different materials, each of different thickness and cross-sectional profile. The FET is drawn to scale, with the 1 μm thickness of the buffer layer serving as a reference. FDM can be used to calculate the potential and field distribution throughout the entire cross section of the FET. This information can be used by the designer to investigate such effect as material breakdown near the metallic electrodes. In addition, the computed field information can be used to determine the parasitic capacitance matrix of the structure, which can be used to improve the circuit model of this device and is very important in digital circuit design. Finally, it should be noted that the losses associated with the silicon can also be computed using FDM as shown in Eq. (25).

It should be added that in addition to displaying the potential distribution over the cross section of the FET, Fig. 13 also illustrates the implementation of open boundary truncation operators. Since the device is located in an open boundary environment, it was necessary to artificially truncate the computation space (or 2-D grid). Note that, as demonstrated in Ref. 25, only the first-order operator was sufficient to obtain accurate representation of the potential in the vicinity of the electrodes as well as near the boundary truncation surface.

A sample with three-dimensional geometry that can be easily analyzed with the FDM is shown in Fig. 14. The insulator in the multilayer ceramic capacitor is assumed to be anisotropic barium titanate dielectric, which is commonly used in such components. The permittivity tensor is diagonal and its elements are $\epsilon_{xx} = 1540$, $\epsilon_{yy} = 290$, and $\epsilon_{zz} = 1640$. To demonstrate the effect of anisotropy on this passive electrical component, its capacitance was calculated as a function of the misalignment angle between the crystal axes of the insulator and the geometry of the structure (see Fig. 15).

For the misalignment angle (or rotation of axes) in the yz plane, the capacitance of this structure was computed. The results of the computations are plotted in Fig. 16. Note that the capacitance varies considerably with the rotation angle. Such information is invaluable to a designer, since the goal of the design is to maximize the capacitance for the given dimensions of the structure.

The above examples are intended to demonstrate the applicability of FDM to the solution of practical engineering boundary-value problems. FDM has been used extensively in analysis of other practical problems. The interested reader can find additional examples where FDM was used in Refs. 31–37.

SUMMARY

Since the strengths and weaknesses of FDM were mentioned throughout this article, as were the details dealing with the derivation and numerical implementation of this method,

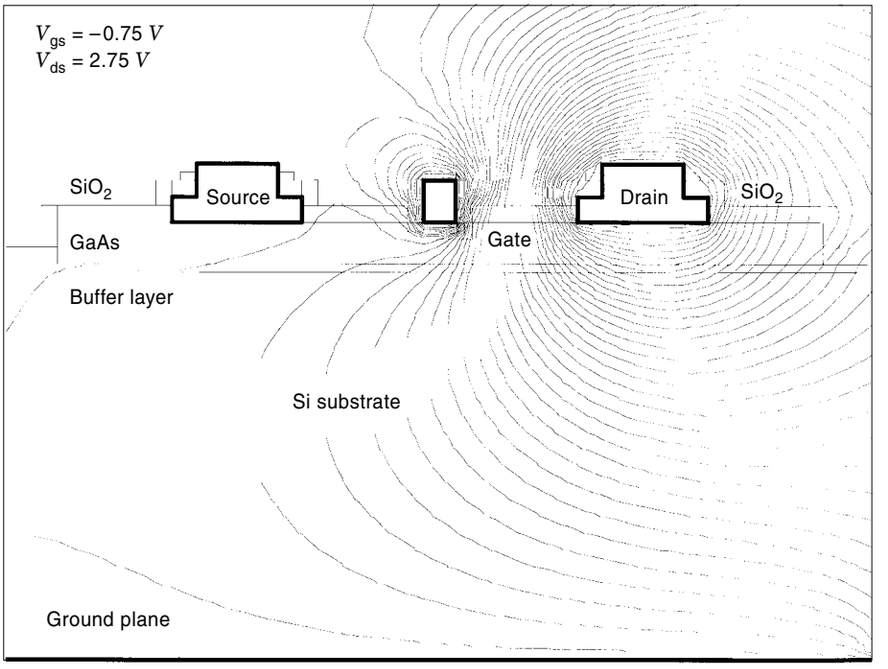


Figure 13. Equipotential map of dc-biased microwave FET. From Computer-aided quasi-static analysis of coplanar transmission lines for microwave integrated circuits using the finite difference method, B. Beker and G. Cokkinides, *Int. J. MIMICAE,* **4** (1): 111–119. Copyright © 1994, Wiley.)

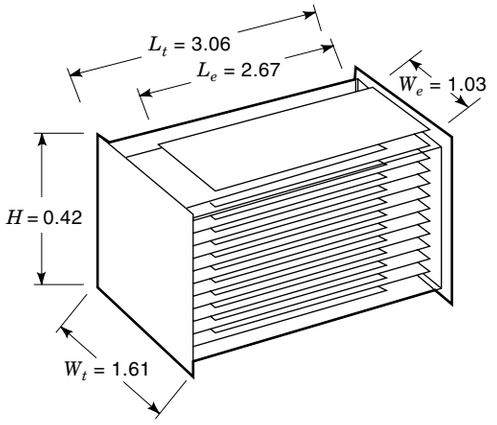


Figure 14. Geometry of a multilayer ceramic chip capacitor. All dimensions are in millimeters.

they need not be repeated. However, the reader should realize that FDM is best suited for boundary-value problems with complex geometries and arbitrary material composition. The complexity of the problem is the primary motivating factor for investing the effort into developing a general-purpose volumetric analysis tool.

ACKNOWLEDGMENTS

The authors wish to express their sincere thanks to many members of the technical staff at AVX Corporation for initiating, supporting, and critiquing the development and implementation of many concepts presented in this article, as well as for suggesting practical uses of FDM. Many thanks also go to Dr. Deepak Jatkar for his help in extending FDM to general anisotropic materials.

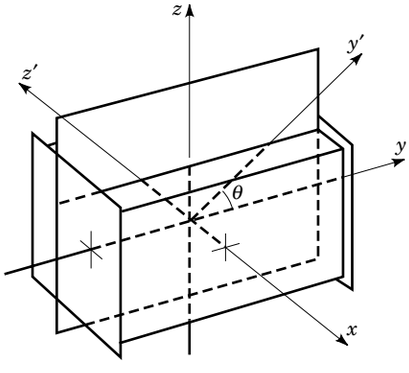


Figure 15. Definition of rotation angle for anisotropic insulator.

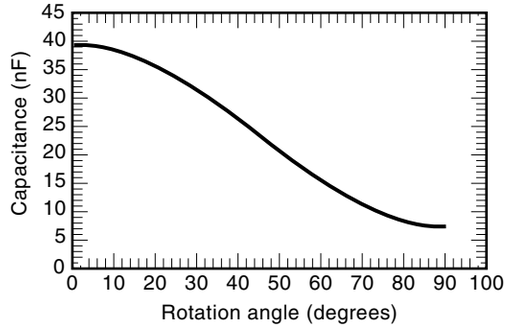


Figure 16. Capacitance of multilayer chip capacitor as a function of rotation angle of the insulator.

BIBLIOGRAPHY

1. H. Liebmann, *Sitzungsber. Bayer. Akad. München,* 385, 1918.
2. R. V. Southwell, *Relaxation Methods in Engineering Science,* Oxford: Clarendon Press, 1940.
3. R. V. Southwell, *Relaxation Methods in Theoretical Physics,* Oxford: Clarendon Press, 1946.
4. H. E. Green, The numerical solution of some important transmission line problems, *IEEE Trans. Microw. Theory Tech.,* **17** (9): 676–692, 1969.
5. F. Sandy and J. Sage, Use of finite difference approximation to partial differential equations for problems having boundaries at infinity, *IEEE Trans. Microw. Theory Tech.,* **19** (5): 484–486, 1975.
6. G. Liebmann, Solution to partial differential equations with resistance network analogue, *Br. J. Appl. Phys.,* **1** (4): 92–103, 1950.
7. K. S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. Antennas Propag.,* **14** (3): 302–307, 1966.
8. A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method,* Boston, MA: Artech House, 1995.
9. N. N. Rao, *Elements of Engineering Electromagnetics,* 4th ed., Englewood Cliffs, NJ: Prentice-Hall, 1994.
10. S. R. H. Hoole and P. R. P. Hoole, *A Modern Short Course in Engineering Electromagnetics,* New York: Oxford University Press, 1996.
11. T. G. Jurgens, A. Taflove, K. Umashankar, and T. G. Moore, Finite-difference time-domain modeling of curved surfaces. *IEEE Trans. Antennas Propag.,* **40** (4): 357–366, 1992.
12. M. Naghed and I. Wolf, Equivalent capacitances of coplanar waveguide discontinuities and interdigitated capacitors using three-dimensional finite difference method, *IEEE Trans. Microw. Theory Tech.,* **38** (12): 1808–1815, 1990.
13. M. F. Iskander, *Electromagnetic Fields & Waves,* Englewood Cliffs, NJ: Prentice-Hall, 1992, Section 4.8.
14. R. Haberman, *Elementary Applied Partial Differential Equations with Fourier Series and Boundary Value Problems,* Englewood Cliffs, NJ: Prentice-Hall, 1983, Chapter 13.
15. L. Lapidus and G. H. Pinder, *Numerical Solutions of Partial Differential Equations in Science and Engineering,* New York: Wiley, 1982.
16. W. F. Tinney and J. W. Walker, Direct solution of sparse network equations by optimally ordered triangular factorization, *IEEE Proc.,* **55** (11), 1967.
17. W. T. Press, B. P. Flanney, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing,* 2nd ed., Cambridge: Cambridge University Press, 1992, Section 2.7.
18. Y. Saad, *Iterative Methods for Sparse Linear Systems,* Boston: PWS Publishing Co., 1996.
19. G. H. Golub and C. F. Van Loan, *Matrix Computations,* 3rd ed., Baltimore: Johns Hopkins University Press, 1996, Chapter 10.
20. R. E. Philips and F. W. Schmidt, Multigrid techniques for the numerical solution of the diffusion equation, *Num. Heat Transfer,* **7**: 251–268, 1984.
21. J. H. Smith, K. M. Steer, T. F. Miller, and S. J. Fonash, Numerical modeling of two-dimensional device structures using Brandt's multilevel acceleration scheme: Application to Poisson's equation, *IEEE Trans. Comput.-Aided Des. Integr. Circuit Syst.,* **10** (6): 822–824, 1991.
22. A. Kherib, A. B. Kouki, and R. Mittra, Higher order asymptotic absorbing boundary conditions for the finite element modeling of two-dimensional transmission line structures, *IEEE Trans. Microw. Theory Tech.,* **38** (10): 1433–1438, 1990.
23. A. Kherib, A. B. Kouki, and R. Mittra, Asymptotic absorbing boundary conditions for the finite element analysis of three-dimensional transmission line discontinuities, *IEEE Trans. Microw. Theory Tech.,* **38** (10): 1427–1432, 1990.
24. R. K. Gordon and H. Fook, A finite difference approach that employs an asymptotic boundary condition on a rectangular outer boundary for modeling two-dimensional transmission line structures. *IEEE Trans. Microw. Theory Tech.,* **41** (8): 1280–1286, 1993.
25. B. Beker and G. Cokkinides, Computer-aided analysis of coplanar transmission lines for monolithic integrated circuits using the finite difference method, *Int. J. MIMICAE,* **4** (1): 111–119, 1994.
26. T. L. Simpson, Open-boundary relaxation, *Microw. Opt. Technol. Lett.,* **5** (12): 627–633, 1992.
27. D. Jatkar, *Numerical Analysis of Second Order Effects in SAW Filters,* Ph.D. Dissertation, University of South Carolina, Columbia, SC, 1996, Chapter 3.
28. S. R. Hoole, *Computer-Aided Analysis and Design of Electromagnetic Devices,* New York: Elsevier, 1989.
29. V. K. Tripathi and R. J. Bucolo, A simple network analog approach for the quasi-static characteristics of general, lossy, anisotropic, layered structures, *IEEE Trans. Microw. Theory Tech.,* **33** (12): 1458–1464, 1985.
30. R. E. Collin, *Foundations for Microwave Engineering,* 2nd ed., New York: McGraw-Hill, 1992, Appendix III.
31. B. Beker, G. Cokkinides, and A. Templeton, Analysis of microwave capacitors and IC packages, *IEEE Trans. Microw. Theory Tech.,* **42** (9): 1759–1764, 1994.
32. D. Jatkar and B. Beker, FDM analysis of multilayer-multiconductor structures with applications to PCBs, *IEEE Trans. Comp. Pack. Manuf. Technol.,* **18** (3): 532–536, 1995.
33. D. Jatkar and B. Beker, Effects of package parasitics on the performance of SAW filters, *IEEE Trans. Ultrason. Ferroelect. Freq. Control,* **43** (6): 1187–1194, 1996.
34. G. Cokkinides, B. Beker, and A. Templeton, Direct computation of capacitance in integrated passive components containing floating conductors, *IEEE Trans. Comp. Pack. Manuf. Technol.,* **20** (2): 123–128, 1997.
35. B. Beker, G. Cokkinides, and A. Agrawal, Electrical modeling of CBGA packages, *Proc. IEEE Electron. Comp. Technol. Conf.,* 251–254, 1995.
36. G. Cokkinides, B. Beker, and A. Templeton, Cross-talk analysis using the floating conductor model, *Proc. ISHM-96, Int. Microelectron. Soc. Symp.,* 511–516, 1996.
37. B. Beker and T. Hirsch, Numerical and experimental modeling of high speed cables and interconnects, *Proc. IEEE Electron. Comp. Technol. Conf.,* 898–904, 1997.

BENJAMIN BEKER
GEORGE COKKINIDES
MYUNG JIN KONG
University of South Carolina

/ /

Calculus
Standard Article

Keith E. Holbert[1] and A. Sharif Heger[2]
[1]Arizona State University, Tempe, AZ
[2]Los Alamos National Laboratory, Los Alamos, NM

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2404.pub2
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (1837K)

Abstract

The sections in this article are

History

Notation and Definitions

Differential Calculus

Integral Calculus

Additional Topics in Calculus

| | |

Copyright © 1999-2008 All Rights Reserved.

CALCULUS

Calculus has its foundation in taking a limit. For example, one can obtain the area of a circle as the limit of the areas of regular inscribed polygons as the number of sides increases without bound. This example can be extended to determining the perimeter of a circle or the volume of a sphere. Similarly in algebra, this limiting approach is used to seek the value of a repeating decimal. In plane analytic geometry, this concept is used to explain tangents to curves.

The two fundamental operations in calculus are differentiation and integration. Both of these fundamental tools have played an important role in the development of many scientific theories. The *Fundamental Theorem of Calculus* provides the connection between differentiation and integration and was discovered independently by Sir Isaac Newton and Baron Gottfried Wilhelm Leibniz.

In the remainder of this article, a brief history of development of calculus is presented. This introduction is followed by a discussion of the principle of differentiation. A similar discussion on integrals is presented next. Other relevant topics important to electrical engineers are also presented. Each section is augmented with examples using classic problems in engineering to illustrate the practical use of calculus.

HISTORY

The methods used by the Greeks for determining the area of a circle and a segment of a parabola, as well as the volumes of the cylinder, cone, and sphere, were in principle akin to the method of integration. During the first half of the 17th century, methods of more or less limited scope began to appear among mathematicians for constructing tangents, determining maxima and minima, and finding areas and volumes. In particular, Fermat, Pascal, Roberval, Descartes, and Huygens discussed methods of drawing tangents to particular curves and finding areas bounded by certain special curves. Each problem was considered by itself, and few general rules were developed. The essential ideas of the derivative and definite integral were, however, beginning to be formulated. With this mathematical heritage, Newton and Leibniz, working independently of each other during the latter half of the 17th century, defined the concepts of derivatives and integrals. Leibniz used the notation dy/dx for the derivative and introduced the integration symbol $\int$. The portion of mathematics that includes only topics that depend on calculus is called *analysis*. Included in this category are differential and integral equations, theory of functions of real and complex variables, and algebraic and elliptic functions.

Calculus has helped the development of other fields of science and engineering. Geometry and number theory make use of this powerful tool. In the development of modern physics and engineering, the concepts developed in calculus and its extensions are continually utilized. For example, in dealing with electricity, the current, I, through a circuit due to the flow of charge, Q, is expressed as $I \equiv dQ/dt$; the voltage, v, across an inductor, L, is defined as $v \equiv L\, dI/dt$; and the voltage through a capacitor, C, is defined as $v \equiv (1/C) \int I\, dt$.

NOTATION AND DEFINITIONS

Within this article, the parameters u, v, and w represent functions of independent variable x, while other alphabetic letters represent fixed real numbers. A variable in boldface type denotes a vector quantity.

Limits

Of fundamental importance to the field of calculus is the concept of the *limit*, which represents the value of an entity under a given extreme condition. For instance, a limit can be used to define the natural exponential function, e:

$$\lim_{h\to 0}(1+h)^{1/h} = e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n \tag{1}$$

Given here are rules for computing limits. The limit of a constant is the constant:

$$\lim_{x\to a} c = c \tag{2}$$

The limit of a function scaled by a constant is the constant times the limit of the function:

$$\lim_{x\to a} c\, f(x) = c \lim_{x\to a} f(x) \tag{3}$$

The limit of a sum (or difference) is the sum (or difference) of the limits:

$$\lim_{x\to a}[f(x) \pm g(x)] = \lim_{x\to a} f(x) \pm \lim_{x\to a} g(x) \tag{4}$$

The limit of a product is the product of the limits:

$$\lim_{x\to a} f(x)g(x) = [\lim_{x\to a} f(x)][\lim_{x\to a} g(x)] \tag{5}$$

The limit of a quotient is the quotient of the limits, if the denominator does not equal zero:

$$\lim_{x\to a} \frac{f(x)}{g(x)} = \frac{\lim\limits_{x\to a} f(x)}{\lim\limits_{x\to a} g(x)} \tag{6}$$

The limit of a function raised to a positive integer power, n, is

$$\lim_{x\to a}[f(x)]^n = [\lim_{x\to a} f(x)]^n \tag{7}$$

The limit of a polynomial function $f(x) = b_n x^n + b_{n-1}x^{n-1} + \cdots + b_1 x + b_0$ is

$$\lim_{x\to a} f(x) = b_n a^n + b_{n-1}a^{n-1} + \cdots + b_1 a + b_0 \tag{8}$$

The limits of a function are sometimes broken into left-hand and right-hand limits. A function $f(t)$ has a limit at a if and only if the right-hand and left-hand limits at a exist and are equal.

L'Hôpital's Rule. If $f(x)/g(x)$ has the indeterminate form 0/0 or ∞/∞ at $x = a$, then

$$\lim_{x\to a} \frac{f(x)}{g(x)} = \lim_{x\to a} \frac{f'(x)}{g'(x)} \tag{9}$$

provided that the limit exists or becomes infinite.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 2007 John Wiley & Sons, Inc.

Limits Example. A common application using limits is the initial and final value theorems. Consider a time function, $f(t) = 5e^{-2t}$, whose transformation to the Laplacian domain is

$$F(s) = \frac{5}{s+2}$$

The *final value* may be obtained from

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s) = \lim_{s\to 0} \frac{5s}{s+2} = \frac{0}{2} = 0$$

Likewise, the *initial value* is determined from

$$\lim_{t\to 0} f(t) = \lim_{s\to\infty} sF(s) = \lim_{s\to\infty} \frac{5s}{s+2} = \frac{\infty}{\infty}$$

Hence, L'Hôpital's rule must be used to find its initial value:

$$\lim_{t\to 0} f(t) = \lim_{s\to\infty} \frac{\dfrac{d}{ds}(5s)}{\dfrac{d}{ds}(s+2)} = \lim_{s\to\infty} \frac{5}{1} = 5$$

Continuity

A function $y = f(x)$ is continuous at $x = a$ if and only if all three of the following conditions are satisfied:

1. $f(a)$ exists where a is in the domain of $f(x)$;
2. $\lim_{x\to a} f(x)$ exists; and
3. $\lim_{x\to a} f(x) = f(a)$.

If any of these three conditions fails to hold, then $f(x)$ is *discontinuous* at a. If $f(x)$ is continuous at every point of its domain, $f(x)$ is said to be a *continuous function.* The sine and cosine are examples of continuous functions.

DIFFERENTIAL CALCULUS

Derivative

If y is a single-valued function of x, $y = f(x)$, the *derivative* of y with respect to x is defined to be

$$\frac{dy}{dx} = \lim_{\Delta x\to 0}\left[\frac{f(x+\Delta x) - f(x)}{\Delta x}\right] \tag{10}$$

This quantity is often written as $dy/dx = \lim_{\Delta x\to 0} (\Delta y/\Delta x)$, where Δx is an arbitrary increment of x and $\Delta y = f(x + \Delta x) - f(x)$. The derivative of a function, $y = f(x)$, may be represented in several different ways:

$$\frac{dy}{dx} = D_x y = y' = f'(x) \tag{11}$$

Likewise, a second derivative can be denoted by

$$\frac{d^2y}{dx^2} = D_x^2 y = y'' = f''(x) \tag{12}$$

The symbol D_x is referred to as the *differential operator.* Inverse functions are denoted as $f^{-1}(x)$. Therefore,

$$f^{-1}(y) = f^{-1}[f(x)] = x \tag{13}$$

This operation should not be confused with the reciprocal of a function; that is,

$$[f(x)]^{-1} = 1/f(x) \tag{14}$$

It is important to note that dy/dx is not a quotient. It is a number that is approached by the quotient $\Delta y/\Delta x$ in the limit. The symbols dy and dx, as they appear in dy/dx, have no meaning by themselves. The term dy/dx represents the limit of $\Delta y/\Delta x$.

The *differential* of y for a given value of x is defined as

$$dy = d[f(x)] = \frac{dy}{dx}dx = \frac{d[f(x)]}{dx}dx = f'(x)\,dx \tag{15}$$

Each derivative expression has a differential formula associated with it. For example, the *chain rule*

$$\frac{d}{dx}(uv) = u\frac{dv}{dx} + v\frac{du}{dx} \tag{16}$$

has an equivalent differential formula:

$$d(uv) = u\,dv + v\,du \tag{17}$$

Application. A basic application of the first derivative is the calculation of speed $v(t)$ and acceleration $a(t)$ from a position function $s(t)$:

$$v(t) = \frac{d}{dt}s(t) = s'(t)$$

$$a(t) = \frac{d}{dt}v(t) = v'(t)$$

$$= \frac{d}{dt}\left(\frac{d}{dt}s(t)\right) = \frac{d^2}{dt^2}s(t) = s''(t)$$

This latter expression illustrates the concept of higher-order derivatives—in this case, the second derivative. The derivative is important in many applications—for example, determining the tangents to curves and finding the maxima and minima of a given function.

Tangents

The concept of derivative is best illustrated by considering the construction of a tangent to a curve. Consider a parabola that is represented by the equation $y = x^2$ as shown in Fig. 1. Let Q be any point on the parabola, distinct from another point P. The line that joins Q and P is a secant to the parabola. As Q approaches P, the secant rotates about P. In the limit, as Q is infinitesimally near P, without attaining it, the secant approaches the line that touches the parabola at P without cutting across it. This line is *tangent* to the parabola at point P.

The angle between the secant and the x-axis is the *inclination angle,* θ. The *slope* of a line is defined as the trigonometric tangent of the inclination of the line. To determine the slope of the tangent, it must be noted that in the limit, as Q approaches P, the inclination angle of the secant approaches that of the slope at P. If lines PM and QM are perpendicular to each other, then the slope of PQ is QM/PM.

Let P have coordinates (x, y). As noted earlier, Q is any point on the parabola with coordinates $(x + \Delta x, y + \Delta y)$,

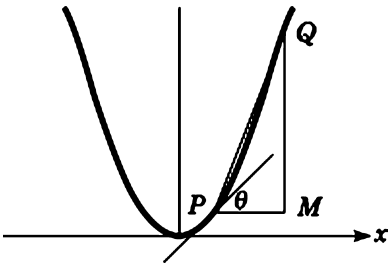


Figure 1. The tangent to the curve at P is the secant PQ in the limit as Q approaches P. The slope of the tangent at this point is the trigonometric tangent of θ as defined by the ratio of QM and PM.

where Δx and Δy equal PM and QM, respectively. Therefore, the slope of PQ is $\Delta y/\Delta x$. The slope of the tangent at P is then the value of this ratio as Q approaches P—that is, as Δx and Δy approach zero. Using the equation of the parabola, $y = x^2$ we obtain:

$$\begin{aligned} y + \Delta y &= (x + \Delta x)^2 \\ &= x^2 + 2x\,\Delta x + (\Delta x)^2. \end{aligned}$$

By definition, $y = x^2$, therefore:

$$\Delta y = 2x\,\Delta x + (\Delta x)^2, \quad \text{or} \quad \Delta y/\Delta x = 2x + \Delta x.$$

For the parabola, as Δx approaches zero, $\Delta y/\Delta x$, the slope of the tangent at P approaches $2x$.

To generalize, consider any function of x, say $y = f(x)$. For the points P and Q on y, the limit of $\Delta y/\Delta x$, as Q approaches P is the *derivative* of $f(x)$, evaluated at point P. The expression dy/dx represents the derivative of the function $y = f(x)$ for any value of x. As was demonstrated in the previous paragraphs, when $y = x^2$ we have $dy/dx = 2x$.

Since each value of x corresponds to a definite value of dy/dx, the derivative of y is also a function of x. The process of finding the derivative of a function is called *differentiation,* as was demonstrated for $y = x^2$. It is important to point out that there are classes of functions for which derivatives do not exist. For example, in the limit as Δx approaches zero, the function's value may either become infinite or oscillate without reaching a limit. In particular, the function $f(x) = |x|$ is not differentiable at $x = 0$ since the right-hand limit (which is 1) does not equal the left-hand limit (which is -1).

Partial Derivatives

Extension of differentiation to multivariable functions is the important field of *partial differential equations.* Applications of this type involve surfaces and finding the maxima and minima of these functions. Selected operations specific to partial differential equations are listed below. The reader is referred to calculus texts for a more extensive discussion on this topic.

Consider a function of the form $z = f(x, y)$. The first partial derivatives of f with respect to x and y, where y and x are held constant, respectively, can be defined as

$$\begin{aligned} \frac{\partial f}{\partial x} &= \lim_{\Delta x \to 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \\ \frac{\partial f}{\partial y} &= \lim_{\Delta y \to 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \end{aligned} \tag{18}$$

The function has three different second partial derivatives:

$$\begin{aligned} \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) &= \frac{\partial^2 f}{\partial x^2} \\ \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial y}\right) &= \frac{\partial^2 f}{\partial y^2} \\ \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) &= \frac{\partial^2 f}{\partial x\,\partial y} \end{aligned} \tag{19}$$

If the function and its partial derivatives are continuous, then the order of differentiation is immaterial for the mixed derivatives and they satisfy the following relationship:

$$\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) = \frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right) \tag{20}$$

Mean Value Theorem

The *Mean Value Theorem* states that if $f(x)$ is defined and continuous on the closed interval $[a,b]$ and differentiable on the open interval (a,b), then there is at least one number c in (a,b) (that is, $a < c < b$) such that

$$f\prime(c) = \frac{f(b) - f(a)}{b - a} \tag{21}$$

For a continuous function $f(x,y,z)$ with continuous partial derivatives, the mean value theorem is

$$\begin{aligned} &f(x_0 + h,\, y_0 + k,\, z_0 + \ell) - f(x_0, y_0, z_0) \\ &\quad = h\frac{\partial f}{\partial x} + k\frac{\partial f}{\partial y} + \ell\frac{\partial f}{\partial z} \end{aligned} \tag{22}$$

Maxima and Minima

Consider a function $y = f(x)$ that has a derivative for every x in a given range. At a point where y reaches a maximum or a minimum, the slope of the tangent to the function is zero. Because the first derivative of a function represents the slope of the function at any point, the second derivative represents the rate of change of the slope. Hence, a positive second derivative indicates an increasing slope, whereas a negative second derivative denotes a decreasing slope. The *concavity* of a function is determined using the second derivative of the function:

If $f''(x) > 0$, then the function is concave upward.
If $f''(x) < 0$, the function is concave downward (convex).

A *point of inflection* denotes the location where curvature of the function changes from convex to concave, and the second derivative of the function is zero. Maxima, minima, and points of inflection are also known as *critical points* of a function. The derivative tests for critical points are listed in Table 1.

For all continuous functions, a maximum or minimum is located where the first derivative equals zero, and a point

Table 1. Conditions for Existence of Critical Points of a Function

First Derivative	Second Derivative	Critical Point
Zero	Negative	Maximum (local/global)
Zero	Positive	Minimum (local/global)
Any value	Zero	Probably an inflection point

of inflection is located where the second derivative equals zero. The converse of these statements is not true. For example, a straight horizontal line has a zero slope at all points but this does not indicate a critical point. Also, any linear function ($y = mx + b$) has a zero-valued second derivative, but this does not indicate points of inflection. Table 2 shows the necessary and sufficient conditions for existence of the maximum and minimum points of the function $z = f(x, y)$ using partial derivatives.

Critical Points Example. Consider the use of calculus to find the critical points of an alternating-current (ac) voltage source. Without specific knowledge of the cosine function, the critical points are found where the first derivative is zero; the second derivative is then used to classify the nature of these points. The voltage and its first and second derivatives are

$$v(t) = V_M \cos(\omega t + \theta)$$
$$v'(t) = \frac{d}{dt}v(t) = -V_M\omega \sin(\omega t + \theta)$$
$$v''(t) = \frac{d^2}{dt^2}v(t) = -V_M\omega^2 \cos(\omega t + \theta)$$

The critical points—that is, where $v'(t) = 0$—are located at $t = (n\pi - \theta)/\omega$, where n is an integer. Substitution of these values of t into the second derivative finds two results:

$$v''[t = (n\pi - \theta)/\omega] = -V_M\omega^2 \cos(n\pi) = \begin{cases} V_M\omega^2 & n \text{ odd} \\ -V_M\omega^2 & n \text{ even} \end{cases}$$

Hence, maxima exist at even values of n and minima at odd n values.

The points of inflection occur where the second derivative is zero, $v''(t) = 0$, specifically here for $t = [(2n + 1)\pi/2 - \theta]/\omega$. These points of inflection identify concavity changes. Regions of specific concavity behavior can be ascertained using $v''(t)$, namely,

$$v''\left(\frac{(2n+1)\pi/2 - \theta}{\omega} < t < \frac{(2n+3)\pi/2 - \theta}{\omega}\right)$$
$$\begin{cases} < 0 & n \text{ odd} \Rightarrow \text{concave down} \\ > 0 & n \text{ even} \Rightarrow \text{concave up} \end{cases}$$

These results are shown in Fig. 2.

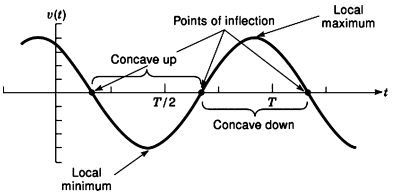


Figure 2. Critical points for an ac voltage source, $v(t) = V_M \cos(\omega t + \theta)$. The maxima and minima are located where $v'(t) = 0$. The points of inflection occur at $v''(t) = 0$ where the concavity of the curve changes.

Differentiation Rules

The following formulas represent the fundamental rules of differentiation. The derivatives of elaborate functions can be systematically evaluated using these rules. All arguments in trigonometric functions are measured in radians, and all inverse trigonometric and hyperbolic functions represent principal values.

Constants. The derivative of a constant is zero:

$$\frac{d}{dx}(a) = 0 \tag{23}$$

Scaling. If u is multiplied by a constant b, so is its derivative:

$$\frac{d}{dx}(bu) = b\frac{du}{dx} \tag{24}$$

Linearity. The derivative of the sum or difference of two or more functions is the sum or difference of the derivatives of the functions:

$$\frac{d}{dx}(u \pm v \pm \cdots) = \frac{du}{dx} \pm \frac{dv}{dx} \pm \cdots \tag{25}$$

Product Rule. The derivative of the product of two functions is

$$\frac{d}{dx}(uv) = u\frac{dv}{dx} + v\frac{du}{dx} \tag{26}$$

For three functions the product rule is

$$\frac{d}{dx}(uvw) = uv\frac{dw}{dx} + uw\frac{dv}{dx} + vw\frac{du}{dx} \tag{27}$$

which can be generalized to the product of more functions.

Quotient Rule. The derivative of the ratio of two functions can be expressed as

$$\frac{d}{dx}\left(\frac{u}{v}\right) = \frac{1}{v^2}\left(v\frac{du}{dx} - u\frac{dv}{dx}\right) \tag{28}$$

Chain Rule. Let y be a function of u, which in turn depends on x; then

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx} \tag{29}$$

Table 2. The Necessary and Sufficient Conditions for Existence of the Maximum and Minimum Points of the Function $z = f(x, y)$ Based on Its Partial Derivatives

Partial Derivatives	Conditions for Maximum	Conditions for Minimum
$\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$	Both zero	Both zero
$\frac{\partial^2 z}{\partial x^2}\frac{\partial^2 z}{\partial y^2} - \left(\frac{\partial^2 z}{\partial x\,\partial y}\right)^2$	Positive	Positive
$\frac{\partial^2 z}{\partial x^2}$ and $\frac{\partial^2 z}{\partial y^2}$	Both negative	Both positive

Given $w = f(u, v)$, $u = g(x, y)$, and $v = h(x, y)$, the chain rule for partial derivatives may be applied as

$$\frac{\partial w}{\partial x} = \frac{\partial w}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial w}{\partial v}\frac{\partial v}{\partial x}$$
$$\frac{\partial w}{\partial y} = \frac{\partial w}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial w}{\partial v}\frac{\partial v}{\partial y} \tag{30}$$

Derivative of Integrals. Given t as an independent variable, we obtain

$$\frac{d}{dt}\int_a^t y(x)\,dx = y(t) \tag{31}$$

Power Rule.

$$\frac{d}{du}u^n = nu^{n-1}\frac{du}{dx} \tag{32}$$

The derivatives of a few selected functions appear in Table 3.

Differentiation Example. A classic network problem requiring differential calculus is the determination of an analytical expression for the load resistance that results in the maximum power transfer in a direct-current (dc) circuit. Consider a reduced circuit consisting of a voltage source, v, in series with a Thévenin equivalent resistance, R_{Th}, and the load resistance, R_{L}. The power delivered to the load is

$$P = i^2 R_{\text{L}} = \left(\frac{v}{R_{\text{Th}} + R_{\text{L}}}\right)^2 R_{\text{L}}$$

A maximum/minimum for P will occur where its derivative with respect to R_{L} is zero; that is, $dP/dR_L = 0$. To determine the derivative, the quotient (or product), power, and chain rules along with the scaling property are utilized:

$$\begin{aligned}
\frac{dP}{dR_L} &= \frac{d}{dR_L}\left[v^2\frac{R_L}{(R_{Th}+R_L)^2}\right] \\
&= \frac{v^2}{(R_{Th}+R_L)^4}\left[(R_{Th}+R_L)^2\frac{d\,R_L}{dR_L} - R_L\frac{d(R_{Th}+R_L)^2}{dR_L}\right] \\
&= \frac{v^2}{(R_{Th}+R_L)^4}\left[(R_{Th}+R_L)^2(1) - R_L\,2(R_{Th}+R_L)\frac{d(R_{Th}+R_L)}{dR_L}\right] \\
&= \frac{v^2}{(R_{Th}+R_L)^3}[(R_{Th}+R_L) - R_L\,2] = \frac{v^2(R_{Th}-R_L)}{(R_{Th}+R_L)^3}.
\end{aligned}$$

Setting this last expression equal to zero yields the classic solution of $R_{\text{L}} = R_{\text{Th}}$.

Mathematically speaking at this point, it is indeterminate as to whether this value of R_L provides the minimum or maximum power transfer. To verify that this solution is indeed the maximum, the second derivative of the power with respect to the load resistance at the point $R_{\text{L}} = R_{\text{Th}}$ is calculated:

$$\frac{d^2P}{dR_{\text{L}}^2} = \frac{d}{dR_{\text{L}}}\left(\frac{dP}{dR_{\text{L}}}\right) = \frac{d}{dR_{\text{L}}}\left[\frac{v^2(R_{\text{Th}} - R_{\text{L}})}{(R_{\text{Th}} + R_{\text{L}})^3}\right]$$

The product (versus quotient) rule is used here to broaden the scope of this example:

$$\begin{aligned}
\frac{d^2P}{dR_{\text{L}}^2} &= v^2\frac{d}{dR_{\text{L}}}\left[(R_{\text{Th}} - R_{\text{L}})(R_{\text{Th}} + R_{\text{L}})^{-3}\right] \\
&= v^2\left[(R_{\text{Th}} - R_{\text{L}})\frac{d}{dR_{\text{L}}}(R_{\text{Th}} + R_{\text{L}})^{-3}\right. \\
&\quad \left. + (R_{\text{Th}} + R_{\text{L}})^{-3}\frac{d}{dR_{\text{L}}}(R_{\text{Th}} - R_{\text{L}})\right] \\
&= v^2\left[(R_{\text{Th}} - R_{\text{L}})(-3)(R_{\text{Th}} + R_{\text{L}})^{-4} + (R_{\text{Th}} + R_{\text{L}})^{-3}(-1)\right] \\
&= \frac{v^2}{(R_{\text{Th}} + R_{\text{L}})^4}\left[-3(R_{\text{Th}} - R_{\text{L}}) - (R_{\text{Th}} + R_{\text{L}})\right] \\
&= \frac{2\,v^2(R_{\text{L}} - 2R_{\text{Th}})}{(R_{\text{Th}} + R_{\text{L}})^4}
\end{aligned}$$

Finally, the second derivative at the point of interest is

$$\left.\frac{d^2P}{dR_{\text{L}}^2}\right|_{R_{\text{L}}=R_{\text{Th}}} = v^2\left.\frac{2R_{\text{L}} - 4R_{\text{Th}}}{(R_{\text{Th}} + R_{\text{L}})^4}\right|_{R_{\text{L}}=R_{\text{Th}}} = v^2\frac{-2R_{\text{Th}}}{(2R_{\text{Th}})^4}$$

Since the second derivative is negative for all R_{Th}, it may be concluded that the maximum power transfer does occur at $R_{\text{L}} = R_{Th}$.

Power Series

A *power series* is an infinite series of the form

$$\sum_{m=0}^{\infty} a_m(x - x_0)^m = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \cdots \tag{33}$$

where x_0 is the *center*. Variables x, x_0, and $a_0, a_1, a_2, \ldots$ are real.

Table 3. Derivatives of a Few Commonly Used Functions

$\frac{d}{dx}u^n = nu^{n-1}\frac{du}{dx}$	$\frac{d}{dx}a^u = a^u \ln(a)\frac{du}{dx}, a > 0$
$\frac{d}{dx}\ln(u) = \frac{1}{u}\frac{du}{dx}$	$\frac{d}{dx}e^{au} = ae^{au}\frac{du}{dx}$
$\frac{d}{dx}\sin(au) = a\cos(au)\frac{du}{dx}$	$\frac{d}{dx}\cos(au) = -a\sin(au)\frac{du}{dx}$

Maclaurin Series. The Maclaurin series uses the origin, $x_0 = 0$, as its reference point to expand a function:

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \cdots + \frac{f^{(n)}(0)}{n!}x^n + \cdots \tag{34}$$

Use of the Maclaurin series leads quickly to series expansions for the exponential, (co)sine, and hyperbolic (co)sine functions as given below:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots + \frac{x^n}{n!} + \cdots \tag{35}$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^n\frac{x^{2n+1}}{(2n+1)!} + \cdots \tag{36}$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots + (-1)^n\frac{x^{2n}}{(2n)!} + \cdots \tag{37}$$

$$\sinh(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \cdots + \frac{x^{2n+1}}{(2n+1)!} + \cdots \tag{38}$$

$$\cosh(x) = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \cdots + \frac{x^{2n}}{(2n)!} + \cdots \tag{39}$$

Maclaurin Series Example. The Maclaurin series may be used to expand e^x to find Euler's identities. Begin with

$$\begin{aligned} e^{\pm j\theta} &= 1 + (\pm j\theta) + \frac{(\pm j\theta)^2}{2!} + \frac{(\pm j\theta)^3}{3!} + \frac{(\pm j\theta)^4}{4!} + \frac{(\pm j\theta)^5}{5!} + \cdots \\ &= 1 \pm j\theta - \frac{\theta^2}{2!} \mp \frac{j\theta^3}{3!} + \frac{\theta^4}{4!} \pm \frac{j\theta^5}{5!} \cdots \\ &= 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} \cdots \pm j\left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} \cdots\right) \\ &= \cos(\theta) \pm j\sin(\theta) \end{aligned}$$

Adding and subtracting these two sinusoidal expressions

$$e^{j\theta} = \cos(\theta) + j\sin(\theta)$$
$$e^{-j\theta} = \cos(\theta) - j\sin(\theta)$$

along with a division by 2 and $2j$, respectively, form *Euler's identities:*

$$\cos(\theta) = \frac{e^{j\theta} + e^{-j\theta}}{2} \quad \text{and} \tag{40}$$

$$\sin(\theta) = \frac{e^{j\theta} - e^{-j\theta}}{2j} \tag{41}$$

In the special case of $\theta = \pi$, the identity becomes Euler's formula of

$$e^{j\pi} + 1 = 0$$

This formula connects both the fundamental values (of 0, 1, j, e and π) and the basic mathematical operators (addition, multiplication, raised power and equals).

Taylor Series. The Taylor series is more general than the Maclaurin series because it uses an arbitrary reference point, x_0:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(x_0)}{3!}(x - x_0)^3 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + \cdots \tag{42}$$

Binomial Series. Related is the binomial series expansion, which converges for $x^2 < a^2$:

$$\begin{aligned} (a+x)^n &= a^n + na^{n-1}x + \frac{n(n-1)}{2!}a^{n-2}x^2 \\ &\quad + \frac{n(n-1)(n-2)}{3!}a^{n-3}x^3 + \cdots + x^n \\ &= a^n + \binom{n}{1}a^{n-1}x + \binom{n}{2}a^{n-2}x^2 \\ &\quad + \cdots + \binom{n}{k}a^{n-k}x^k + \cdots + x^n \end{aligned} \tag{43}$$

where the binomial coefficients are given by

$$\binom{n}{k} = \frac{n!}{k!(n-k!)} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!} \tag{44}$$

Binomial Series Example. The binomial series expansion may be used to derive the classic expression for kinetic energy from the relativistic expression below:

$$E_k = m_0c^2\left[\frac{1}{\sqrt{1-\beta^2}} - 1\right]$$

where $\beta = v/c$, the fraction of light speed an object is traveling. The reciprocated square root term is expanded using the binomial formula above where $n = -\frac{1}{2}$, $a = 1$, and $x = -\beta^2$, which meets the convergence restriction. The ex-

pansion then is

$$(1-\beta^2)^{-1/2} = (1)^{-1/2} + \frac{-1}{2}(1)^{-3/2}(-\beta^2) + \frac{\frac{-1}{2}\left(\frac{-1}{2}-1\right)}{2!}(1)^{-5/2}(-\beta^2)^2 + \cdots = 1 + \frac{1}{2}\beta^2 + \frac{3}{8}\beta^4 + \cdots$$

If $v < c$, the β^4 and higher terms become insignificant. Substituting the expansion into the relativistic expression for kinetic energy yields

$$E_k \approx m_0 c^2 \left(1 + \frac{1}{2}\frac{v^2}{c^2} - 1\right) = \frac{1}{2} m_0 v^2$$

Numerical Differentiation

Numerical differentiation, although perhaps less common than numerical integration (presented later), is, to a first order, a straightforward extension of Equation (14). For small values of Δx, the first derivative at x_x is

$$f\prime(x_i) = \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x} \tag{45}$$

If Δx is positive, the above expression is referred to as a forward-difference formula, whereas if Δx is negative, it is termed a backward-difference formula. Greater accuracy can be obtained using formulas that employ data points on both sides of x_i. For instance, although $f(x_i)$ does not explicitly appear in the following equations, they are known as three-point and five point formulas respectively

$$f\prime(x_i) = \frac{f(x_i + \Delta x) - f(x_i - \Delta x)}{2\,\Delta x} \tag{46}$$

$$f\prime(x_i) = \frac{f(x_i - 2\,\Delta x) - 8\,f(x_i - \Delta x) + 8\,f(x_i + \Delta x) - f(x_i + 2\,\Delta x)}{12\,\Delta x} \tag{47}$$

INTEGRAL CALCULUS

Indefinite Integrals

Differentiation and integration are inverse operations. There are two fundamental issues associated with integral calculus. The first is to find integrals or antiderivatives of a function, that is, given an expression, find another function that has the first function as its derivative. The second problem is to evaluate a definite integral as a limit of a sum. As an example, consider $y = x^2$, which is an integral of $2x$. It is important to point out that the integral is not unique and that x^2 represents a family of functions with the same derivative. Therefore, the solution should be augmented with an *integration constant, c*, added to each expression to represent the *indefinite integral*. This is so because the derivative of a constant is zero. If $F(x)$ is an integral of $f(x)$, then

$$F(x) = \int f(x)\,dx + c \tag{48}$$

The addition of the integration constant represents all integrals of a function. The symbol $\int$, a medieval S, stands for *summa* (sum).

The process of finding the integral of a function is called *integration*. While the determination of the derivative of a function is rather straightforward since definite rules exist, there is no general method for finding the integral of a mathematical expression. Calculus gives rules for integrating large classes of functions. When these rules fail, approximate or numerical methods permit the evaluation of the integral for a given value of x.

Selected indefinite integrals are given in Table 4. Although extensive integral tables exist, there are expressions whose integrals are not listed. Therefore, it is important to be cognizant of rules such as integration by parts or some form of transformation to arrive at the integral of the desired mathematical expression.

Integration Rules

Properties that hold for the definite integral include scaling and linearity:

$$\int_a^b [cf(x) \pm kg(x)]dx = c\int_a^b f(x)\,dx \pm k\int_a^b g(x)\,dx \tag{49}$$

They also include particular properties due to the limits of integration:

$$\int_a^b f(x)\,dx = -\int_b^a f(x)\,dx \tag{50}$$

$$\int_a^b f(x)\,dx = \int_a^c f(x)\,dx + \int_c^b f(x)\,dx \tag{51}$$

Transformations

Transformation is one method to facilitate evaluating integrals. Perhaps the simplest form of transformation is substitution. Other complex types of transformation are also possible, and some integral tables suggest appropriate substitutions for integrals, which are similar to the integrals in the table. Experience as well as intuition are the two most important factors in finding the right transformation. In performing the substitution with the definite integrals, it is important to change the limits. Particularly, the *change of limits rule* states that if the integral $\int f(g(x))g'(x)\,dx$ is subjected to the substitution $u = g(x)$, so that the integral becomes $\int f(u)\,du$, then

$$\int_a^b f(g(x))g'(x)\,dx = \int_{g(a)}^{g(b)} f(u)\,du \tag{52}$$

Substitution Example. To determine the area of the ellipse $x^2/a^2 + y^2/b^2 = 1$, as shown in Fig. 3, the function may be rearranged to

$$y = b\sqrt{1 - (x/a)^2}$$

Table 4. Definite Integrals Without Constant of Integration

General Forms

$$\int u\,dv = uv - \int v\,du$$

$$\int a\,dx = ax$$

$$\int x^n dx = \frac{x^{n+1}}{n+1}, \qquad n \neq -1$$

$$\int \frac{1}{x}\,dx = \ln(x)$$

$$\int b^{ax} dx = \frac{b^{ax}}{a\ln(b)}, \qquad b > 0$$

$$\int a^x \ln(a)\,dx = a^x, \qquad a > 0$$

Forms with $a + bx$

$$\int (a+bx)^n dx = \frac{(a+bx)^{n+1}}{(n+1)b}, \qquad n \neq -1$$

$$\int \frac{1}{a+bx}\,dx = \frac{1}{b}\ln(a+bx)$$

$$\int \frac{x}{a+bx}\,dx = \frac{x}{b} - \frac{a}{b^2}\ln(a+bx)$$

$$\int \frac{1}{x(a+bx)}\,dx = -\frac{1}{a}\ln\left(\frac{a+bx}{x}\right)$$

$$\int \frac{1}{a+bx+cx^2}\,dx = \begin{cases} \frac{2}{\sqrt{4ac-b^2}}\tan^{-1}\left(\frac{2cx+b}{\sqrt{4ac-b^2}}\right), & 4ac - b^2 > 0 \\ \frac{2}{\sqrt{b^2-4ac}}\tanh^{-1}\left(\frac{2cx+b}{\sqrt{b^2-4ac}}\right), & 4ac - b^2 < 0 \end{cases}$$

Forms with $\sqrt{a+bx}$

$$\int \sqrt{a+bx}\,dx = \frac{2}{3b}\sqrt{(a+bx)^3}$$

$$\int x\sqrt{a+bx}\,dx = \frac{-2(2a-3bx)}{15b^2}\sqrt{(a+bx)^3}$$

$$\int \frac{1}{\sqrt{a+bx}}\,dx = \frac{2}{b}\sqrt{a+bx}$$

$$\int \frac{x}{\sqrt{a+bx}}\,dx = \frac{-2(2a-bx)}{3b^2}\sqrt{a+bx}$$

$$\int \frac{1}{x\sqrt{a+bx}}\,dx = \begin{cases} \frac{1}{\sqrt{a}}\ln\left(\frac{\sqrt{a+bx}-\sqrt{a}}{\sqrt{a+bx}+\sqrt{a}}\right), & a > 0 \\ \frac{2}{\sqrt{-a}}\tan^{-1}\left(\sqrt{\frac{a+bx}{-a}}\right), & a < 0 \end{cases}$$

Forms with $a^2 \pm x^2$ or $x^2 \pm a^2$

$$\int \frac{1}{a^2+x^2}\,dx = \frac{1}{a}\tan^{-1}\left(\frac{x}{a}\right)$$

$$\int \frac{1}{a^2-x^2}\,dx = \frac{1}{a}\tanh^{-1}\left(\frac{x}{a}\right)$$

$$\int \frac{x}{a^2 \pm x^2}\,dx = \pm\frac{1}{2}\ln(a^2 \pm x^2)$$

$$\int \frac{1}{x^2-a^2}\,dx = \frac{1}{2a}\ln\left[\frac{x-a}{x+a}\right], \qquad x^2 > a^2$$

Forms with $\sqrt{a^2 \pm x^2}$ or $\sqrt{x^2 \pm a^2}$

$$\int \frac{1}{\sqrt{a^2-x^2}}\,dx = \sin^{-1}\left(\frac{x}{|a|}\right)$$

$$\int \sqrt{x^2 \pm a^2}\,dx = \frac{1}{2}\left[x\sqrt{x^2 \pm a^2} \pm a^2\ln\left(x + \sqrt{x^2 \pm a^2}\right)\right]$$

$$\int \frac{1}{\sqrt{x^2 \pm a^2}}\,dx = \ln\left(x + \sqrt{x^2 \pm a^2}\right)$$

$$\int \frac{x}{\sqrt{x^2 \pm a^2}}\,dx = \sqrt{x^2 \pm a^2}$$

$$\int \frac{1}{x\sqrt{x^2-a^2}}\,dx = \frac{1}{|a|}\sec^{-1}\left(\frac{x}{a}\right)$$

$$\int \frac{1}{x\sqrt{a^2+x^2}}\,dx = \frac{-1}{a}\ln\left(\frac{a+\sqrt{a^2+x^2}}{x}\right)$$

Forms with $a + bx^2$

$$\int \frac{1}{a+bx^2}\,dx = \begin{cases} \frac{1}{\sqrt{ab}}\tan^{-1}\left(\frac{x\sqrt{ab}}{a}\right), & ab > 0 \\ \frac{1}{\sqrt{-ab}}\tanh^{-1}\left(\frac{x\sqrt{-ab}}{a}\right), & ab < 0 \end{cases}$$

$$\int \frac{x}{a+bx^2}\,dx = \frac{1}{2b}\ln(a+bx^2)$$

$$\int \frac{1}{x(a+bx^2)}\,dx = \frac{1}{2a}\ln\left(\frac{x^2}{a+bx^2}\right)$$

Logarithmic Forms

$$\int \ln(x)\,dx = x\ln(x) - x$$

$$\int x\ln(x)\,dx = \frac{x^2}{2}\ln(x) - \frac{x^2}{4}$$

$$\int x^n \ln(ax)\,dx = \frac{x^{n+1}}{n+1}\ln(ax) - \frac{x^{n+1}}{(n+1)^2}, \qquad n \neq -1$$

$$\int \frac{\ln(x)}{x}\,dx = \frac{1}{2}[\ln(x)]^2$$

$$\int \frac{\ln(x)}{x^2}\,dx = \frac{-\ln(x)}{x} - \frac{1}{x}$$

$$\int \frac{1}{x\ln(x)}\,dx = \ln[\ln(x)]$$

$$\int [\ln(x)]^2\,dx = x[\ln(x)]^2 - 2x\ln(x) + 2x$$

Exponential Forms

$$\int e^{ax}\,dx = \frac{e^{ax}}{a}$$

$$\int xe^{ax}\,dx = \frac{e^{ax}}{a^2}(ax-1)$$

$$\int x^2 e^{ax}\,dx = \frac{e^{ax}}{a^3}(a^2x^2 - 2ax + 2)$$

$$\int xe^{-x^2}\,dx = \frac{-1}{2}e^{-x^2}$$

$$\int \frac{1}{1+e^x}\,dx = x - \ln(1+e^x) - \ln\left(\frac{e^x}{1+e^x}\right)$$

$$\int \frac{1}{a+be^{nx}}\,dx = \frac{x}{a} - \frac{1}{an}\ln(a+be^{nx})$$

$$\int e^{ax}\sin(bx)\,dx = \frac{e^{ax}}{a^2+b^2}[a\sin(bx) - b\cos(bx)]$$

$$\int e^{ax}\cos(bx)\,dx = \frac{e^{ax}}{a^2+b^2}[a\cos(bx) + b\sin(bx)]$$

$$\int e^{ax}\sin^2(bx)\,dx = \frac{e^{ax}}{a^2+4b^2}\left\{[a\sin(bx) - 2b\cos(bx)]\sin(bx) + \frac{2b^2}{a}\right\}$$

$$\int e^{ax}\cos^2(bx)\,dx = \frac{e^{ax}}{a^2+4b^2}\left\{[a\cos(bx) + 2b\sin(bx)]\cos(bx) + \frac{2b^2}{a}\right\}$$

$$\int xe^{ax}\sin(bx)\,dx = \frac{xe^{ax}}{a^2+b^2}[a\sin(bx) - b\cos(bx)] - \frac{e^{ax}}{(a^2+b^2)^2}[(a^2-b^2)\sin(bx) - 2ab\cos(bx)]$$

$$\int xe^{ax}\cos(bx)\,dx = \frac{xe^{ax}}{a^2+b^2}[a\cos(bx) + b\sin(bx)] - \frac{e^{ax}}{(a^2+b^2)^2}[(a^2-b^2)\cos(bx) + 2ab\sin(bx)]$$

Table 4. *(Continued)*

Trigonometric Forms with Sine

$$\int \sin(ax)\,dx = -\frac{1}{a}\cos(ax)$$

$$\int \sin(a+bx)\,dx = \frac{-1}{b}\cos(a+bx)$$

$$\int \sin^2(ax)\,dx = \frac{x}{2} - \frac{\sin(2ax)}{4a} = \frac{x}{2} - \frac{1}{2a}\sin(ax)\cos(ax)$$

$$\int \sin(ax)\sin(bx)\,dx = \frac{\sin[(a-b)x]}{2(a-b)} - \frac{\sin[(a+b)x]}{2(a+b)}, \qquad a^2 \neq b^2$$

$$\int x\sin(ax)\,dx = \frac{1}{a^2}\sin(ax) - \frac{x}{a}\cos(ax)$$

$$\int x^2\sin(ax)\,dx = \frac{2x}{a^2}\sin(ax) - \frac{a^2x^2-2}{a^3}\cos(ax)$$

Trigonometric Forms with Cosine

$$\int \cos(ax)\,dx = \frac{1}{a}\sin(ax)$$

$$\int \cos(a+bx)\,dx = \frac{1}{b}\sin(a+bx)$$

$$\int \frac{1}{1+\cos(ax)}\,dx = \frac{1}{a}\tan\left(\frac{ax}{2}\right)$$

$$\int \frac{\cos(ax)}{1+\cos(ax)}\,dx = x - \frac{1}{a}\tan\left(\frac{ax}{2}\right)$$

$$\int \cos^2(ax)\,dx = \frac{x}{2} + \frac{\sin(2ax)}{4a} = \frac{x}{2} + \frac{1}{2a}\sin(ax)\cos(ax)$$

$$\int \cos(ax)\cos(bx)\,dx = \frac{\sin[(a-b)x]}{2(a-b)} + \frac{\sin[(a+b)x]}{2(a+b)}, \qquad a^2 \neq b^2$$

$$\int x\cos(ax)\,dx = \frac{1}{a^2}\cos(ax) + \frac{x}{a}\sin(ax)$$

$$\int x^2\cos(ax)\,dx = \frac{2x}{a^2}\cos(ax) + \frac{a^2x^2-2}{a^3}\sin(ax)$$

Trigonometric Forms with Sine and Cosine

$$\int \sin(ax)\cos(bx)\,dx = -\frac{\cos[(a-b)x]}{2(a-b)} - \frac{\cos[(a+b)x]}{2(a+b)}, \qquad a^2 \neq b^2$$

$$\int \sin(ax)\cos(ax)\,dx = \frac{1}{2a}\sin^2(ax)$$

$$\int \sin^2(ax)\cos^2(ax)\,dx = \frac{x}{8} - \frac{1}{32a}\sin(4ax)$$

$$\int \sin(ax)\cos^n(ax)\,dx = \frac{-\cos^{n+1}(ax)}{(n+1)a}$$

$$\int \sin^n(ax)\cos(ax)\,dx = \frac{\sin^{n+1}(ax)}{(n+1)a}$$

$$\int \frac{\sin(ax)}{\cos^2(ax)}\,dx = \frac{1}{a\cos(ax)} = \frac{\sec(ax)}{a}$$

$$\int \frac{\cos(ax)}{\sin^2(ax)}\,dx = \frac{-1}{a\sin(ax)} = \frac{-\csc(ax)}{a}$$

$$\int \frac{x+\sin(x)}{1+\cos(x)}\,dx = x\tan\left(\frac{x}{2}\right)$$

Trigonometric Forms Without Sine and Cosine

$$\int \tan(ax)\,dx = -\frac{1}{a}\ln|\cos(ax)| = \frac{1}{a}\ln|\sec(ax)|$$

$$\int \cot(ax)\,dx = \frac{1}{a}\ln|\sin(ax)| = \frac{-1}{a}\ln|\csc(ax)|$$

$$\int \sec(ax)\,dx = \frac{1}{a}\ln|\sec(ax) + \tan(ax)| = \frac{1}{a}\ln\left[\tan\left(\frac{ax}{2} + \frac{\pi}{4}\right)\right]$$

$$\int \csc(ax)\,dx = \frac{1}{a}\ln|\csc(ax) - \cot(ax)| = \frac{1}{a}\ln\left[\tan\left(\frac{ax}{2}\right)\right]$$

$$\int \tan^2(ax)\,dx = \frac{1}{a}\tan(ax) - x$$

$$\int \sec^2(ax)\,dx = \int \frac{1}{\cos^2(ax)}\,dx = \frac{1}{a}\tan(ax)$$

$$\int \csc^2(ax)\,dx = \int \frac{1}{\sin^2(ax)}\,dx = \frac{-1}{a}\cot(ax)$$

Inverse Trigonometric Forms

$$\int \sin^{-1}(ax)\,dx = x\sin^{-1}(ax) + \frac{\sqrt{1-a^2x^2}}{a}$$

$$\int \cos^{-1}(ax)\,dx = x\cos^{-1}(ax) - \frac{\sqrt{1-a^2x^2}}{a}$$

$$\int \tan^{-1}(ax)\,dx = x\tan^{-1}(ax) - \frac{1}{2a}\ln(1+a^2x^2)$$

Hyperbolic Forms

$$\int \sinh(ax)\,dx = \frac{1}{a}\cosh(ax)$$

$$\int \cosh(ax)\,dx = \frac{1}{a}\sinh(ax)$$

$$\int \tanh(ax)\,dx = \frac{1}{a}\ln|\cosh(ax)|$$

$$\int \sinh^{-1}\left(\frac{x}{a}\right)dx = x\sinh^{-1}\left(\frac{x}{a}\right) - \sqrt{x^2+a^2}$$

$$\int x\sinh(ax)\,dx = \frac{x\cosh(ax)}{a} - \frac{\sinh(ax)}{a^2}$$

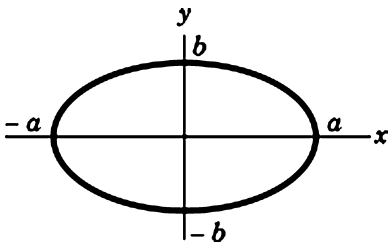


Figure 3. The area of the region encompassed by the ellipse $x^2/a^2 + y^2/b^2 = 1$ may be obtained by taking advantage of the symmetric structure of the function. To this end, the total area is twice the area of the region above the x-axis, which is equal to $\int_{-a}^{+a} b\sqrt{1-(x/a)^2}\,dx$.

Taking advantage of the symmetric nature of the function, the area of the ellipse is twice the area of its upper half:

$$A = 2\int_{-a}^{a} b\sqrt{1-(x/a)^2}\,dx$$

Let $u = x/a$, which results in $du = dx/a$. When $x = -a$ we obtain $u = -1$; similarly, $u = 1$ for $x = a$. Thus

$$\begin{aligned} A &= 2\int_{-a}^{a} b\sqrt{1-(x/a)^2}dx = 2ba\int_{-a}^{a}(1/a)\sqrt{1-(x/a)^2}dx \\ &= 2ba\int_{-1}^{1}\sqrt{1-(u)^2}du \end{aligned}$$

We know that in general

$$\int \sqrt{c^2-u^2}dx = \frac{u}{2}\sqrt{c^2-u^2} + \frac{c^2}{2}\sin^{-1}\left(\frac{u}{c}\right)$$

Since here $c = 1$, the area is

$$\begin{aligned} A &= 2ba\int_{-1}^{1}\sqrt{1-(u)^2}du \\ &= 2ba\left[\frac{u}{2}\sqrt{1-u^2} + \frac{1}{2}\sin^{-1}(u)\right]\Big|_{-1}^{1} \\ &= 2ba\left\{\left[\frac{1}{2}\sqrt{1-(1)^2} + \frac{1}{2}\sin^{-1}(1)\right]\right. \\ &\quad \left. -\left[\frac{-1}{2}\sqrt{1-(-1)^2} + \frac{1}{2}\sin^{-1}(-1)\right]\right\} \\ &= 2ba\left\{\left[0+\left(\frac{1}{2}\right)\left(\frac{\pi}{2}\right)\right] - \left[0+\left(\frac{1}{2}\right)\left(\frac{-\pi}{2}\right)\right]\right\} \\ &= \pi ab. \end{aligned}$$

Integration Example. Calculation of the *root-mean-square* (*rms*) value of a function is a classic use of the integral. The rms value is found by first squaring the waveform, followed by computing its average, and finally by taking its square root. Consider the determination of the rms value of a sinusoidal current, $i(t) = I_M \cos(\omega t + \theta)$, of constant frequency, ω, and constant phase shift, θ. The rms current is found over a representative period, $T = 2\pi/\omega$:

$$I_{rms} = \sqrt{\frac{1}{T}\int_0^T i^2(t)\,dt} = \sqrt{\frac{1}{T}\int_0^T I_M^2\cos^2(\omega t+\theta)\,dt}$$

The solution to the integral may be found using a change of variables and the table of integrals. First, let $u = \omega t + \theta$, such that $du = \omega\,dt$. The variable change modifies the upper and lower limits of integration to $\omega T + \theta$ and θ, respectively. The expression for integral now appears as

$$\begin{aligned} \int_0^T I_M^2\cos^2(\omega t+\theta)\,dt &= \int_\theta^{\omega T+\theta} I_M^2\cos^2(u)\frac{du}{\omega} \\ &= \frac{I_M^2}{\omega}\int_\theta^{\omega T+\theta}\cos^2(u)\,du \end{aligned}$$

Using the table of integrals (Table 4), we obtain

$$\begin{aligned} \int_\theta^{\omega T+\theta}\cos^2(u)\,du &= \frac{u}{2} + \frac{\sin(2u)}{4}\Big|_\theta^{\omega T+\theta} \\ &= \left\{\frac{\omega T+\theta}{2} + \frac{\sin[2(\omega T+\theta)]}{4}\right\} \\ &\quad - \left\{\frac{\theta}{2} + \frac{\sin(2\theta)}{4}\right\} \\ &= \frac{\omega T}{2} + \frac{\sin[2(\omega T+\theta)] - \sin(2\theta)}{4} \\ &= \frac{2\pi}{2} + \frac{\sin[2(2\pi+\theta)] - \sin(2\theta)}{4} = \pi \end{aligned}$$

Thus, the rms value is

$$I_{rms} = \sqrt{\frac{1}{T}\frac{I_M^2}{\omega}\pi} = \sqrt{\frac{I_M^2\pi}{2\pi}} = \frac{I_M}{\sqrt{2}}$$

Integration by Parts

One of the most important techniques of integration is the principle of integration by parts. Let $f(x)$ and $g(x)$ be any two functions and let $G(x)$ be an antiderivative of $g(x)$. Using the product rule for derivatives, the integral of the product of the two functions can be derived as

$$\int f(x)g(x)\,dx = f(x)G(x) - \int f'(x)G(x)\,dx \tag{53}$$

For definite integrals we obtain

$$\int_a^b f(x)g(x)\,dx = f(x)G(x)\Big|_a^b - \int_a^b f'(x)G(x)\,dx \tag{54}$$

Integration by Parts Example. This example illustrates integration by parts in evaluating the *Laplace transform,* which is defined by

$$v = \int e^{-st}\,dt = -e^{-st}/s \tag{55}$$

Here we transform a ramp function, $f(t) = at$. Let $u = at$ and $dv = e^{-st}\,dt$. Hence, $du = a\,dt$ and $v = \int e^{-st}\,dt = -e^{-st}/s$.

The Laplace transform of a ramp is

$$\begin{aligned} F(s) &= \int_0^\infty at\,e^{-st}\,dt = at\,\frac{-e^{-st}}{s}\,|_0^\infty - \int_0^\infty \frac{-e^{-st}}{s}\,a\,dt \\ &= \left[a\cdot\infty\,\frac{-e^{-s\infty}}{s} - a\cdot 0\,\frac{-e^{-s0}}{s}\right] + \frac{a}{s}\left(\frac{e^{-st}}{-s}\right)|_0^\infty \\ &= 0 + \frac{-a}{s^2}(e^{-s\infty} - e^{-s0}) = \frac{a}{s^2}\,. \end{aligned}$$

Definite Integrals

A definite integral is the limit of a sum. Common applications of the definite integral include determination of area, arc length, volume, and function average. These quantities can be approximated by sums obtained by dividing the given quantity into small parts and approximating each part. The definite integral allows one to arrive at the exact values of these quantities instead of their approximate values.

The symbol $\int_a^b f(x)\,dx$ is the *definite integral* of $f(x)\,dx$ on interval $[a, b]$. Let $f(x)$ be a single-valued function of x, defined at each point on $[a, b]$. Choose points x_i on the interval such that

$$a = x_0 < x_1 < x_2 < \cdots < x_n = b$$

Let $\Delta x_i = x_i - x_{i-1}$. Choose in each interval Δx_i a point t_i. Form the sum

$$S_n = \sum_{i=1}^{n} f(t_i)\Delta x_i = f(t_1)\Delta x_1 + f(t_2)\Delta x_2 + \cdots + f(t_n)\,\Delta x_n$$

The limit of this sum, as the largest interval approaches zero, is defined as the *definite integral* $\int_a^b f(x)\,dx$, if it can exist. The existence of f is guaranteed if it is a continuous function on $[a, b]$. If $F(x)$ is a function whose derivative is $f(x)$, then it can be shown that

$$\int_a^b f(x)\,dx = F(b) - F(a) \tag{56}$$

This is essentially the fundamental theorem of calculus. If F does not exist, numerical methods may be used to obtain the value of the integral. Several definite integrals important in engineering are listed in Table 5. For a more comprehensive list of integrals, the reader is referred to a number of calculus texts.

Applications. One use of definite integrals is to find the areas bounded by certain curves. For example, the area bounded by the polar function $f(\theta)$ and the lines θ_a and θ_b is

$$A = \frac{1}{2}\int_{\theta_a}^{\theta_b} [f(\theta)]^2\,d\theta \tag{57}$$

Another application of integration is to find an average (mean) value:

$$\bar{x} = \frac{1}{b-a}\int_a^b x(t)\,dt \tag{58}$$

Integration is used to find arc length from point a to point b:

$$L = \int_a^b \sqrt{1 + [f'(x)]^2}\,dx \tag{59}$$

Or it is used to find arc length in polar coordinates:

$$L = \int_{\theta_a}^{\theta_b} \sqrt{[f(\theta)]^2 + [f'(\theta)]^2}\,d\theta \tag{60}$$

Multiple Integration

The double integral of $f(x, y)$ over some region R is the generalization of the definite integral and is denoted as

$$\iint_R f(x, y)\,dx\,dy \tag{61}$$

It is typically applied to find the volume encompassed by a surface, the center of gravity of a given structure, and moments of inertia.

Let $f(x, y)$ be a function of two variables, and let $g(x)$ and $h(x)$ be two functions of x alone. Furthermore, let a and b be real numbers. Then, an *iterated integral* is an expression of the form

$$\int_a^b \left(\int_{g(x)}^{h(x)} f(x, y)\,dy\right) dx \tag{62}$$

where $f(x, y)$ is first treated as a function of y alone. The inner integral is evaluated between the limits $y = g(x)$ and $y = h(x)$, which results in an expression that is a function of x alone. The resultant integrand is then evaluated between the limits of $x = a$ and $x = b$.

A similar principle applies to function of three or more independent variables. A change of variables in multiple integrals is generally accomplished with the aid of the Jacobian.

For a transformation of the form

$$x = f(u, v, w) \quad y = g(u, v, w) \quad z = h(u, v, w)$$

the Jacobian of the transformation is defined as

$$\frac{\partial(x, y, z)}{\partial(u, v, w)} = \begin{vmatrix} \dfrac{\partial x}{\partial u} & \dfrac{\partial x}{\partial v} & \dfrac{\partial x}{\partial w} \\ \dfrac{\partial y}{\partial u} & \dfrac{\partial y}{\partial v} & \dfrac{\partial y}{\partial w} \\ \dfrac{\partial z}{\partial u} & \dfrac{\partial z}{\partial v} & \dfrac{\partial z}{\partial w} \end{vmatrix} \tag{63}$$

Special Functions

Various other special functions exist. The gamma function is defined by the integral

$$\Gamma(n) = \int_0^\infty t^{n-1}e^{-t}\,dt \quad , n > 0 \tag{64}$$

The error function is given by

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt \tag{65}$$

Table 5. Definite Integrals

$$\int_0^\infty \frac{a}{a^2+x^2}dx = \begin{cases} \pi/2, & a>0 \\ 0, & a=0 \\ -\pi/2, & a<0 \end{cases}$$

$$\int_1^\infty \frac{1}{x^m}dx = \frac{1}{m-1}, \quad m>1$$

$$\int_0^\infty \frac{1}{(1+x)\sqrt{x}}dx = \pi$$

$$\int_0^\infty x^n e^{-ax}dx = \frac{n!}{a^{n+1}}, \quad a>0, n \text{ a positive integer}$$

$$\int_0^\infty e^{-ax^2}dx = \frac{\sqrt{\pi}}{2a}, \quad a>0$$

$$\int_0^\infty e^{-ax}\cos(bx)\,dx = \frac{a}{a^2+b^2}, \quad a>0$$

$$\int_0^\infty e^{-ax}\sin(bx)\,dx = \frac{b}{a^2+b^2}, \quad a>0$$

$$\int_0^\infty \frac{\sin(mx)}{x}dx = \begin{cases} \pi/2, & m>0 \\ 0, & m=0 \\ -\pi/2, & m<0 \end{cases}$$

$$\int_0^\infty \frac{\cos(x)}{x}dx = \infty$$

$$\int_0^\infty \frac{\tan(x)}{x}dx = \frac{\pi}{2}$$

$$\int_0^\infty \cos(ax^2)dx = \int_0^\infty \sin(ax^2)dx = \frac{1}{2}\sqrt{\frac{\pi}{2a}}$$

$$\int_0^\pi \sin(mx)\sin(nx)\,dx = \begin{cases} 0, & m,n \text{ integers and } m \neq n \\ \pi/2, & m,n \text{ integers and } m=n \end{cases}$$

$$\int_0^\pi \cos(mx)\cos(nx)\,dx = \begin{cases} 0, & m,n \text{ integers and } m \neq n \\ \pi/2, & m,n \text{ integers and } m=n \end{cases}$$

$$\int_0^\pi \sin(mx)\cos(nx)\,dx = \begin{cases} 0, & m+n \text{ even} \\ \dfrac{2m}{m^2-n^2}, & m+n \text{ odd} \end{cases}$$

$$\int_0^\infty \frac{\sin^2(px)}{x^2}dx = \frac{\pi p}{2}$$

$$\int_0^\infty \frac{1-\cos(px)}{x^2}dx = \frac{\pi p}{2}$$

$$\int_0^\infty \operatorname{sinc}(ax)\,dx = \int_0^\infty \frac{\sin(\pi ax)}{\pi ax} = \frac{1}{2a}, \quad a>0$$

$$\int_0^\infty \operatorname{sinc}^2(ax)\,dx = \int_0^\infty \frac{\sin^2(\pi ax)}{(\pi ax)^2} = \frac{1}{2a}, \quad a>0$$

$$\int_0^\infty \frac{\sin(x)}{\sqrt{x}}dx = \int_0^\infty \frac{\cos(x)}{\sqrt{x}}dx = \sqrt{\frac{\pi}{2}}$$

The complementary error function is simply: $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$.

Numerical Integration

Numerical methods may be used to approximate the definite integral in cases where either an analytical solution is unavailable or the function is unknown (as in the case of sampled data). The simplest numerical integration uses the Riemann sum in which the integral symbol becomes a summation, and the dx term becomes a partition, $\Delta x_i = [x_i - x_{i-1}]$, in $[a, b]$:

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} f(w_i)\,\Delta x_i \tag{66}$$

where w_i is any number, usually the midpoint, in partition Δx_i. The partition is typically a constant proportional to the number of partitions, $\Delta x = (b-a)/n$ (rectangle rule). As the magnitude of Δx decreases, the accuracy of the numerical estimates increases. A traditional approach of testing the solution convergence is to repeatedly halve the partition width until an acceptable error is reached.

Trapezoidal Rule. The trapezoidal rule improves the numerical estimate of the integral (as compared with the rectangle rule above) by fitting a piecewise linear approximation to each subinterval using its endpoints (see Fig. 4):

$$\int_a^b f(x)\,dx \approx \Delta x\left[\frac{f(a)+f(b)}{2} + \sum_{i=1}^{n-1} f(a+i\Delta x)\right] \tag{67}$$

Simpson's Rule. Simpson's rule is a further improvement employing a piecewise quadratic approximation. In this method, the number of subintervals must be even (i.e.,

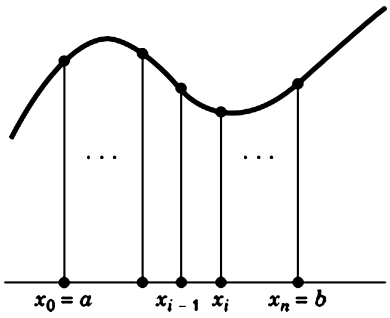


Figure 4. For trapezoidal numerical integration the curve is subdivided into equal increments between the left-hand limit at $x_0 = a$ and the right-hand limit at $x_n = b$. The area within each subinterval is approximated as $(x_i - x_{i-1})f(x_i) + f(x_{i-1})/2$. The integral is then numerically approximated by the summation of the subinterval areas.

$m = 2n$) and $\Delta x = (b - a)/m$. The numerical area is

$$\int_a^b f(x)\,dx \approx \frac{\Delta x}{3}\left\{f(a) - f(b) + \sum_{i=1}^{n}[4f(a + (2i-1)\Delta x) + 2f(a + 2i\Delta x)]\right\} \tag{68}$$

Calculus Software

With the advent of powerful personal computers, software has been developed for solving calculus problems and providing graphical visualization of their solutions. Many of these programs rely on symbolic processing that was pioneered in artificial intelligence. Caution should, however, be heeded in the use of these programs as they can result in nonsensical solutions. Some of the more advanced and commercial programs are Maple®, Mathematica®, Matlab®, and MathCad®. A discussion on these programs and their use in solving calculus problems is omitted here due to the evolving nature of such software, but the reader is referred to the Internet for Web-based calculus software resources.

ADDITIONAL TOPICS IN CALCULUS

Although differentiation and integration form the pillars of the use of calculus in engineering there are other mathematical tools, such as vectors and the convergence theorem, which transcend the boundaries of calculus. These topics are presented here.

Transformation of Coordinates

In some engineering applications, it is necessary to transform a given mathematical expression from one coordinate system to another. Examples of this transformation are those for the Laplacian operator, which appear later in this section. For the coordinate systems that appear in Fig. 5, the transformations appear in Table 6.

Vector Calculus

Consider a vector function

$$\boldsymbol{v}(x, y, z) = v_x(x, y, z)\boldsymbol{i} + v_y(x, y, z)\boldsymbol{j} + v_z(x, y, z)\boldsymbol{k} = \langle v_x, v_y, v_z\rangle \tag{69}$$

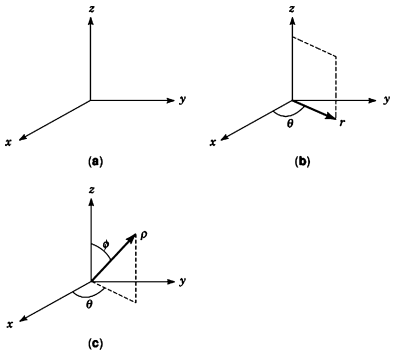


Figure 5. Cartesian (a), cylindrical (b), and spherical (c) coordinate systems are used in many engineering analyses. To facilitate an analysis, the coordinates of a given point may be transformed from one coordinate system to another. The transformation rules appear in Table 6.

where $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ are unit vectors in the positive x, y, and z directions, respectively. The magnitude of the vector is $\sqrt{v_x^2 + v_y^2 + v_z^2}$. The *dot product* (also referred to as the scalar or inner product) of $\boldsymbol{v}$ and $\boldsymbol{w}$ is defined as

$$\boldsymbol{v} \cdot \boldsymbol{w} \equiv v_x w_x + v_y w_y + v_z w_z = |\boldsymbol{v}||\boldsymbol{w}|\cos(\theta) \tag{70}$$

where θ is the angle between $\boldsymbol{v}$ and $\boldsymbol{w}$. Two vectors are orthogonal if and only if $\boldsymbol{v} \cdot \boldsymbol{w} = \boldsymbol{0}$.

The *cross product* or vector product of $\boldsymbol{v}$ and $\boldsymbol{w}$ is defined as

$$\boldsymbol{v} \times \boldsymbol{w} \equiv \begin{vmatrix} \boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k} \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{vmatrix} \quad \text{with} \quad |\boldsymbol{v} \times \boldsymbol{w}| = |\boldsymbol{v}||\boldsymbol{w}|\sin(\theta) \tag{71}$$

Two vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ are parallel if and only if $\boldsymbol{v} \times \boldsymbol{w} = \boldsymbol{0}$.

The *vector differential operator* ∇ ("del") is defined in three dimensions as

$$\nabla \equiv \frac{\partial}{\partial x}\boldsymbol{i} + \frac{\partial}{\partial y}\boldsymbol{j} + \frac{\partial}{\partial z}\boldsymbol{k} \tag{72}$$

The *gradient* of a scalar field, $f(x, y, z)$, is defined as

$$\text{grad}\, f = \nabla f = \frac{\partial f}{\partial x}\boldsymbol{i} + \frac{\partial f}{\partial y}\boldsymbol{j} + \frac{\partial f}{\partial z}\boldsymbol{k} \tag{73}$$

The *divergence* of a vector field is the dot product of the gradient operator and the vector field:

$$\text{div}\,\boldsymbol{v} = \nabla \cdot \boldsymbol{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \tag{74}$$

The *curl* of a vector field is the cross product of the gradient and the vector function:

$$\text{curl}\,\boldsymbol{v} = \nabla \times \boldsymbol{v} = \left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z}\right)\boldsymbol{i} + \left(\frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}\right)\boldsymbol{j} + \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}\right)\boldsymbol{k} \tag{75}$$

The curl of any gradient is the zero vector, $\nabla \times (\nabla f) = \boldsymbol{0}$. The divergence of any curl is zero, $\nabla \cdot (\nabla \times \boldsymbol{F}) = 0$. The *divergence* of a gradient of f is its Laplacian, denoted as $\nabla^2 f$ or Δf. For the Cartesian coordinate system the Laplacian is repre-

Table 6. Coordinate System Transformations

Cartesian	Cylindrical	Spherical
x	$r\cos\theta$	$\rho\sin\phi\cos\theta$
y	$r\sin\theta$	$\rho\sin\phi\sin\theta$
z	z	$\rho\cos\phi$
–	$r^2 = x^2 + y^2$	$\rho^2 = x^2 + y^2 + z^2$
$dV = dx\,dy\,dz$	$r\,dr\,d\theta\,dz$	$\rho^2\sin\phi\,d\rho\,d\phi\,d\theta$

sented as

$$\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \tag{76}$$

for the cylindrical coordinate system it is represented as

$$\nabla^2 = \frac{1}{r}\frac{\partial}{\partial r}r\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial\theta^2} + \frac{\partial^2}{\partial z^2} \tag{77}$$

and for the spherical coordinate system it is represented as

$$\nabla^2 = \frac{1}{\rho^2}\frac{\partial}{\partial\rho}\left(\rho^2\frac{\partial}{\partial\rho}\right) + \frac{1}{\rho^2\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial}{\partial\phi}\right) + \frac{1}{\rho^2\sin^2\phi}\frac{\partial^2}{\partial\theta^2} \tag{78}$$

Functions that satisfy Laplace's equation, $\nabla^2 f = 0$, are said to be *harmonic*.

Vector Calculus Example. Let $\boldsymbol{v} = x^2y\boldsymbol{i} + z\boldsymbol{j} + xyz\boldsymbol{i}$. The divergence of the vector is

$$\begin{aligned}\operatorname{div}\boldsymbol{v} &= \frac{\partial}{\partial x}(x^2y) + \frac{\partial}{\partial y}(z) + \frac{\partial}{\partial z}(xyz)\\ &= 2xy + 0 + xy = 3xy\end{aligned}$$

The curl of $\boldsymbol{v}$ is

$$\nabla\times\boldsymbol{v} = \begin{vmatrix}\boldsymbol{i} & \boldsymbol{j} & \boldsymbol{k}\\ \partial/\partial x & \partial/\partial y & \partial/\partial z\\ x^2y & z & xyz\end{vmatrix} = (xz-1)\boldsymbol{i} - yz\boldsymbol{j} - x^2\boldsymbol{k}$$

Gauss's and Stokes's Theorems. Maxwell's equations for electromagnetic fields are derived using the concepts of vector calculus applied to Faraday's law, Ampere's law, and Gauss's laws for electric and magnetic fields. The derivation is accomplished using Stokes's theorem and Gauss's divergence theorem.

The divergence theorem of Gauss provides a transformation of volume integrals into surface intervals, and conversely. Given a vector function $\boldsymbol{F}$ with continuous first partial derivatives in a region R bounded by a closed surface S

$$\iiint_R \nabla\cdot\boldsymbol{F}\,dV = \iint_S \boldsymbol{F}\cdot\boldsymbol{n}\,dS \tag{79}$$

where $\boldsymbol{n}$ is the outer unit normal to S. Physically, the flux of $\boldsymbol{F}$ across a closed surface is the integral of the divergence of $\boldsymbol{F}$ over the region.

Stokes's theorem provides a transformation of surface integrals into line integrals, and vice versa. The surface integral of the normal component of curl $\boldsymbol{F}$ over S equals the line integral of the tangential component of $\boldsymbol{F}$ taken along the simple (nonintersecting) closed curve C, which forms the boundary of the open surface S

$$\iint_S (\nabla\times\boldsymbol{F})\cdot\boldsymbol{n}\,dS = \oint_C \boldsymbol{F}\cdot d\boldsymbol{r} \tag{80}$$

where $\boldsymbol{r}$ is the position vector of the point on C. Stokes's theorem is a generalization of Green's theorem to three dimensions.

Singularity Functions in Engineering

Although strictly speaking they are not part of calculus, there are several singularity functions used in engineering problems worth examining while the subjects of differentiation and integration are explored. Two of the most common singularity functions are the unit step, $u(t)$, and the unit impulse or delta function, $\delta(t)$. The unit step is defined as

$$u(t-\tau) = \begin{cases}0, & t < \tau\\ 1, & t > \tau\end{cases} \tag{81}$$

The unit step function is discontinuous at $t = \tau$, where it abruptly jumps from zero to unity. Two unit step functions are oftentimes combined into a gate function as $u(t-\tau) - u[t-(\tau+T)]$, which is a pulse of period T. The delta function is a pulse of infinitesimal width and area (strength) of one, and it is defined as

$$\int_{-\infty}^{\infty} f(t)\delta(t-\tau)\,dt = f(\tau) \tag{82}$$

Hence, the unit step function is the integral of the unit impulse:

$$u(t) = \int_{-\infty}^{t}\delta(x)\,dx \tag{83}$$

The integration of the step function results in a ramp function.

BIBLIOGRAPHY

Numerous standard college texts on calculus exist. Some of these books and those that are more advanced are listed below.

E. Kreyszig, *Advanced Engineering Mathematics*, 7th ed., New York: Wiley, 1988.

E. W. Swokowski, *Calculus with Analytic Geometry*, 2nd ed., Boston: Prindle, Weber & Schmidt, 1979.

W. H. Beyer, *CRC Standard Mathematical Tables and Formulae*, 29th ed., Boca Raton, FL: CRC Press, 1991.

L. J. Goldstein, D. C. Lay, D .I. Schneider, *Calculus and its Applications*, 4th ed., Englewood Cliffs, NJ: Prentice-Hall, 1987.

M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*, New York: McGraw-Hill, 1968.

J. E. Marsden, A. J. Tromba, A. Weinstein, *Basic Multivariable Calculus*, New York: Springer-Verlag, 1993.

J. E. Marsden, A. J. Tromba, *Vector Calculus*, San Francisco: Freeman Co., 1988.

W. Kaplan, *Advanced Calculus*, 4th ed., Reading, MA: Addison-Wesley, 1991.

KEITH E. HOLBERT
A. SHARIF HEGER
Arizona State University,
Tempe, AZ
Los Alamos National
Laboratory, Los Alamos, NM

Chaos Time Series Analysis
Standard Article

Maurice E. Cohen[1] and Donna L. Hudson[2]
[1]University of California, San Francisco, Fresno, CA
[2]University of California, San Francisco, Fresno, CA

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2469
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (169K)

Abstract

The sections in this article are

Methods for Evaluation of Time Series Data

Evaluation of Experimental Data

Continuous Chaotic Modeling Versus Discrete Chaotic Modeling

| | |

Copyright © 1999-2008 All Rights Reserved.

CHAOTIC SYSTEMS CONTROL

Almost all real physical, biological, and chemical as well as many other systems are inherently nonlinear. This is also the case with electrical and electronic circuits. Apart from systems designed to perform linear operations (usually in such cases they just operate in a small region in which they behave linearly) there exists an abundance of systems that are nonlinear by their principle of operation. Rectifiers, flip-flops, modulators and demodulators, memory cells, analog to digital (A/D) converters, and different types of sensors are good examples of such systems. In many cases the designed circuit, when implemented, performs in a very unexpected way, totally different from that for which it was designed. In most cases, engineers do not care about the origins and mecha-

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

nisms of the malfunction; for them a circuit that does not perform as desired is of no use and has to be rejected or redesigned. Many of these unwanted phenomena, such as excess noise, false frequency lockings, squegging, and phase slipping have been found to be associated with bifurcations and chaotic behavior. Also many nonlinear phenomena in other science and engineering disciplines have a strong link with "electronic chaos." Examples are aperiodic electrocardiogram waveforms (reflecting fibrillations, arrythmias, or other types of heart malfunction), epileptic foci in electroencephalographic patterns, or other measurements taken by electronic means in plasma physics, lasers, fluid dynamics, nonlinear optics, semiconductors, and chemical or biological systems.

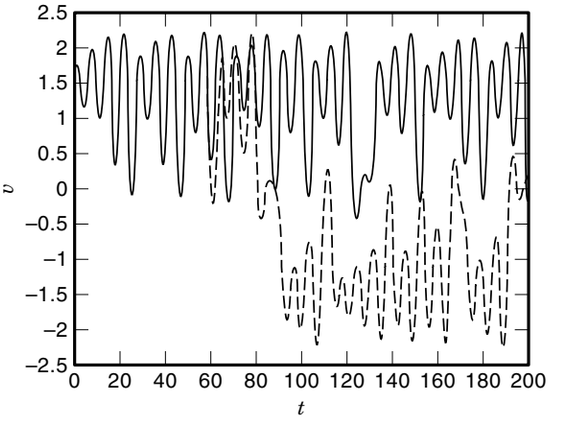


Figure 1. Illustration of the sensitive dependence on initial conditions—first fundamental property of chaotic systems. Two trajectories of Chua's oscillator starting from initial conditions with the difference of 0.001 in the first component for a short time stay close to each other but eventually separate resulting in waveforms of different shape.

DEFINITION OF IS CHAOTIC BEHAVIOR

In this section we consider only deterministic systems (i.e., systems for which knowledge of the initial state at some initial time t_0, equations of evolution and input signals fully determine the state and outputs for any $t \geq t_0$).

Typically deterministic systems display three types of behavior of their solutions: they approach constant solutions, they converge toward periodic solutions, or they converge toward quasi-periodic solutions. These are the situations known to every practicing engineer.

Now it has been confirmed that almost every physical system can also display behaviors that cannot be classified in any of the above-mentioned three categories; the systems become aperiodic (chaotic) if their parameters, internal variables, or external stimulations are chosen in a specific way. How can we describe chaos except saying that it is the kind of behavior that is not constant, periodic, or quasi-periodic or convergent to any of the above? For the purpose of this article we consider some specific properties to qualify behavior as chaotic:

1. The solutions show sensitive dependence on initial conditions (trajectories are unstable in the Lyapunov sense) but remain bounded in space as time elapses (are stable in the Lagrange sense).
2. Trajectory moves over a strange attractor, a geometric invariant object that can possess fractal dimension. The trajectory passes arbitrarily close to any point of the attractor set—that is, there is a dense trajectory.
3. Chaotic behavior appears in the system as via a "route" to chaos that typically is associated with a sequence of bifurcations, qualitative changes of observed behavior when varying one or more of the parameters.

Sensitive dependence on initial conditions means that trajectories of a chaotic system starting from nearly identical initial conditions will eventually separate and become uncorrelated (but they will always remain bounded in space). Large variations in the observed long-term behavior due to very small changes of initial state are often referred to as "the butterfly effect" (increment of butterfly wings can change weather in a far away region of earth). Figure 1 gives an example of two trajectories starting from initial conditions differing by 0.001; after remaining close to each other for some period, they eventually separate. Sensitive dependence on initial conditions for a system is realized only with some finite accuracy ϵ. If two initial conditions are closer to each other than ϵ, then they are not distinguishable in measurements. The trajectories of a chaotic system starting from such initial conditions will, after a finite time, diverge and become uncorrelated. For any precision we use in measurements (experiments) the behavior of trajectories is not predictable—the solutions look virtually random despite being produced by a deterministic system. There is also another consequence of this property that may be appealing for control purposes: a very small stimulus in the form of tiny change of parameters can have a very large effect on the system's behavior.

The second property can be explained easily by Fig. 2. It is clear that the trajectory shown in this figure "fills" out some

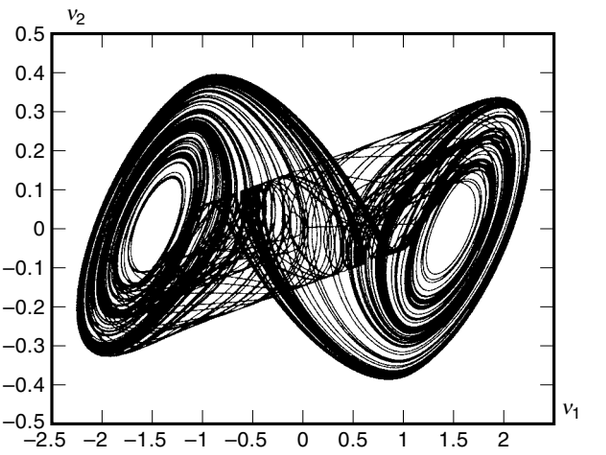


Figure 2. An example of a chaotic trajectory. Two-dimensional projection of the double scroll attractor observed in Chua's circuit is shown. The curve never closes itself, moves around in an unpredictable way, and densely fills some part of the space (here, the plane).

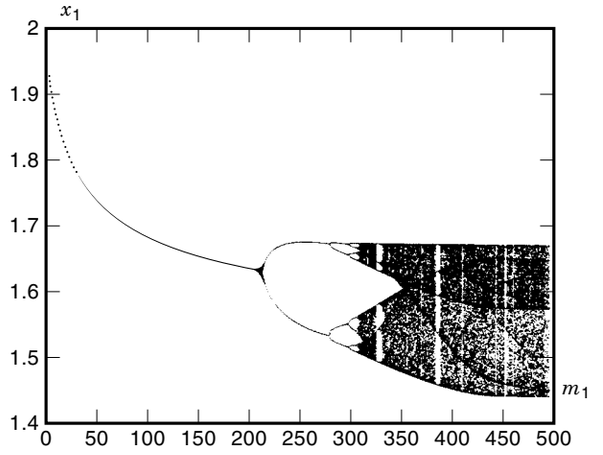


Figure 3. Bifurcation diagram for the *RC*-ladder chaos generator with slope m_1 chosen as bifurcation parameter. The diagram is obtained in such a way as for every chosen parameter value (abcissa) the long-term behavior of the chosen system variable is observed and coordinates of intersections of the orbit with a chosen plane are recorded and plotted. Thus for a chosen parameter value, the number of points plotted tells exactly what kind of behavior is observed. One point corresponds to a period-one orbit, two points to a period-two orbit, and a large number of points spread in an interval can be interpreted as chaotic behavior. Visible chaos appears via a "route" when the parameter is changed continuously—here, branching of the bifurcation tree can be interpreted as period doubling route to chaos. The diagram also confirms existence of a large variety of qualitatively different behaviors existing for suitably chosen values of parameter.

part of the space. If we arbitrarily choose a point within this region of space and a small ball of radius ϵ around it, the trajector will eventually pass through this ball after a finite time (which might be very long). As an example of the third property we give a typical bifurcation diagram obtained in numerical experiments (Fig. 3). By a suitable choice of parameter m_1 one can choose almost every type of periodic behavior apart from many chaotic states. There is an important fact often associated with bifurications: in many cases creation of new types of new trajectories that are observable in experiments (stable) via bifurcation is accompanied by creation of unstable orbits—invisible in experiments. Many of these unstable orbits persist also within the chaotic attractor. Many authors consider as fundamental the property of existence of a countable (infinite) number of unstable periodic orbits within an attractor.

Using proprietary numerical procedures it is possible to detect some of such orbits in numerical experiments (1). Figure 4 shows some of the periodic orbits uncovered from the double scroll attractor shown in Fig. 2.

The above-described fundamental properties of chaotic systems (their solutions) is the basis of the chaos control approaches described below.

WHAT CHAOS CONTROL MEANS

Chaos, so commonly encountered in physical systems, represents a rather peculiar type of behavior commonly considered as causing malfunctions, disastrous in most applications. It is obvious that an amplifier, a filter, an A/D converter, a phase-locked loop, or a digital filter generating chaotic responses is of no use—at least for its original purpose. Similarly, we would like to avoid situations where the heart does not pump blood properly (fibrillation or arrythmias) or epileptic attacks. Even more spectacular potential applications might be influencing rainfall and avoiding hurricanes and other atmospheric disasters believed to be associated with large-scale chaotic behavior.

The most common goal of control for a chaotic system is suppression of oscillations of the "bad" kind and influencing the system in such a way that it will produce a prescribed, desired motion. The goals vary depending on a particular application. The most common goal is to convert chaotic motion into a stable periodic or constant one. It is not at all obvious how such a goal could be achieved, because one of the fundamental features of chaotic systems, the sensitive dependence on initial conditions, seems to contradict any stable system operation. Recently, several applications have been mentioned in the literature in which the desired state of system operation is chaotic. The control problems in such cases are defined as: converting unwanted chaotic behavior into another kind of chaotic motion with prescribed properties (this is the goal of chaos synchronization) or changing periodic behavior into chaotic motion (which might be the goal in the case of epileptic seizures). The last-mentioned type of control is often referred to as *anticontrol* of chaos.

Many chaotic systems display what is called multiple basins of attracton and fractal basin boundaries. This means that, depending on the initial conditions, trajectories can converge to different steady states. Trajectories in nonlinear systems may possess several different limit sets and thus exhibit a variety of steady-state behaviors depending on the initial condition, chaotic or otherwise. In many cases, the sets of initial states leading to a particular type of behavior are intertwined in a complicated way forming fractal structures. Thus we could consider elimination of multiple basins of attraction as another kind of control goal.

In some cases, chaos is the dynamic state in which we would like the system to operate. We can imagine that mixing of components in a chemical reactor would be much quicker in a chaotic state than in any other one, or that chaotic signals could be useful for hiding information. In such cases, however, we need a "wanted kind" of chaotic behavior with precisely prescribed features and/or we need techniques to switch between different kinds of behavior (chaos-order or chaos-chaos).

Considering the possibilities of influencing the dynamics of a chaotic circuit we can distinguish four basic approaches:

- variation of an existing accessible system parameter
- change in the system design-modification of its internal structure
- injection of an external signal(s)
- introduction of a controller (classical PI, PID, linear or nonlinear, neural, stochastic, etc.)

Because of the very rich dynamic phenomena encountered in typical chaotic systems, there are a large variety of approaches to controlling such systems. This article presents selected methods developed for controlling chaos in various aspects—starting from the most primitive concepts like

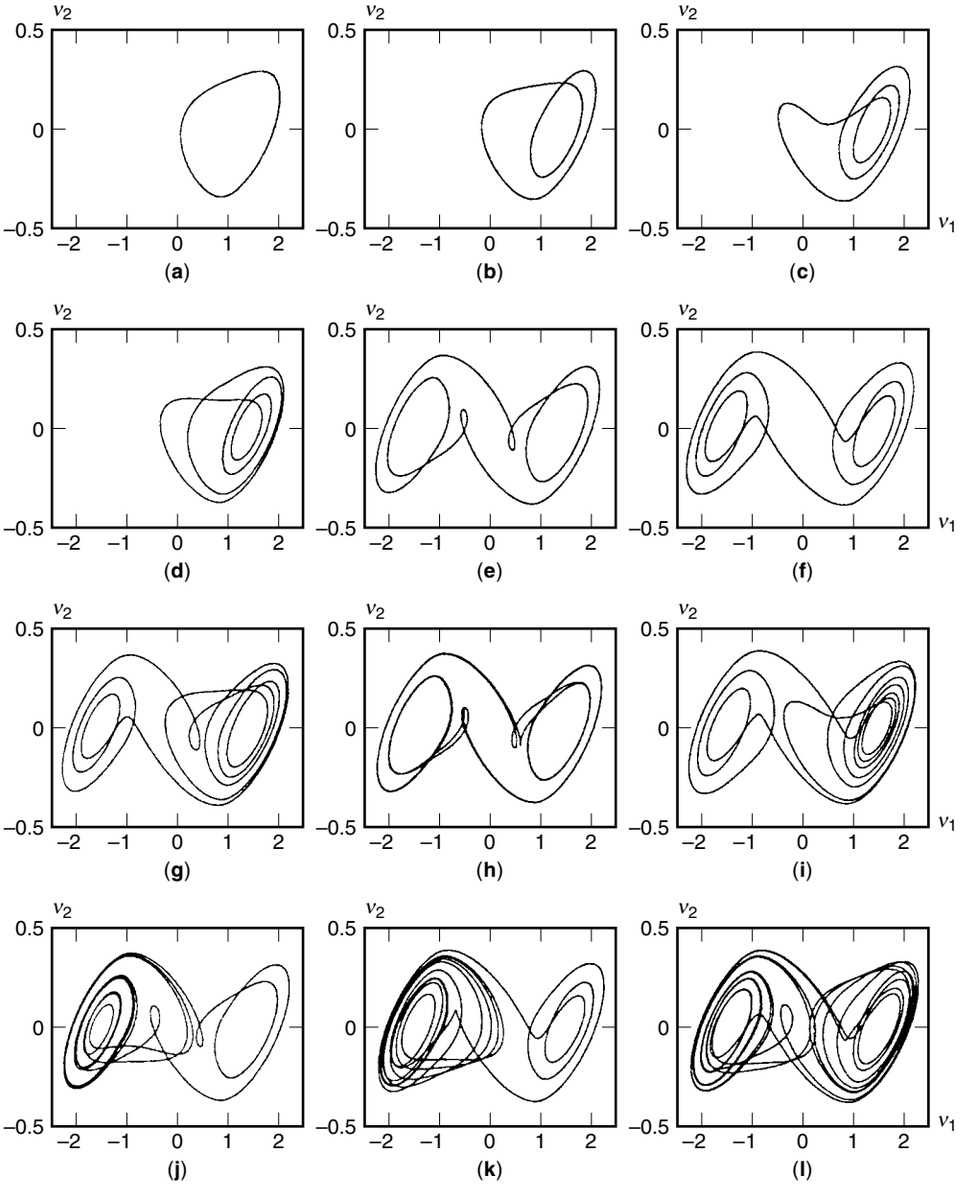


Figure 4. Second fundamental property of chaos. Within an attractor (visible in experiments and depicted in Fig. 2) an infinite but countable number of unstable periodic orbits exist. Such orbits are impossible to observe in experiments but can be detected using computer methods. In this picture some approximations to actual unstable periodic orbits are shown. These are uncovered using numerical calculations from time series measured for the double scroll attractor shown in Fig. 2. Notice the shape of the orbits—when superimposed these orbits reproduce the shape of the chaotic attractor.

parameter variation, through classical controller applications (open- and closed-loop control), to quite sophihsticated ones like stabilization of unstable periodic orbits embedded within a chaotic attractor.

GOALS OF CONTROL

As already mentioned, systems displaying chaotic behavior possess specific properties. Now we will exploit these properties when attacking the control problem. In what way does a chaotic system differ from any other object of control? How could its specific properties be advantageous for control?

The route to chaos via a sequence of bifurcations has two important implications for chaos control: first, it gives an insight into other accessible behaviors that can be obtained by changing parameters (this may be used for redesigning the system); second, stable and unstable orbits that are created or annihilated in bifurcations may still exist in the chaotic range and constitute potential goals for control.

Three fundamental properties of chaotic systems are of potential use for control purposes. For a long time the instabil-

ity property (sensitive dependence on initial conditions) has been considered the main obstacle for control. How can one visualize successful control if the dynamics may change drastically with small changes of the initial conditions or parameters? How can one produce a prescribed kind of behavior if errors in initial conditions will be exponentially amplified?

This fundamental property does not, however, necessarily mean that control is impossible. It has been shown that despite the divergence of nearby starting trajectories, they can be convergent to another prescribed kind of trajectory—one simply has to employ a different notion of stability. In fact, we do not require that the nearby trajectories converge—the requirement is quite different—the trajectories should merely converge to some goal trajectory $g(t)$

$$\lim_{t\to\infty} |x(t) - g(t)| = 0 \tag{1}$$

Depending on a particular application $g(t)$ could be one of the solutions existing in the system or any external waveform we would like to impose. Extreme sensitivity may even be of prime importance as control signals are in such cases very small.

The second important property of chaotic systems that will be exploited is the existence of a countable infinity of unstable periodic orbits within the attractor, already considered earlier. These orbits, although invisible during experiments, constitute a dense set supporting the attractor. Indeed, the trajectory passes arbitrarily close to every such orbit. This invisible structure of unstable periodic orbits plays a crucial role in many methods of chaos control; with specific methods the chaotic trajectory can be perturbed in such a way that it will stay in the vicinity of a chosen unstable orbit from the dense set.

These fundamental properties of chaotic signals and systems offer some very interesting issues for control not available in other classes of systems (2,3). Namely,

- because of sensitive dependence on initial conditions it is possible to influence the dynamics of the systems using very small perturbations; moreover, the response of the system is very fast
- the existence of a countable infinity of unstable periodic orbits within the attractor offers extreme flexibility and a wide choice of possible goal behaviors for the same set of parameter values

SUPPRESSING CHAOTIC OSCILLATIONS BY CHANGING SYSTEM DESIGN

Effects of Large Parameter Changes

The simplest way of suppressing chaotic oscillations is to change the system parameters (system design) in such a way as to produce the desired kind of behavior. The influence of parameter variations on the asymptotic behavior of the system can be studied using a standard tool for analysis of chaotic systems—the bifurcation diagram. The typical bifurcation diagram reveals a variety of dynamic behaviors for appropriate choices of system parameters and tells us what parameter values should be chosen to obtain the desired behavior. In electronic circuits, changes in the dynamic behavior are obtained by changing the value of one of its passive elements (which means replacing one of the resistors, capacitors, or inductors). In Fig. 3 a sample bifurcation diagram reveals a variety of dynamic behaviors observed in the *RC* chaos generator (4) (when changing one of the slopes of the nonlinear element). Thus when the generator is operating in a chaotic range, one can tune (control) it using a potentiometer to obtain a desired periodic state existing and displayed in the bifurcation diagram.

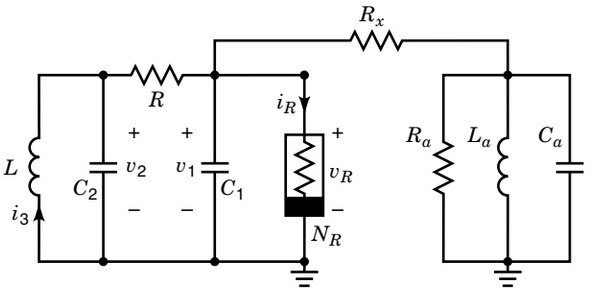


Figure 5. Chaos can be stabilized by adding a stabilizing subsystem to the chaotic one. As an example, a parallel *RLC* circuit is connected to the chaotic Chua's circuit and acts as a chaotic oscillation absorber.

This method, although intuitively simple, is hardly acceptable in practice; it requires large parameter variations (large energy control). This requirement cannot be met in many physical systems where the construction parameters are either fixed or can be changed over very small ranges. This method is also difficult to apply on the design stage as there are no simulation tools for electronic circuits allowing bifurcation analysis (e.g., SPICE has no such capability). On the other hand, programs offering such types of analysis require a description of the problem in closed mathematical form, such as differential or difference equations. Changes of parameters are even more difficult to introduce once the circuitry is fabricated or breadboarded, and if possible at all can be done only on a trial-and-error basis.

"Shock Absorber" Concept—Change in System Structure

This simple technique is being used in a variety of applications. The concept comes from mechanical engineering, where devices absorbing unwanted vibrations are commonly used (e.g., beds of machine-tools, shock absorbers in vehicle suspensions). The idea is to modify the original chaotic system design (add the "absorber" without major changes in the design or construction) in order to change its dynamics in such a way that a new stable orbit appears in a neighborhood of the original chaotic attractor. In an electronic system, the absorber can be as simple as an additional shunt capacitor or an *LC* tank circuit. Kapitaniak et al. (5) proposed such a "chaotic oscillation absorber" for Chua's circuit—it is a parallel *RLC* circuit coupled with the original Chua's circuit via a resistor (Fig. 5)—depending on its value the original chaotic behavior can be converted to a chosen stable oscillation. The equations describing dynamics of this modified system can be given in a dimensionless form:

$$\begin{aligned}
\dot{x} &= \alpha[y - x - g(x)] \\
\dot{y} &= x - y + z + \epsilon(y^1 - y) \\
\dot{z} &= -\beta y \\
\dot{y}' &= \alpha'[-\gamma' y' + z' + \epsilon(y - y')] \\
\dot{z}' &= -\beta' y'
\end{aligned} \tag{2}$$

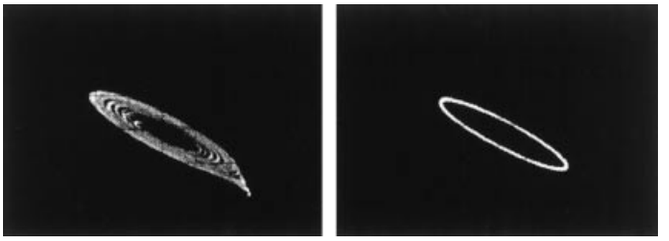


Figure 6. The "shock absorber" eliminates changes in the system behavior. For example, the spiral-type Chua's attractor can be quenched and a period-one orbit appears when parameters of the parallel *RLC* oscillation absorber, shown in Figure 5, are properly adjusted.

In terms of circuit equations, we have an additional set of two equations for the "absorber" (y^1, z^1) and a small term [$\epsilon(y^1 - y)$] through which the original equations of Chua's circuit are modified. Figure 6 shows the result of a laboratory experiment. Addition of a "shock absorber" in Chua's circuit changes chaotic behavior [Fig. 6(a)] to a periodic one [Fig. 6(b)].

EXTERNAL PERTURBATION TECHNIQUES

Several authors have demonstrated that a chaotic system can be forced to perform in a desired way by injecting external signals that are independent of the internal variables or structure of the system. Three types have been considered: (a) aperiodic signals ("resonant stimulation"), (b) periodic signals of small amplitude, and (c) external noise.

"Entrainment"—Open Loop Control

Aperiodic external driving is a classical control method and was one of the first methods introduced by Hübler (6,7) (resonant stimulation). A mathematical model of the considered experimental system is needed (e.g., in the form of a differential equation: $dx/dt = F(x)$, $x \in R^n$, where $F(x)$ is differentiable and a unique solution exists for every $t \geq 0$).

The goal of the control is to entrain the solution $x(t)$ to an arbitrarily chosen behavior $g(t)$:

$$\lim_{t\to\infty} |x(t) - g(t)| = 0 \tag{3}$$

Entrainment can be obtained by injecting the control signal:

$$\frac{dx}{dt} = F(x) + [\dot{g} - F(g)]1(t) \tag{4}$$

where $1(t)$ is 0 for $t < 0$ and 1 for $t > 0$. The entrainment method has the advantage that no feedback is required and no parameters are changed—thus the control signal can be computed in advance and no equipment for measuring the state of the system is needed. The goal does not depend on the system being considered, and in fact it could be any signal at all (except that solutions of the autonomous system since $\dot{g} - F(g) \equiv 0$ in this case, and there is no control signal). It should be noted, however, that this method has limited applicability since a good model of the system dynamics is necessary, and the set of initial statistics for which the system trajectories will be entrained is not known.

Weak Periodic Perturbation

Interesting results have been reported by Breiman and Goldhirsch (8), who studied the effects of adding a small periodic driving signal to a system behaving in a chaotic way. They discovered that external sinusoidal perturbation of small amplitude and appropriately chosen frequency can eliminate chaotic oscillations in a model of the dynamics of a Josephson junction and cause the system to operate in some stable periodic mode. Unfortunately, there is little theory behind this approach and the possible goal behaviors can be learned only by trial and error. Some hope for further understanding and applications can be based on using theoretical results known from the theory of synchronization.

Noise Injection

A noise signal of small amplitude injected in a suitable way into the circuit (system) offers potentially new possibilities for stabilization of chaos. The first observations date back to the work of Herzel (9). The effects of noise injection were also studied in an *RC*-ladder chaotic oscillator (10). In particular it has been observed that injection of noise of sufficiently high level can eliminate multiple domains of attraction. In the experiments with the *RC*-ladder chaos generator it has been found that the two main branches, representing two distinct, coexisting solutions, as shown in Fig. 3, will join together if white noise of high level is added. This approach, although promising, needs further investigation because there is little theory available to support experimental observations.

CONTROL ENGINEERING APPROACHES

Several investigators have tried to use known methods belonging to the "control engineer's toolkit." For example, PI and PID controllers for chaotic circuits, applications of stochastic control techniques, Lyapunov-type methods, robust controllers, and many other methodologies, including intelligent control and neural controllers, have been described in the literature. Chen and Dong (11) and Chapter 5 in Madan's book (12) give an excellent review of applications of such methods. In electronic circuits two schemes—linear feedback and time-delay feedback—seem to find the most successful applications.

Error Feedback Control

Several methods of chaos control have been developed that rely on the common principle that the control signal is some function ϕ of the difference between the actual system output $x(t)$ and the desired goal dynamics $g(t)$. This control signal could be an actual system parameter:

$$p(t) = \phi[x(t) - g(t)] \tag{5}$$

or an additive signal produced by a linear controller:

$$u(t) = K[x(t) - g(t)] \tag{6}$$

The control term is simply added to the system equations. One can readily see that, although mathematically simple, such an "addition" operation might pose serious problems in real applications. The block diagram of the control scheme is

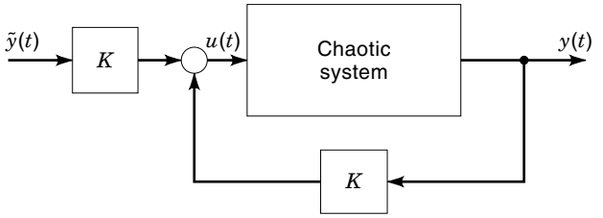


Figure 7. Standard control engineering methods can be used to stabilize chaotic systems, for example the linear feedback control scheme proposed by Chen and Dong, shown here.

shown in Fig. 7. Using error feedback, chaotic motion has been successfully converted into periodic motion both in discrete- and continuous-time systems. In particular, chaotic motions in Duffing's oscillator and Chua's circuit have been controlled (directed) toward fixed points or periodic orbits (11). The equations of the controlled circuit read:

$$\begin{aligned} \dot{x} &= \alpha[y - x - g(x)] \\ \dot{y} &= x - y + z - K_{22}(y - \tilde{y}) \\ \dot{z} &= -\beta y \end{aligned} \tag{7}$$

Thus we have a single term added to the original equations.

Figure 8 shows a double scroll Chua's attractor and large saddle-type unstable periodic orbit toward which the system has been controlled.

The important properties of the linear feedback chaos control method are that the controller has a very simple structure and that access to the system parameters is not required. The method is immune to small parameter variations but might be difficult to apply in real systems (interactions of many system variables are needed). The choice of the goal orbit poses the most important problem; usually the goal is chosen in multiple experiments or can be specified on the basis of model calculations.

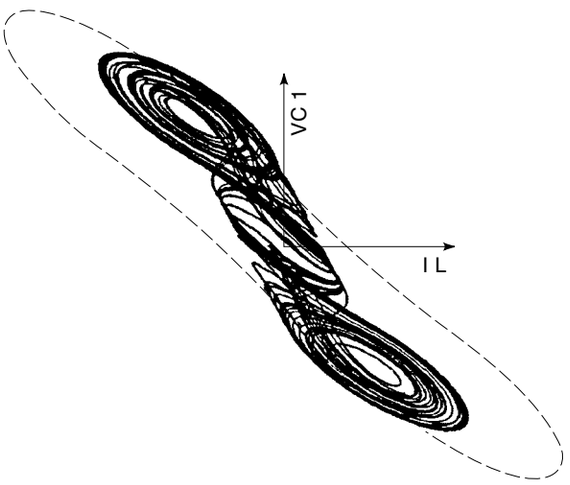


Figure 8. Linear feedback method in many cases enables stabilization of a simple orbit which is a solution of the system. For example, the double scroll (chaotic) attractor and a saddle type unstable periodic orbit coexist in Chua's circuit. This periodic orbit can be stabilized using linear feedback.

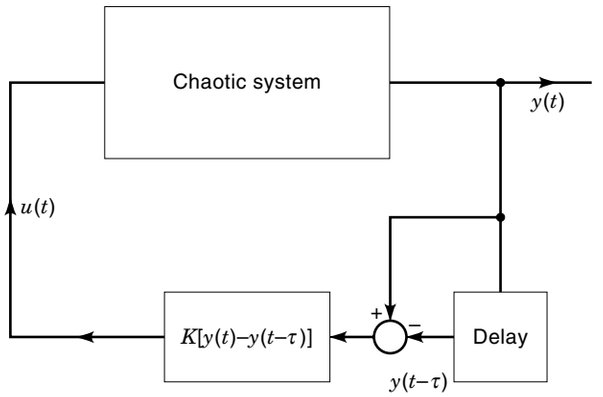


Figure 9. Block diagram of the delay feedback control scheme proposed by Pyragas. Injection of signal proportional to the difference between the original output and its delayed copy can stabilize operation of a chaotic system when the time delay and gain in the feedback loop are chosen appropriately.

STABILIZING UNSTABLE PERIODIC ORBITS

Time-Delay Feedback Control (Pyragas Method)

An interesting method has been proposed by Pyragas (13). The control signal applied to the system is proportional to the difference between the output and a delayed copy of the same output:

$$\frac{dx}{dt} = F[x(t)] + K[y(t) - y(t - \tau)] \tag{8}$$

Tuning the delay τ one can approach many of the periods of the unstable periodic orbits embedded within the chaotic attractor. In such a situation, the control signal approaches 0. A block diagram of the control scheme is shown in Fig. 9. Depending on the delay constant τ and the linear factor K, various kinds of periodic behaviors can be observed in the chaotic system. In the case of Chua's circuit we were able, for example, to convert chaotic motion into a periodic one, as shown in Fig. 10.

Pyragas obtained very promising results in the control of many different chaotic systems, and despite the lack of mathematical rigor, this method is being successfully used in several applications.

An interesting application of this technique is described by Mayer-Kress et al. (14). Pyragas's control scheme has been used for tuning chaotic Chua's circuits to generate musical

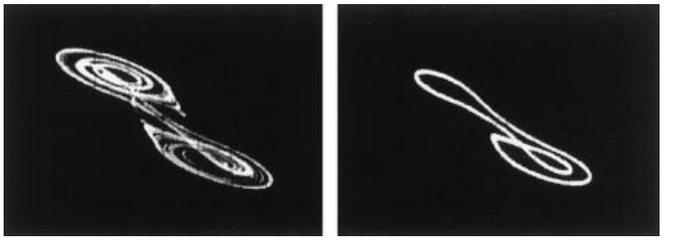


Figure 10. The double scroll attractor can be eliminated and the behavior converted to one of the periodic orbits in experiments in the delayed feedback control of Chua's circuit.

tones and signals. More recently Celka (15) used Pyragas's method to control a real electrooptical system.

The positive features of the delay feedback control method are that no external signals are injected and no access to system parameters is required. Any of the unstable periodic orbits can be stabilized provided that delay is chosen in an appropriate way. The control action is immune to small parameter variations. In real electronic systems, the required variable delay element is readily available (for example, analog delay lines are available as off-the-shelf components). The primary drawback of the method is that there is no a priori knowledge of the goal (the goal is arrived at by trial and error).

Ott–Grebogi–Yorke Local Linearization Approach

Ott, Grebogi, and Yorke (16,17) in 1990 proposed a feedback method to stabilize any chosen unstable periodic orbit within the countable set of unstable periodic orbits existing in the chaotic attractor. To visualize best how the method works, let us assume that the dynamics of the system are described by a k-dimensional map:$x_{n+1} = F(x_n, p)$, $x_i \in R^k$. This map, in the case of continuous-time systems, can be constructed (e.g., by introducing a transversal surface of section for system trajectories, p is some accessible system parameter that can be changed in some small neighborhood of its nominal value p^*). To explain the method we will concentrate now on stabilization of a period-one orbit. Let $x_F = F(x_F, p^*)$ be the chosen fixed point (period one) of the map around which we would like to stabilize the system. Assume further that the position of this orbit changes smoothly with p parameter changes (i.e., p^* is not a bifurcation value) and there are small changes in the local system behavior for small variatons of p. In a small vicinity of this fixed point we can assume with good accuracy that the dynamics are linear and can be expressed approximately by:

$$x_{n+1} - x_0 = A(x_n - x_0) + g(p_n - p^*) \tag{9}$$

The elements of the matrix $A = \partial F/\partial x\ (x_F, p^*)$ and vector $g = \partial F/\partial p\ (x_F, p^*)$ can be calculated using the measured chaotic time series and analyzing its behavior in the neighborhood of the fixed point. Further, the eigenvalues λ_s, λ_u and eigenvectors e_s, e_u of this matrix can be found

$$Ae_u = \lambda_u e_u \quad \text{and} \quad Ae_s = \lambda_s e_s \tag{10}$$

where the subscripts "u" and "s" correspond to unstable and stable directions respectively. These eigenvectors determine the stable and unstable directions in the small neighborhood of the fixed point (Fig. 11).

$$A = [e_u \quad e_s] \begin{bmatrix} \lambda_u & 0 \\ 0 & \lambda_s \end{bmatrix} [e_u \quad e_s]^{-1} \tag{11}$$

Let us denote by f_s, f_u the contravariant eigenvectors [$f_s^T e_s = f_u^T e_u = 1$, $f_s^T e_u = f_u^T e_s = 0$; see Fig. 11(c)]. Thus

$$A = [e_u \quad e_s] \begin{bmatrix} \lambda_u & 0 \\ 0 & \lambda_s \end{bmatrix} \begin{bmatrix} f_u^T \\ f_s^T \end{bmatrix} = \lambda_u e_u f_u^T + \lambda_s e_s f_s^T \tag{12}$$

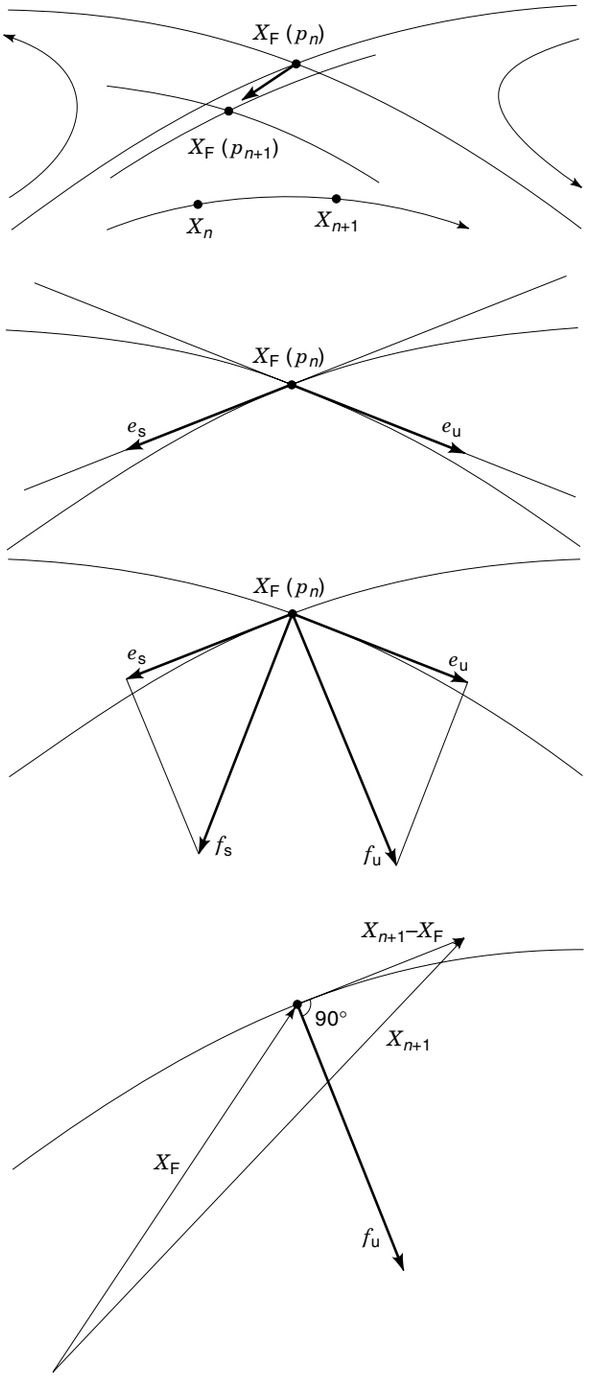


Figure 11. Explanation of the linearization technique used by the Ott–Grebogi–Yorke chaos stabilization method. (a) Parameter change causes displacement of the fixed point. In a small neighborhood of the fixed point the behavior of trajectories and displacement of the fixed point can be considered as linear. (b) Stable and unstable eigenvectors of the linearization matrix A. (c) New contravariant basis vectors. (d) Action of the control—the trajectory is forced to move onto the stable manifold of the fixed point.

This implies that f_u^T is a left eigenvector of A with the same eigenvalue e_u:

$$f_u^T A = f_u^T (\lambda_u e_u f_u^T + \lambda_s e_s f_s^T) = \lambda_u f_u^T \tag{13}$$

The control idea (16–18) now is to monitor the system behavior until it comes close to the desired fixed point (we assume that the system is ergodic and the trajectory fills the attractor densely; thus eventually it will pass arbitrarily close to any chosen point within the attractor) and then change p by a small amount so the next state x_{n+1} should fall on the stable manifold of x_0 [i.e., choose p_n such that $f_u^T(x_{n+1} - x_F) = 0$]:

$$p_n = -\left(\frac{\lambda_u}{f_u^T g}\right) f_u^T (x_n - x_F) + p^* \tag{14}$$

which can be expressed as a local linear feedback action:

$$p_{n+1} = p_n + C f_u^T [x_n - x_F(p_n)] \tag{15}$$

The actuation of the value of the control signal to be applied at the next iterate is porportional to the distance of the system state from the desired fixed point $[x_n - x_F(p_n)]$ projected onto the perpendicular unstable direction f_u. The constant C depends on the magnitude of the unstable eigenvalue λ_u and the shift g of the attractor position with respect to the change of the system parameter projected onto the unstable direction f_u. The Ott–Grebogi–Yorke (OGY) technique has the notable advantage of not requiring analytical models of the system dynamics and is well-suited for experimental systems. One can use either the full information from the process of the delay coordinate embedding technique using single variable experimental time series [see Dressler and Nitsche (19)]. The procedure can also be extended to higher-period orbits. Any accessible variable (controllable) system parameter can be used for applying perturbation, and the control signals are very small. The method also has several limitations. Its application in multiattractor systems is problematic. It is sensitive to noise, and the transients before achieving control might be very long in many cases. We have carried out an extensive study of application of the OGY technique to controlling chaos in Chua's circuit (12). Using an application-specific software package (20), we were able to find some of the unstable periodic orbits embedded in the double scroll Chua's chaotic attractor and use them as control goals.

Figure 12 shows the time evolution of the voltages when attempting to stabilize unstable period-one orbit in Chua's circuit. Before control is achieved, the trajectories exhibit chaotic transients before entering the close neighborhood of the chosen orbit.

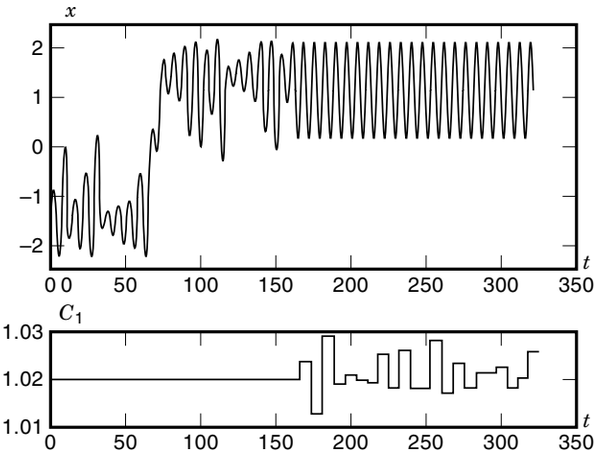


Figure 12. Typical results of stabilization of a period-one orbit in Chua's circuit using the OGY method. Time-waveform of voltage across the C_1 capacitor and variations of the control signal are shown.

Sampled Input Waveform Method

A very simple, robust, and effective method of chaos control in terms of stabilization of an unstable periodic orbit has been proposed (21). A sampled version of the output signal, corresponding to a chosen unstable periodic trajectory uncovered from a measured time series, is applied to the chaotic system causing the system to follow this desired orbit. In real systems, this sampled version of the unstable periodic orbit can be programmed into a programmable waveform generator and used as the forcing signal.

The block diagram of this control scheme is shown in Fig. 13. For controlling chaos in Chua's circuit (compare the circuit diagram shown as the left-side subcircuit in Fig. 5) we try to force the system with a sampled version of a signal $\hat{V}_1(t)$ [$(\hat{V}_1(t) = C^T\hat{x}(t)$]. Forcing the system with a continuous signal $\hat{V}_1(t)$ will force the system to exhibit a solution $x(t)$, which tends asymptotically toward $\hat{x}(t)$. This is obvious since forcing $V_1(t)$ will instantaneously force the current through the piecewise linear resistance to a "desired" value $i_R(t)$. The remaining subcircuit (R, L, C_2), which is an RLC stable circuit, will then exhibit a voltage $V_2(t)$ and a current $i_3(t)$, which will asymptotically converge towards $\hat{V}_2(t)$ and $\hat{i}_3(t)$.

The sampled input control method is very attractive as the goal of the control can be specified using analysis of the output time-series of the system; access to system parameters is not required. The control technique is immune to parameter variations, noise, scaling, and quantization. Instead of a controller, we need a generator to synthesize the goal signal. Signal sampling reduces the memory requirements for the gener-

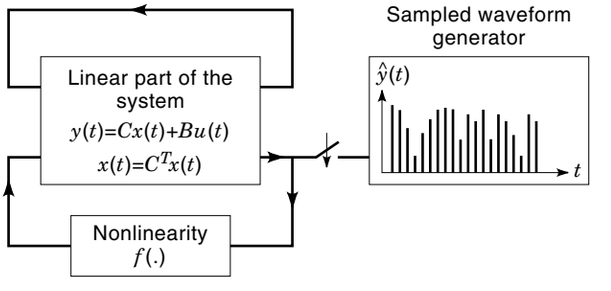


Figure 13. Block diagram of the sampled input chaos control system. A sampled version of a periodic signal corresponding to an unstable orbit uncovered from measured output is used to force the chaotic system which here has a special structure. This structure consists of a stable linear part and a scalar, static nonlinearity in the feedback path. Forcing signal is applied to the input of the nonlinearity.

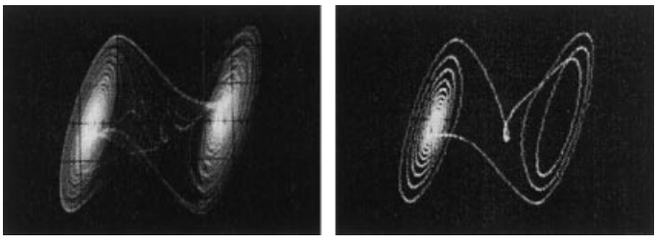


Figure 14. Using the sampled input forcing the double scroll attractor (a) observed in the experimental system can be converted into a long periodic orbit (b) stabilized during laboratory experiments.

ator. Figure 14 shows the chaotic attractor and two sample orbits controlled within the chaos range.

CHAOS CONTROL BY OCCASIONAL PROPORTIONAL FEEDBACK

In real applications, a "one-dimensional" version of the OGY method—the occasional proportional feedback (OPF) method—has proved to be most efficient. To explain the action of the OPF method let us consider a return map as shown in Fig. 15. For present consideration we take an approximate one-dimensional map obtained for the *RC*-ladder chaos generator (4). For nominal parameter values the position of the graph of the map is as shown by the rightmost curve; all periodic points are unstable. In particular, the point P is an unstable equilibrium. Looking at the system operation starting from point v_n, at the next iteration (the next passage of the trajectory through the Poincaré plane) one would obtain v_{n+1}. We would like to direct the trajectories toward the fixed point P. This can be achieved by changing a chosen system parameter such that the graph of the return map moves to a new position as marked on the diagram, thus forcing the next iteration to fall at v^*_{n+1}; after this is done the perturbation can be removed and activated again if necessary.

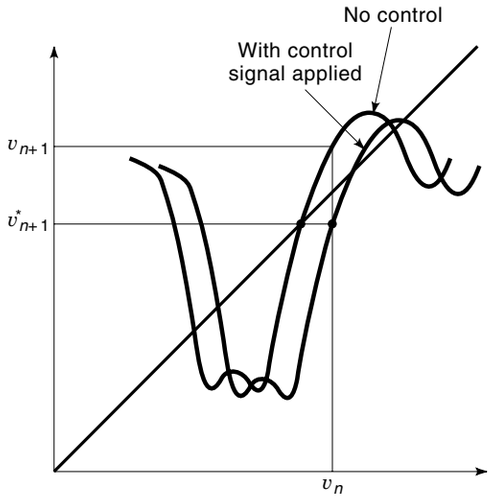


Figure 15. Explanation of the action of the occasional proportional feedback method using a graph of the first return map. Variation of an accessible system parameter causes displacement of the graph—when the control signal is chosen appropriately this displacement can be such that from a given coordinate the next iterate will fall exactly onto the unstable fixed point.

In mathematical terms we can compute the control signal using only one variable, for example ξ_1:

$$p(\xi) = p_0 + c(\xi_1 - \xi_{F1}) \tag{16}$$

This method has been successfully implemented in a continuous-time analog electronic circuit and used in a variety of applications ranging from stabilization of chaos in laboratory circuits (22–24) to stabilization of chaotic behavior in lasers (25–27). The OPF method may be applied to any real chaotic system (also higher-dimensional ones) where the output can be measured electronically and the control signal can be applied via a single electrical variable. The signal processing is analog and therefore is fast and efficient. Processing in this case means detecting the position of a one-dimensional projection of a Poincaré section (map), which can be accomplished by the window comparator, taking the input waveform. The comparator gives a logical high when the input waveform is inside the window. A logical AND operation is performed on this signal and on the delayed output from the external frequency generator. This logical signal drives the timing block that triggers the sample-and-hold and then the analog gate. The output from the gate, which represents the error signal at the sampling instant, is then amplified and applied to the interface circuit that transforms the control pulse into a perturbation of the system. The frequency, delay, control pulse width, window position, width, and gain are all adjustable. The interface circuit used depends on the chaotic system under control.

One of the major advantages of Hunt's controller over OGY is that the control law depends on only one variable and does not require any complicated calculations in order to generate the required control signal. The disadvantage of the OPF method is that there is no systematic method for finding the embedded unstable orbits (unlike OGY). The accessible goal trajectories must be determined by trial and error. The applicability of the control strategy is limited to systems in which the goal is suppression of chaos without more strict requirements.

IMPROVED ELECTRONIC CHAOS CONTROLLER

Recently, in collaboration with colleagues from University College, Dublin, we have proposed an improved electronic chaos controller that uses Hunt's method without the need for an external synchronizing oscillator. Hunt's OPF controller used the peaks of one of the system variables to generate the 1^D map. Hunt then used a window around a fixed level to set the region where control was applied. In order to find the peaks, Hunt's scheme used a synchronizing generator. In our modified controller (28,29), we simply take the derivative of the input signal and generate a pulse when it passes through zero. We use this pulse instead of Hunt's external driving oscillator as the "synch" pulse for our Poincaré map. This obviates the need for the external generator and so makes the controller simpler and cheaper to build.

The variable level window comparator is implemented using a window comparator around zero and a variable level

shift. Two comparators and three logic gates form the window around zero. The synchronizing generator used in Hunt's controller is replaced by an inverting differentiator and a comparator. A rising edge in the comparator's output corresponds to a peak in the input waveform. We use the rising edge of the comparator's output to trigger a monostable flip-flop. The falling edge of this monostable's pulse triggers another monostable, giving a delay. We use the monostable's output pulse to indicate that the input waveform peaked at a previous fixed time. If this pulse arrives when the output from the window comparator is high then a monostable is triggered. The output of this monostable triggers a sample-and-hold on its rising edge that samples the error voltage; on its falling edge, it triggers another monostable. This final monostable generates a pulse that opens the analog gate for a specific time (the control pulse width). The control pulse is then applied to the interface circuit, which amplifies the control signal and converts it into a perturbation of one of the system parameters, as required.

We tested our controller using a chaotic Colpitts oscillator (30) and laboratory implementation of Chua's circuit. Implementation of a laboratory Chua's circuit together with interface circuit to connect the controller is shown in Fig. 16. Figure 17 shows an example of stabilization of a period-four orbit (found by trial-and-error search) using the improved chaos controller. In Fig. 18 we show oscilloscope traces for the goal trajectory and the control signal (bottom trace). It is interesting to note the impulsive action of the controller.

CHAOS-TO-CHAOS CONTROL

Synchronization of a given system solution with an externally supplied chaotic signal can be considered a particular type of control problem. The goal of the control scheme is to track (follow) the desired (input) chaotic trajectory. In particular, the input signal might come from an identical copy of the considered system, the only difference being the initial conditions. It is only very recently that such a control problem has been recognized in control engineering. The linear coupling technique and the linear feedback approach to controlling chaos can be applied for obtaining any chosen goal—regardless of whether it is chaotic, periodic, or constant in time. For a review of the chaos synchronization concepts and applications we refer the reader to Ogorzalek (31).

One can also envisage controlling a chaotic system toward chaotic targets that are not solutions of the system itself (goals might be chaotic trajectories originating from different systems). An impressive example of this kind of control/influence could be in generating Lorenz-like behavior in Chua's circuit (32). We believe that this kind of chaotic synchronization—control to a chaotic goal—could lead to new developments and possibly new applications of chaotic systems.

CONTROL OF SPATIOTEMPORAL CHAOTIC SYSTEMS

Chaos control becomes much more complicated in the case of large coupled and possibly very high-dimensional systems (such as neural networks), spatiotemporal systems (governed by partial differential equations or time-delay equations), because there exists a very rich repertoire of spatiotemporal behaviors depending on parameters of the system, architecture of interconnections, and external signals applied to it. It is believed that chaos control concepts in spatiotemporal systems might give explanations for the functioning of the brain. In controlling spatiotemporal systems we should consider first of all the goals we would like to achieve—they may be different in this case from the goals considered so far (stabilization of periodic orbits or anticontrol toward a desired chaotic waveform). In particular one can consider:

1. Formation of specific spatial or spatiotemporal patterns; influence on the spatial patterns might be needed, for example, in models of crystal growth, memory patterns, creation of waves with prescribed characteristics, and so on.
2. Stabilization of wanted behavior; this kind of operation might be required, for example, in the case of associative memory.
3. Synchronization/desynchronization; in some cases it might be desirable to obtain a coherent operation of the whole spatial structure or a part of the cells only. One can also envisage "anticontrol" desynchronization, as in the case of epileptic foci and recovery of normal brain functioning.
4. Efficient switching between attractors; we should envisage this kind of goal in the models of brain functions: change of concentration on various objects is linked with attractor switchings.
5. Removal of a specific type of behavior (e.g., spiral waves; this is a medical application such as defibrillation).
6. Cluster stabilization; in this kind of approach only a small spatial cluster in the multidimensional medium is to be stabilized while all the surrounding medium has to operate in a chaotic mode.

There is also more flexibility in applying control signals—they might be applied at the borders, at every cell, at specific locations in space, and so on. Also, connections between the cells in the network might be varied in some cases.

Coupled Map Lattices

A coupled map lattices (CML) system is a good target to study the control of spatiotemporal chaos because of existence of very rich spatiotemporal chaotic behavior in the control-free CML (33). In controlling a one-dimensional CML, stabilizing the system from spatiotemporal chaos not only to homogeneous stationary states but also to periodic states both in space and time has been demonstrated already (34). The idea of pinnings (putting some local control) plays a very important role in stabilizing spatiotemporal chaos. One advantage of the pinnings is to avoid the overflow in numerical simulation. Moreover, Hu and Qu have reported that a lower pinning density shows better control performance than a higher one in numerical experiments (34). Further analysis is needed of the relationship between the pinning density and control performance (34,35).

An important application of controlling CML is to suppress or skip very long transient chaotic (sometimes called "supertransient") waveforms (34). Such phenomena are often observed in CML systems, and sometimes one cannot see the

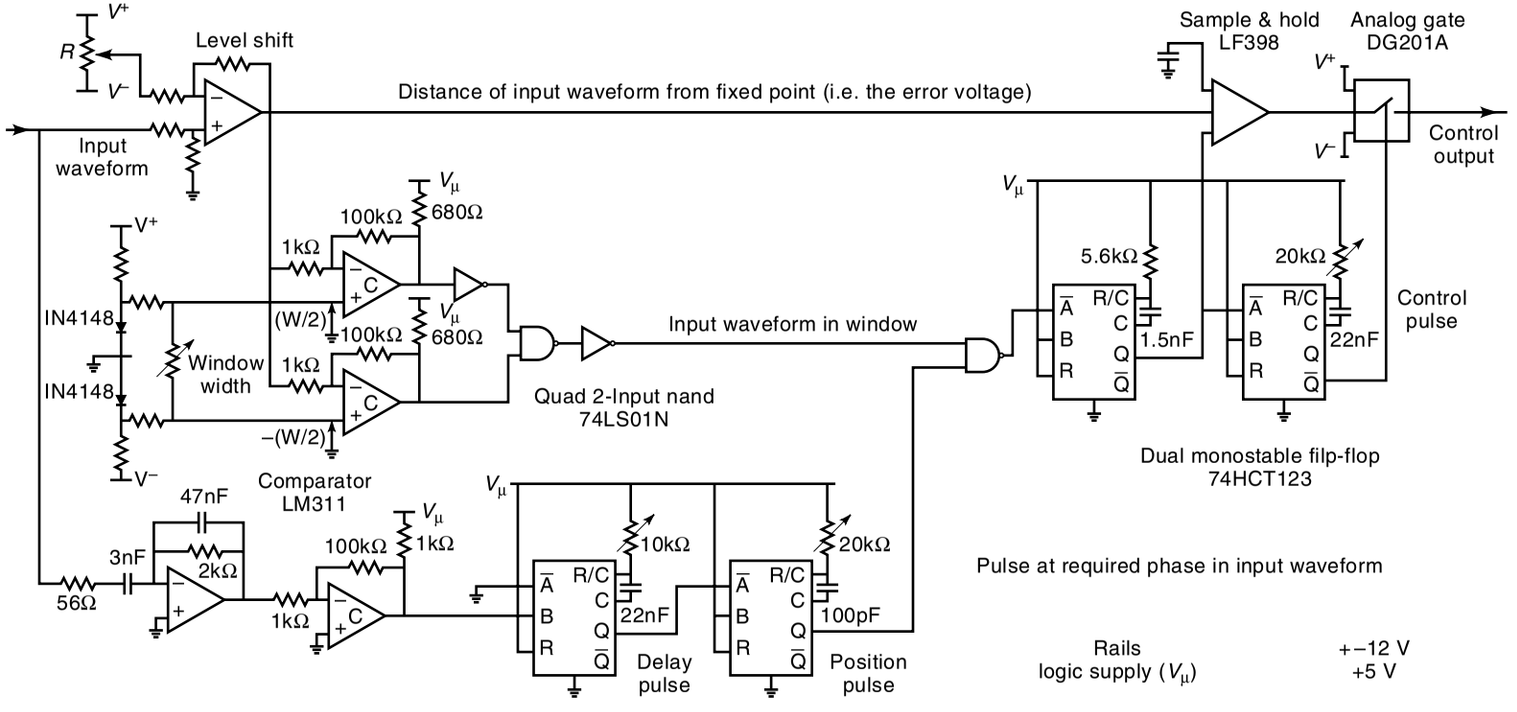


Figure 16. Improved analog chaos occasional proportional feedback controller without external synchronization.

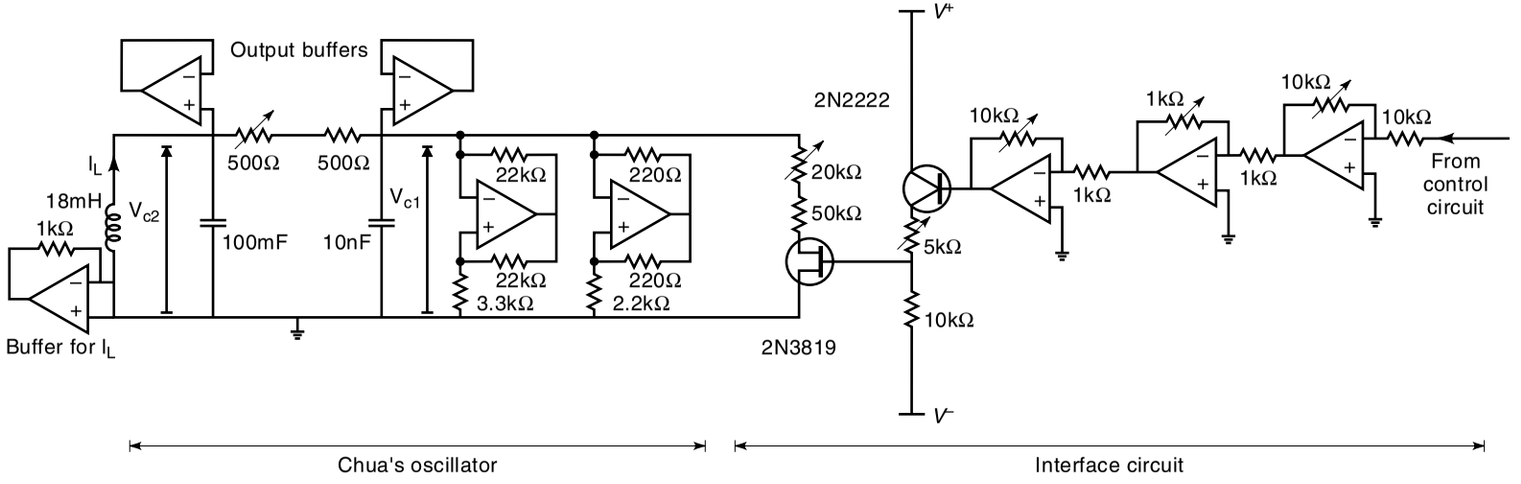


Figure 17. Circuit diagram for the implementation of Chua's circuit and the interface circuit. The interface circuit is specific for the considered chaotic system. Controller circuitry, as shown in Figure 16, is universal.

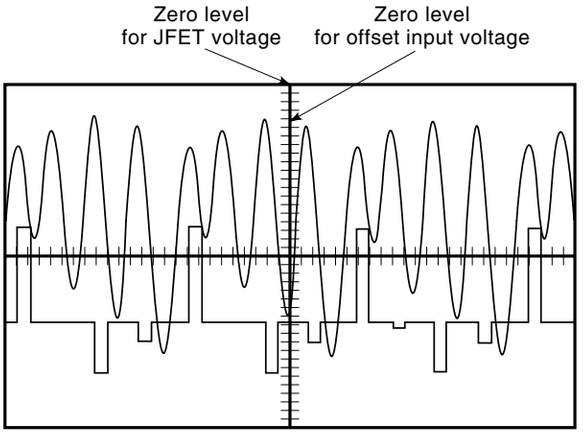


Figure 18. Oscilloscope traces of period-four solution stabilized in Chua's circuit and controlling signal produced by the improved chaos controller.

steady state for millions or more of iterations in numerical experiments. However, how to determine the desired (target) state of control in suppressing or skipping such transient chaos is still an open problem.

Spatial and Temporal Modulation of Extended Systems

The effects of global spatial and temporal modulation on pattern-forming systems have been widely studied. Global modulation means here that control signals are applied to every cell throughout the network. Examples of effects of this type of stimulation/control include pattern instability under periodic spatial forcing, spatial disorder induced in an autowave medium (Belousov–Zhabotinsky reaction), continuous variation of the wavelength of a pattern, or transitions between structures with incommensurate wavelengths [see Perez-Meñuzuri et al. (36) for a good list of references]. This global control method remains purely empirical.

Introducing Disorder to Tame Chaos

Interesting observations have been made recently by Braiman et al. (37). Based on earlier observations that noise injection can remove chaos in low-dimensional systems, they proposed to introduce uncorrelated differences between chaotic oscillators coupled in a large array. They identified two mechanisms by which disorder can stabilize chaos. The first requires small disorder and relies on disturbance of the system "position" in a very high-dimensional parameter space, resulting in change of the observed attractor. The second mechanism requires large perturbations; removing some of the oscillators in the array from their initial chaotic regime can possibly trigger the whole array into orderly behavior. The experiments of Braiman and others suggest that spatial disorder might be one of control mechanisms of pattern formation and self-organization.

Turing Patterns: Defect Removal

Perez-Meñuzuri et al. (36) studied creation of Turing patterns in arrays of discretely coupled dynamic systems. They discovered spontaneous creation of hexagonal or rhombic patterns when systems parameters were adjusted in some specific way. In many cases the observed patterns were not perfectly homogeneous (symmetrical). It turned out from several experiments that the defect can be removed by external side-wall stimulation—boundary control. These experiments demonstrate a potential principle for influencing crystal growth to obtain perfect structures. The control strategy applied in this case is a local one—only boundaries of the network are being excited (in contrast to global modulation).

Control of the Model Cortex

Babloyantz et al. (38) considered applications of feedback control of the Pyragas type to include control mechanisms in a model cortex. They studied a model in which all cells have linear dynamics but the connections are nonlinear of the sigmoid type. A single stabilizable periodic orbit that corresponds to bulk oscillations of the network has been found. Neurological data suggest that synchronized states in the brain are triggered when external stimuli are applied. Based on the simulation experiments, the authors proposed the following theory for attentiveness: it results from momentary (short time scale) control of chaotic activity observed in the cerebral cortex. Since the number of neural cells in the cortex is in the range of 10^{11}, the number of different stabilizable spatiotemporal patterns must be enormous and we can easily imagine that each stimulus can stabilize its corresponding characteristic state. Attentiveness, concentration, and recognition of patterns as well as wakefulness and sleep could be explained in terms of chaos control processes.

Controlling Autowaves: Spatial Memory

A particular type of pattern formation and self-organization in arrays of chaotic systems is autowaves (39). Development of autowaves in an array of chaotic oscillators can be controlled in several ways. First, adjustment of coupling between the oscillators gives a global control mechanism for dynamic phenomena. Second, when the network is operating in an autowave regime, one can observe the memory effect (39): The position of external stimulation controls the form of the observed spatial pattern. Finally, noise injection can destroy or quench patterns, introducing disorder.

Control of Ventricular Fibrillation: Quenching of Spiral Waves

Creation of spiral waves in heart tissue is now believed to be the principal cause of many arrythmias and heart disorders, including often-fatal ventricular fibrillation. Avoiding situations leading to spiral and scroll waves and eventually quenching such developing waves are of paramount importance in cardiology. Biktashev and Holden (40) proposed a feedback version of the resonant drift phenomenon (i.e., directed motion of the autowave vortex by applying an external signal) to remove the unwanted phenomena. Simulation studies confirm that amplitudes of signals needed for defibrillation using the proposed method are substantially less than those of conventional single-shock techniques used currently in medical practice.

Boundary and Defect-Induced Control in a Network of Chua's Circuits

An extensive simulation study has been carried out to discover the possibilities of controlling pattern formation in CNN

(cellular neural network) arrays composed of chaotic Chua's circuits. The open-loop control strategy has been applied at the edge cells only. Thus by the number of cells excited the formation of wavefronts and their shape can easily be modified. Furthermore, it has been found out that the introduction of defects in the network could serve as a means of inducing spiral wave formation with the "tips" positioned at some prescribed locations.

Chaotic Neural Networks

Aihara (41,42) has proposed a neural network model composed of simple mathematical neurons, which are described by difference equations, and exhibit chaotic dynamics. Chaos control in such chaotic neural networks may be useful to improve the performance of the associative memory and to solve optimization problems. Control of a simple chaotic neural network has been reported (43). It has also reported that chaotic neural networks that have global or nearest-neighbor coupling can be controlled by a modified exponential control method (44). However, these results are not sufficient for the applications of controlling chaos mentioned previously because these results are only on the networks with homogeneous synaptic weights (couplings). In order to apply controlling chaos to the networks for associative memory and solving optimization problems, development of control methods for large-scale chaotic neural networks with inhomogeneous synaptic weights is needed.

ELECTRONIC CHAOS CONTROLLERS

The widespread interest in chaos control is due to its extremely interesting and important possible applications. These applications range from biomedical ones (e.g., defibrillation or blocking of epileptic seizures), through solid-state physics, lasers, aircraft wing vibrations and even weather control, just to name a few attempts made so far. Looking at the possible applications alone it becomes obvious that chaos control techniques and their possible implementations will greatly depend on the nature of the process under consideration. From the control implementation perspective, real systems exhibiting chaotic behavior show many differences. The main ones are (45):

- speed of the phenomenon (frequency spectrum of the signals)
- amplitudes of the signals
- existence of corrupting noises, their spectrum and amplitudes
- accessibility of the signals to measurement
- accessibility of the control (tuning) parameters
- acceptable levels of control signals

In most cases, electronic equipment will play a crucial role. In some applications, like the biomedical ones, we would possibly need implantable devices. In looking for an implementation of a particular chaos controller, we must first look at these system-induced limitations. How can we measure and process signals from the system? Are there any sensors available? Are there any accessible system variables and parameters that could be used for the control task? How do we choose the ones that offer the best performance for achieving control? What devices can be used to apply the control signals? Can we make off-line computations? At what speed do we need to compute and apply the control signals? What is the lowest acceptable precision of computation? Can we achieve control in real time?

A slow system like a bouncing magneto-elastic ribbon (with eigenfrequencies below 1 Hz) is certainly not as demanding as a telecommunications channel (possibly running at GHz) or a laser for control.

In electronic implementations, one must look at several closely linked areas: sensors (for measurements of signals from a chaotic process), electronic implementation of the controllers, computer algorithms (if computers are involved in the control process), and actuators (introducing control signals into the system). External to the implementation (but directly involved in the control process and usually fixed using the measured signals) is determining the goal of the control.

Despite the many methods that have been developed and described in the literature (3,11,46), most are still only of academic interest because of the lack of success in implementation. A control method cannot be accepted as successful if computer simulation experiments are not followed by further laboratory tests and physical implementations. Only very few results of such tests are known; among the exceptions are: the control of a green-light laser (27), the control of a magnetoelastic ribbon (47), and a few other examples.

Implementation Problems for the OGY Method

When implementing the OGY method for a real-world application one must perform the following series of elementary operations (45):

1. Data acquisition—measurement of a (usually scalar) signal from the chaotic system under consideration. This operation should be performed in such a way as not to disturb the existing dynamics. For further computerized processing, measured signals must be sampled and digitized (A/D conversion).
2. Selection of appropriate control parameter
3. Finding unstable periodic orbits using experimental data (measured time series) and fixing the goal of control
4. Finding parameters and variables necessary for control
5. Application of the control signal to the system; this step requires continuous measurement of system dynamics in order to determine the moment at which to apply the control signal (i.e., the moment when the actual trajectory passes in a small vicinity of the chosen periodic orbit) and immediate reaction of the controller (application of the control pulse) in such an event.

In computer experiments, it has been confirmed that all these steps of OGY can be carried out successfully in a great variety of systems, achieving stabilization of even long-period orbits.

There are several problems that arise during the attempt to build an experimental setup. Though variables and parameters can be calculated off-line, one must consider that the signals measured from the system are usually corrupted because of noise and several nonlinear operations associated with the A/D conversion (possibly rounding, truncation, finite

word-length, overflow correction, etc.). Use of corrupted signal values and the introduction of additional errors by computer algorithms and linearization used for the control calculation may result in a general failure of the method. Additionally, there are time delays in the feedback loop (e.g., waiting for the reaction of the computer, interrupts generated when sending and receiving data.)

Effects of Calculation Precision. To test the effects of the precision of calculations in (45) the case of calculating control parameters to stabilize a fixed point in the Lozi map [see (45)] was considered. A partial answer to the question of how the A/D conversion accuracy and the resulting calculations of limited precision affect the possibilities for control has been found. In the tests the quality of computations alone, without looking at other problems like time delays in the control loop, was taken into account.

To compare the results of digital manipulations, first the interesting parameters were computed using analytical formulas. Next the same parameters were calculated using different word-length and different implementations of the arithmetic operations (overflow rules, rounding, or truncation, etc.).

Comparing the results of computations, it was found that an accuracy of two to three decimal digits is possible to achieve and the calculations are precise enough to ensure proper functioning of the OGY algorithm in the case of the Lozi system. To have some safety margin and robustness in the algorithm, the acceptable A/D accuracy cannot be lower than 12 bits and probably it would be best to apply 16-bit conversion. This kind of accuracy is nowadays easily available using general purpose A/D converters even at speeds in the MHz range. Implementing the algorithms, one must consider the cost of implementation—with growing precision and speed requirements, the cost grows exponentially. This issue might be a great limitation when it comes to integrated circuit (IC) implementations.

Approximate Procedures for Finding Periodic Orbits. Another possible source of problems in the control procedure is errors introduced by algorithms for finding periodic orbits (goals of the control). Using experimental data one can only find approximations to unstable periodic orbits (48,49).

In control applications we used the procedure introduced by Lathrop and Kostelich for recovering unstable periodic orbits from an experimental time series. The results obtained using this procedure strongly depend on the choice of accuracy ϵ and the length of the measured time series. Further, they depend on the choice of norm and the number of state variables analyzed. Also, the stopping criterion ($\|x_{m+k} - x_m\| < \epsilon$) in the case of discretely sampled continuous-time systems is not precise enough. This means that one can never be sure of how many orbits have been found or whether all orbits of a given period have been recovered. As this step is typically carried out off-line, it does not significantly affect the whole control procedure. It has been found in experiments that when the tolerances chosen for detection of unstable orbits were too large, the actual trajectory stabilized during control showed greater variations and the control signal had to be applied at every iteration to compensate for inaccuracies. Clearly, making the tolerance large can cause failure of control.

Effects of Time Delays. Several elements in the control loop may introduce time delays that can be detrimental to the functioning of the OGY method (45). Although all calculations may be done off-line, two steps are of paramount importance:

- detection of the moment when the trajectory passes the chosen Poincaré section
- determination of the moment at which the control signal should be applied (close neighborhood of chosen orbit)

When these two steps are carried out by a computer with a data acquisition card, at least a few interrupts (and therefore a time delay) must be generated in order to detect the Poincaré section, to decide it is in the right neighborhood, and to send the correct control signal.

Most experiments with OGY control of electronic circuits have been able to achieve control when the systems were running in the 10 Hz to 100 Hz range. We found out that for higher-frequency systems time delays become a crucial point in the whole procedure. The failure of control was mainly due to the late arrival of the control pulse. The system was being controlled at a wrong point in state space where the formulas used for calculations were probably no longer valid; trajectory was already far away from the section plane when the control pulse arrived.

CONCLUSIONS

The control problems existing in the domain of chaotic systems are neither fully identified nor solved completely. Because of the extreme richness of these phenomena, especially in higher-order systems, every month new papers appear describing new problems and proposing new solutions. Among the many unanswered questions these seem to be the most interesting: How can the methods already developed be used in real applications? What are the limitations of these techniques in terms of convergence, initial conditions, and so on? What are the limitations in terms of system complexity and possibilities of implementation? Are these methods useful in biology or medicine? Can we use the "butterfly effect" to tame and influence large-scale systems?

New application areas have opened up thanks to these new developments in various aspects of controlling chaos. These include neural signal processing (50,51), biology and medicine [Nicolis (52), Garfinkel et al. (53), Schiff et al. (54)], and many others. We can expect in the near future a breakthrough in the treatment of cardiac dysfunction thanks to the new generation of defibrillators and pacemakers functioning on the chaos-control principle. There is great hope also that chaos-control mechanisms will give us insight into one of the greatest mysteries—the workings of the human brain.

There is one more control problem associated in a way with chaos control, although not directly. Sensitive dependence on initial conditions, the key property of chaotic systems, offers yet another fantastic control possibility called "targeting" [Kostelich et al. (55), Shinbrot et al. (56)]. A desired point in the phase space is reached by piecing together in a controlled way fragments of chaotic trajectories. This method has already been applied successfully for directing satellites to desired positions using infinitesimal amounts of fuel [see Farquhar et al. (57)].

Finally, we stress that almost all chaotic systems known to date have strong links with electronic circuits; variables are sensed in an electric or electronic way; identification, modeling, and control are carried out using electric analogs; electronic equipment and electronic computers and usually sensors, transducers, and actuators are also electric by principle of operation. This guarantees an infinite wealth of opportunities for researchers and engineers.

BIBLIOGRAPHY

1. M. J. Ogorzałek and Z. Galias, Characterization of chaos in Chua's oscillator in terms of unstable periodic orbits, *J. Circuits Syst. and Computers,* **3**: 411–429, 1993.
2. M. J. Ogorzałek, Taming chaos: Part II—control, *IEEE Trans. Circuits Syst.,* **CAS-40**: 700–706, 1993.
3. M. J. Ogorzałek, Chaos control: How to avoid chaos or take advantage of it, *J. Franklin Inst.,* **331B** (6): 681–704, 1994.
4. M. J. Ogorzałek, *Chaos and complexity in nonlinear electronic circuits,* Singapore: World Scientific, 1997.
5. T. Kapitaniak, L. Kocarev, and L. O. Chua, Controlling chaos without feedback and control signals, *Int. J. Bifurcation Chaos,* **3**: 459–468, 1993.
6. A. Hübler, Adaptive control of chaotic systems, *Helvetica Physica Acta,* **62**: 343–346, 1989.
7. A. Hübler and E. Lüscher, Resonant stimulation and control of nonlinear oscillators, *Naturwissenschaft,* **76**: 76, 1989.
8. Y. Breiman and I. Goldhirsch, Taming chaotic dynamics with weak periodic perturbation, *Phys. Rev. Letters,* **66**: 2545–2548, 1991.
9. H. Herzel, Stabilization of chaotic orbits by random noise, *ZAMM,* **68**: 1–3, 1988.
10. M. J. Ogorzałek and E. Mosekilde, Noise induced effects in an autonomous chaotic circuit, *Proc. IEEE Int. Symp. Circuits Syst.* **1**: 578–581, 1989.
11. G. Chen and X. Dong, From chaos to order—perspectives and methodologies in controlling chaotic nonlinear dynamical systems, *Int. J. Bifurcation and Chaos,* **3**: 1363–1409, 1993.
12. R. N. Madan (ed.), *Chua's circuit; A paradigm for chaos,* Singapore: World Scientific, 1993.
13. K. Pyragas, Continuous control of chaos by self-controlling feedback, *Physics Letters A,* **A170**: 421–428, 1992.
14. G. Mayer-Kress et al., Musical signals from Chua's circuit, *IEEE Trans. Circ. Systems Part II,* **40**: 688–695, 1993.
15. P. Celka, Control of time-delayed feedback systems with application to optics, *Proc. Workshop on Nonlinear Dynamics of Electron. Syst.,* 1994, pp. 141–146.
16. E. Ott, C. Grebogi, and J. A. Yorke, Controlling chaos, *Phys. Rev. Letters,* **64**: 1196–1199, 1990.
17. E. Ott, C. Grebogi, and J. A. Yorke, Controlling Chaotic Dynamical Systems, in D. K. Campbell (ed.), *Chaos: Soviet-American perspectives on nonlinear science,* New York: American Institute of Physics, 1990, pp. 153–172.
18. W. L. Ditto, M. L. Spano, and J. F. Lindner, Techniques for the control of chaos, *Physica,* **D86**: 198–211, 1995.
19. U. Dressler and G. Nitsche, Controlling chaos using time delay coordinates, *Phys. Rev. Letters,* **68**: 1–4, 1992.
20. A. Dąbrowski, Z. Galias, and M. J. Ogorzałek, On-line identification and control of chaos in a real Chua's circuit, *Kybernetika,* **30**: 425–432, 1994.
21. H. Dedieu and M. J. Ogorzałek, Controlling chaos in Chua's circuit via sampled inputs, *Int. J. Bifurcation and Chaos,* **4**: 447–455, 1994.
22. E. R. Hunt, Stabilizing high-period orbits in a chaotic system: The diode resonator, *Phys. Rev. Letters,* **67**: 1953–1955, 1991.
23. E. R. Hunt, Keeping chaos at bay, *IEEE Spectrum,* **30**: 32–36, 1993.
24. G. E. Johnson, T. E. Tigner, and E. R. Hunt, Controlling chaos in Chua's circuit, *J. Circuits Syst. Comput.,* **3**: 109–117, 1993.
25. E. Corcoran, Kicking chaos out of lasers, *Scientific American,* November, p. 19, 1992.
26. I. Peterson, Ribbon of chaos: Researchers develop a lab technique for snatching order out of chaos, *Science News,* **139**: 60–61, 1991.
27. R. Roy et al., Dynamical control of a chaotic laser: Experimental stabilization of a globally coupled system, *Phys. Rev. Letters,* **68**: 1259–1262, 1990.
28. Z. Galias et al., Electronic chaos controller, *Chaos Solitons and Fractals,* **8** (9): 1471–1484, 1997.
29. Z. Galias et al., A feedback chaos controller: Theory and implementation, in *Proc. 1996 IEEE ISCAS Conf.,* **3**: 120–123, 1996.
30. M. P. Kennedy, Chaos in the Colpitts oscillator, *IEEE Trans. Circuit Syst.,* **CAS-41** (11): 771–774, 1994.
31. M. J. Ogorzałek, Taming Chaos: Part I - Synchronization, *IEEE Trans. Circuits Syst.,* **CAS-40**: 693–699, 1993.
32. L. Kocarev and M. J. Ogorzałek, Mutual synchronization between different chaotic systems, *Proc. NOLTA Conf.,* **3**: 835–840, 1993.
33. K. Kaneko, Clustering, coding, switching, hierarchical ordering and control in a network of chaotic elements, *Physica D,* **41**: 137–172, 1990.
34. Gang Hu and Zhilin Qu, Controlling spatiotemporal chaos in coupled map lattice system, *Phys. Rev. Letters,* **72** (1): 68–71, 1994.
35. Gang Hu, Zhilin Qu, and Kaifen He, Feedback control of chaos in spatio-temporal systems, *Int. J. Bif. Chaos,* **5** (4): 901–936, 1995.
36. A. Perez-Meñuzuri et al., Spatiotemporal structures in discretely-coupled arrays of nonlinear circuits: A review, *Int. J. Bif. Chaos,* **5**: 17–50, 1995.
37. Y. Breiman, J. F. Lindner, and W. L. Ditto, Taming spatio-temporal chaos with disorder, *Nature,* **378**: 465–467, 1995.
38. A. Babloyantz, C. Lourenço, and J. A. Sepulchre, Control of chaos in delay differential equations, in a network of oscillators and in model cortex, *Physica D,* **86**: pp. 274–283, 1995.
39. M. J. Ogorzałek et al., Wave propagation, pattern formation and memory effects in large arrays of interconnected chaotic circuits, *Int. J. Bif. Chaos,* **6** (10): 1859–1871, 1996.
40. V. N. Biktashev and A. V. Holden, Design principles of a low voltage cardiac defibrillator based on the effect of feedback resonant drift, *J. Theor. Biol.,* **169**: 101–112, 1994.
41. K. Aihara, Chaotic neural networks, in H. Kawakami (ed.), *Bifurcation Phenomena in Nonlinear Systems and Theory of Dynamical Systems.* Singapore: World Scientific, 1990.
42. K. Aihara, T. Takabe, and M. Toyoda, Chaotic neural networks, *Physics Letters A,* **144**: 333–340, 1990.
43. M. Adachi, Controlling a simple chaotic neural network using response to perturbation. *Proc. NOLTA'95 Conf.,* 989–992, 1995.
44. S. Mizutani et al., Controlling chaos in chaotic neural networks, *Proc. IEEE ICNN'95 Conf., Perth,* 3038–3043, 1995.
45. M. J. Ogorzałek, Design considerations for Electronic Chaos Controllers, *Chaos, Solitons and Fractals* (in press) 1997.
46. M. J. Ogorzałek, Controlling chaos in electronic circuits, *Phil. Trans. Roy. Soc. London,* **353A** (1701): 127–136, 1995.
47. W. L. Ditto and M. L. Pecora, Mastering chaos, *Scientific American,* 62–68, 1993.
48. D. Auerbach et al., Controlling chaos in high dimensional systems, *Phys. Rev. Letters,* **69** (24): 3479–3481, 1992.
49. I. B. Schwartz and I. Triandaf, Tracking unstable orbits in experiments, *Phys. Rev. A,* **46**: 7439–7444.

50. W. Freeman, Tutorial on neurobiology: From single neurons to brain chaos, *Int. J. Bif. Chaos,* **2** (3): 451–482, 1992.
51. Y. Yao and W. J. Freeman, Model of biological pattern recognition with spatially chaotic dynamics, *Neural Networks* **3**: 153–170, 1990.
52. J. S. Nicolis, Chaotic dynamics in biological information processing: A heuristic outline, in H. Degn, A. V. Holden, and L. F. Olsen (eds.), *Chaos in Biological Systems.* New York: Plenum Press, 1987.
53. A. Garfinkel et al., Controlling cardiac chaos, *Science,* **257**: 1230–1235, 1992.
54. S. J. Schiff et al., Controlling chaos in the brain, *Nature,* **370**: 615–620, 1994.
55. E. J. Kostelich et al., Higher-dimensional targeting, *Physical Review E,* **47**: 305–310, 1993.
56. T. Shinbrot et al., Using sensitive dependence of chaos (the "Butterfly effect") to direct trajectories in an experimental chaotic system, *Phys. Rev. Letters,* **68**: 2863–2866, 1992.
57. R. Farquhar et al., Trajectories and orbital maneuvers for the ISEE-3/ICE comet mission, *J. Astronautical Sci.,* **33**: 235–254, 1985.

MACIEJ OGORZAŁEK
University of Mining and
Metallurgy

CHARACTERIZATION OF AMPLITUDE NOISE. See FREQUENCY STANDARDS, CHARACTERIZATION.

CHARACTERIZATION OF FREQUENCY STABILITY. See FREQUENCY STANDARDS, CHARACTERIZATION.

CHARACTERIZATION OF PHASE NOISE. See FREQUENCY STANDARDS, CHARACTERIZATION.

CHARGE FUNDAMENTAL. See ELECTRONS.

/ /

Chaotic Systems Control
Standard Article

Maciej Ogorzaek[1]
[1]University of Mining and Metallurgy, Kraków, Poland

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2541
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (274K)

Abstract

The sections in this article are

Definition of Chaotic Behavior

What Chaos Control Means

Goals of Control

Suppressing Chaotic Oscillations by Changing System Design

External Perturbation Techniques

Control Engineering Approaches

Stabilizing Unstable Periodic Orbits

Chaos Control by Occasional Proportional Feedback

Improved Electronic Chaos Controller

Chaos-to-Chaos Control

Control of Spatiotemporal Chaotic Systems

Electronic Chaos Controllers

Conclusions

| | |

Copyright © 1999-2008 All Rights Reserved.

/ /

- HOME
- ABOUT US
- CONTACT US
- HELP

Wiley Encyclopedia of Electrical and Electronics Engineering

***Browse* this title**

- *Search* this title

Enter words or phrases Go

- Advanced Product Search
- Acronym Finder

Convolution

Standard Article

Bernd-Peter Paris[1]

[1]George Mason University, Fairfax, VA

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2406

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (293K)

Abstract

The sections in this article are

Notation and Basic Definitions

Linear, Time-Invariant Systems

Fundamental Properties

Numerical Convolution

Fast Algorithms for Convolution

Applications and Extensions

Summary

| | |

Copyright © 1999-2008 All Rights Reserved.

CONVOLUTION

Convolution may be the single most important arithmetic operation in electrical engineering because any linear, time-invariant system generates an output signal by convolving the input with the impulse response of the system. Because of its significance, convolution is now a well-understood operation and is covered in any textbook containing the terms *signals* or *systems* in the title.

This article is intended to review some of the most important aspects of convolution. The fundamental relationship between linear, time-invariant systems alluded to in the first paragraph is reexamined and important properties of convolution, including several important transform properties, are presented and discussed.

Then this article discusses computational aspects. Even though the name *convolution* may be a slight misnomer (it appears to intimidate students because of its similarity to the word *convoluted*), it is a fact that continuous-time convolution often cannot be carried out in closed form. This article discusses in some detail procedures for approximating continuous-time convolution through discrete-time convolution.

Continuing with computational considerations, the article addresses the problem of computationally efficient, fast algorithms for convolution. This has been an active area of research until fairly recently, and the article provides insight into the principal approaches for devising fast algorithms.

The article concludes by examining several areas in which convolution or related operations play a prominent role, including error-correcting coding and statistical correlation. Finally, the article provides a brief introduction to the idea of abstract signal spaces.

NOTATION AND BASIC DEFINITIONS

Convolution is an algebraic operation that requires two input signals and produces a third signal as the result. Convolution is defined for signals from both the continuous-time and the discrete-time domain. Continuous-time signals are simply functions of a free parameter t that takes on a continuum of values. We will denote continuous-time signals by a lowercase letter and indicate the continuous-time parameter in parentheses [e.g., $x(t)$]. Similarly, discrete-time signals are functions of a free parameter n that takes integer values only. We denote discrete-time signals by a lowercase letter followed by the discrete-time parameter enclosed in square brackets (e.g., $x[n]$). We will treat continuous-time and discrete-time convolution in parallel and repeatedly explore connections between the two.

Continuous Time

For continuous-time signals the convolution of two signals $x(t)$ and $y(t)$ is denoted as $z(t) = x(t) * y(t)$ and defined as

$$z(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)\,d\tau \tag{1}$$

where we assume the integral exists for all values of t.

To alleviate common confusion about this definition, several observations can be made. First, the result $z(t)$ is a function of t and, thus, a continuous-time signal. Furthermore, the variable τ is simply an integration variable and, therefore, does not appear in the result. Most important, convolution requires integration of the product of two signals; one of these, $y(t-\tau)$, is time reversed with respect to the integration variable τ and its location depends on the variable t. We illustrate these considerations by means of an example.

Let the signals to be convolved be given by

$$x(t) = \exp\left(-\frac{t}{2}\right)u(t) \tag{2}$$

$$y(t) = \begin{cases} \dfrac{t}{5} & \text{for } 0 \le t \le 5, \\ 0 & \text{else} \end{cases} \tag{3}$$

where $u(t)$ denotes the unit-step function [i.e., $u(t) = 1$ if $t \ge 0$ and $u(t) = 0$ otherwise]. The signals $x(t)$ and $y(t)$ are shown in Fig. 1.

The definition of Eq. (1) prescribes that we must integrate over the product of $x(\tau)$ and $y(t-\tau)$. Figure 2 shows these signals for three different values of t in the left-hand column. Considering these graphs from top to bottom, we see that $y(t-\tau)$ slides from left to right with increasing t. Furthermore, the orientation of $y(t-\tau)$ is flipped relative to the orientation of the signal $y(t)$ in Fig. 1. The signal $x(\tau)$ is repeated for reference.

The right-hand column in Fig. 2 shows the product of the two signals in the respective left-column plot. The result of the convolution is the integral of the product (i.e., the area indicated in the plots in the right column). Note that the area depends on the value of t and, hence, the result of the convolution operation is a function of t.

Once the principles of convolution are understood, it is fairly easy to evaluate Eq. (1) analytically for this example.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

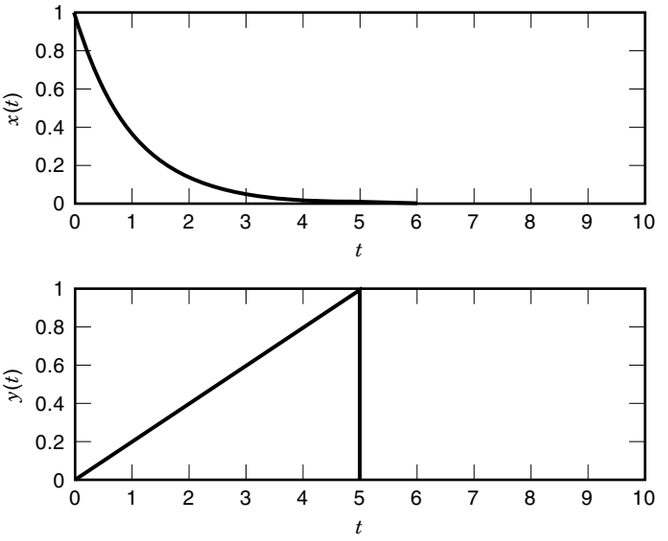


Figure 1. The signals $x(t)$ (top) and $y(t)$ (bottom) used to illustrate convolution.

First, note that $y(t - \tau)$ extends from $t - 5$ to t (i.e., it is zero outside this range). Hence, we should consider three different cases as follows.

1. $t < 0$: In this case, the product of x(τ) and $y(t - \tau)$ is equal to zero and, thus, the result $z(t)$ equals zero for $t \leq 0$. This case is illustrated in the top row of Fig. 2.
2. $0 \leq t < 5$: Here, the nonzero part of $y(t - \tau)$ overlaps partially with the nonzero part of $x(\tau)$. Specifically, the product of $x(\tau)$ and $y(t - \tau)$ is nonzero for τ between zero and t. This is illustrated in the middle row in Fig. 2. Hence, we can write

$$z(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)\,d\tau = \int_{0}^{t} \exp\left(-\frac{\tau}{2}\right)\frac{t-\tau}{5}\,d\tau \tag{4}$$

This integral is easily evaluated by parts and yields

$$z(t) = \frac{2}{5}t - \frac{4}{5}\left(1 - \exp\left(-\frac{t}{2}\right)\right) \tag{5}$$

3. $t \geq 5$: in this case, the nonzero part of $y(t - \tau)$ overlaps completely with the nonzero part of $x(\tau)$. Hence, the product of $x(\tau)$ and $y(t - \tau)$ is nonzero for τ between $t - 5$ and t. The last row in Fig. 2 provides an example for this case. To determine $z(t)$, we can write

$$z(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)\,d\tau = \int_{t-5}^{t} \exp\left(-\frac{\tau}{2}\right)\frac{t-\tau}{5}\,d\tau \tag{6}$$

Thus, the only difference to the previous case is the lower limit of integration. Again, the integral is easily evaluated and yields

$$z(t) = \frac{6}{5}\exp\left(-\frac{t-5}{2}\right) + \frac{4}{5}\exp\left(-\frac{t}{2}\right) \tag{7}$$

The resulting signal $z(t)$ is plotted in Fig. 3.

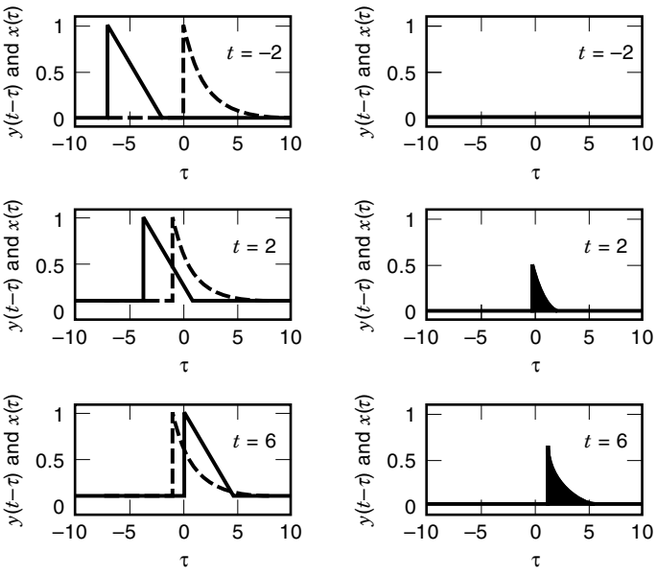


Figure 2. Illustration of convolution operation. The left-hand column shows $x(\tau)$ and $y(t - \tau)$ for three different values of t. The right-hand column indicates the intergral over the product of the two signals in the respective left-hand plots.

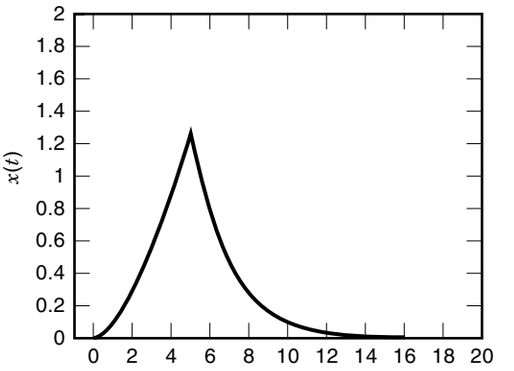


Figure 3. The result $z(t)$ of the convolution. Note that $z(t)$ retains features of both signals. For t between zero and 5, $z(t)$ resembles the ramp signal $y(t)$. After $t = 5$, $z(t)$ is an exponentially decaying signal like $x(t)$.

Discrete Time

For discrete-time signals $x[n]$ and $y[n]$, convolution is denoted by $z[n] = x[n] * y[n]$ and defined as

$$z[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot y[n-k] \tag{8}$$

Notice the similarity between the definitions of Eqs. (1) and (8).

n	0	1	2	3	4	5	6	7	8
$x[n]$	2	4	6	4	2				
$y[n]$	1	−3	3	−1					
$k = 0$: $x[0] \cdot y[n-0]$	2	−6	6	−2					
$k = 1$: $x[1] \cdot y[n-1]$		4	−12	12	−4				
$k = 2$: $x[2] \cdot y[n-2]$			6	−18	18	−6			
$k = 3$: $x[3] \cdot y[n-3]$				2	−6	6	−2		
$k = 4$: $x[4] \cdot y[n-4]$					4	−12	12	−4	
$z[n]$	2	−2	0	−6	12	−12	10	−4	

Figure 4. Convolution of finite length sequences.

A simple algorithm can be used to carry out the computations prescribed by Eq. (8) for finite length signals. Notice that $z[n]$ is computed by summing terms of the form $x[k] \cdot y[n - k]$. We can take advantage of this observation by organizing data in a tableau, as illustrated in Fig. 4. The example in Fig. 4 shows the convolution of $x[n] = \{2, 4, 6, 4, 2\}$ with $y[k] = \{1, -3, 3, -1\}$. We begin by writing out the signal $x[n]$ and $y[n]$. Then we use a process similar to "long multiplication" to form the output by summing shifted rows. The kth shifted row is produced by multiplying the $y[n]$ row by $x[k]$ and shifting the result k positions to the right. The final answer is obtained by summing down the columns. It is easily seen from this procedure that the length of the resulting sequence $z[n]$ must be one less than the sum of the lengths of the inputs $x[n]$ and $y[n]$.

LINEAR, TIME-INVARIANT SYSTEMS

The most frequent use of convolution arises in connection with the large and important class of linear, time-invariant systems. We will see that for any linear, time-invariant system the output signal is related to the input signal through a convolution operation. For simplicity, we will focus on discrete-time systems in this section and comment on the continuous-time case toward the end.

Systems

To facilitate our discussion, let us briefly clarify what is meant by the term *system,* and more specifically *discrete-time system.* As indicated by the block diagram in Fig. 5, a discrete-time system accepts a discrete-time signal $x[n]$ as its input. This input is transformed by the system into the discrete-time output signal $y[n]$. We use the notation

$$x[n] \longmapsto y[n] \tag{9}$$

to symbolize the operation of the system. Linear, time-invariant systems form a subset of all systems. Before proceeding to demonstrate the main point of this section, we pause briefly to define the concepts of linearity and time invariance.

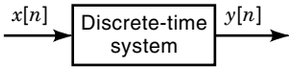


Figure 5. Discrete-time system.

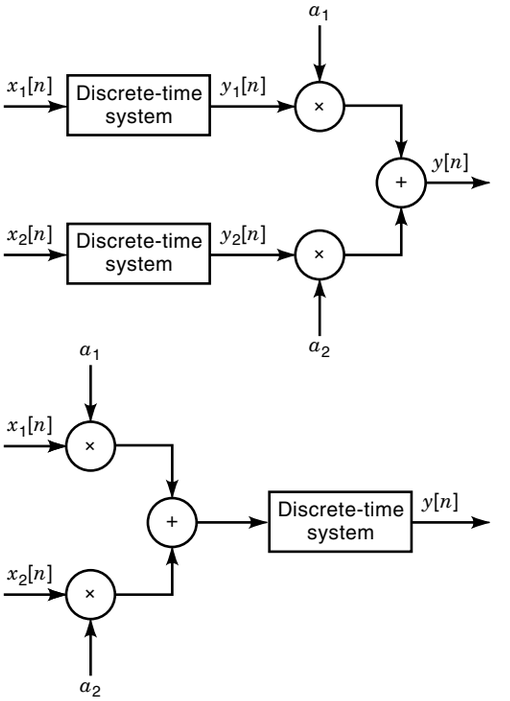


Figure 6. Linearity. For the discrete-time system to be linear, the outputs $y[n]$ of the two blocks must be equal to every choice of constants a_1 and a_2 and for all input signals $x_1[n]$ and $x_1[n]$.

Linearity. Linear system are characterized by the so-called principle of superposition. This principle says that if the input to the system is the sum of two scaled signals, then we can find the output by first computing the outputs due to each of the sequences and then add the two scaled outputs. More formally, linearity is defined as follows. Let $y_1[n]$ and $y_2[n]$ be the outputs of the system due to arbitrary inputs $x_1[n]$ and $x_2[n]$, respectively. Then the system is linear if, for arbitrary constants a_1 and a_2, the output of the system due to input $a_1x_1[n] + a_2x_2[n]$ equals $a_1y_1[n] + a_2y_2[n]$.

This property is illustrated by the block diagrams in Fig. 6. The figure also indicates that linearity implies that the addition and scaling of signals may be interchanged with the operation of the system.

Time Invariance. A system is time invariant if a delay of the input signal results in an equally delayed output signal. More specifically, let $y[n]$ be the output when $x[n]$ is the input. If the input is delayed by n_0 samples and becomes $x[n - n_0]$, then the resulting output must be $y[n - n_0]$ for the system to be time invariant.

Figure 7 illustrates the concept of time invariance. The diagram implies that for time-invariant systems the delay and the operation of the system can be interchanged.

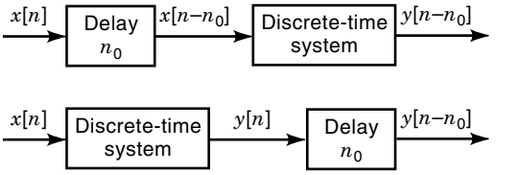


Figure 7. Time invariance. The outputs $y[n - n_0]$ must be equal for all delays n_0 for the system to be time invariant.

Impulse Response. The output of a system in response to an impulse input is called the impulse response. Mathematically, impulses are described by delta functions, and for discrete-time signals the delta function is defined as

$$\delta[n] = \begin{cases} 1 & \text{for } n = 0 \\ 0 & \text{for } n \neq 0 \end{cases} \tag{10}$$

It is customary to denote the impulse response as $h[n]$. Hence, we may write

$$\delta[n] \longmapsto h[n] \tag{11}$$

We have now accumulated enough definitions to proceed and demonstrate that there exists an intimate link between convolution and the operation of linear, time-invariant systems.

Convolution and Linear, Time-Invariant Systems

We will show that the output $y[n]$ of any linear, time-invariant system in response to an input $x[n]$ is given by the convolution of $x[n]$ and the impulse response $h[n]$. This is an amazing result, as it implies that a linear, time-invariant system is completely described by its impulse response $h[n]$. Furthermore, even though linear, time-invariant systems form a very large and rich class of systems with numerous applications wherever signals must be processed, the only operation performed by these systems is convolution.

To begin, recall that the output of a system in response to the input $\delta[n]$ is the impulse response $h[n]$. For time-invariant systems, the response to a delayed impulse $\delta[n - k]$ must be a correspondingly delayed impulse response $h[n - k]$. Furthermore, the relationship $\delta[n - k] \mapsto h[n - k]$ must hold for any (integer) value k if the system is time invariant.

Additionally, if the system is linear, we may scale the input by an arbitrary constant and effect only an equal scale on the output signal. In particular, the following relationships are all true for linear and time-invariant systems:

$$\begin{aligned} &\vdots \\ x[-1]\delta[n+1] &\longmapsto x[-1]h[n+1] \\ x[0]\delta[n] &\longmapsto x[0]h[n] \\ x[1]\delta[n-1] &\longmapsto x[1]h[n-1] \\ &\vdots \\ x[k]\delta[n-k] &\longmapsto x[k]h[n-k] \\ &\vdots \end{aligned} \tag{12}$$

Here $x[n]$ is an arbitrary signal.

Finally, because of linearity, we may sum up all the signals on the right-hand side and be assured that this sum is the output for an input signal that is equal to the sum of the signals on the left-hand side. This means that

$$\sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \longmapsto \sum_{k=-\infty}^{\infty} x[k]h[n-k] = x[n] * h[n] \tag{13}$$

Thus, the output signal is equal to the convolution of $x[n]$ and $h[n]$.

The left-hand side requires a little more thought. For a given k, $x[k]\ \delta[n - k]$ is a signal with a single nonzero sample at $n = k$. Hence, the sum of all such signals is itself a signal and the samples are equal to $x[n]$. Thus, we conclude that

$$x[n] = x[n] * \delta[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \tag{14}$$

We will revisit this fact later in this article.

The preceding discussion can be summarized by the relationship

$$x[n] \longmapsto x[n] * h[n] \tag{15}$$

In words, the output of a linear, time-invariant system with impulse response $h[n]$ and input $x[n]$ is given by $x[n] * h[n]$. Recall that we have only invoked linearity and time invariance to derive this relationship. Hence, this fundamental result is true for any linear, time-invariant system.

Continuous-Time Systems

The entire preceding discussion is valid for continuous-time systems, too. In particular, every linear, time-invariant system is completely characterized by its impulse response $h(t)$, and the output of the system in response to an input $x(t)$ is given by $y(t) = x(t) * h(t)$. A proof of this relationship is a little more cumbersome than in the discrete time case, mainly because the continuous-time impulse $\delta(t)$ is more cumbersome to manipulate than its discrete-time counterpart. We will discuss $\delta(t)$ later.

FUNDAMENTAL PROPERTIES

The convolution operation possesses several useful properties. In many cases these properties can be exploited to simplify the manipulation of expressions involving convolution. We will rely on many of the properties presented here in the subsequent exposition.

Symmetry

The order in which convolution is performed does not affect the final result [i.e., $x(t) * y(t)$ equals $y(t) * x(t)$]. This fact is easily shown by substituting σ for $t - \tau$ in Eq. (1). Then we obtain

$$z(t) = \int_{-\infty}^{\infty} x(t-\sigma)y(\sigma)\,d\sigma \tag{16}$$

which obviously equals $y(t) * x(t)$. The corresponding relationship for discrete-time signals can be established in the same manner.

Convolving with Delta Functions

The delta function is of fundamental importance in the analysis of signals and systems. The continuous-time delta function is defined implicitly through the relationship

$$\int_{-\infty}^{\infty} x(t)\delta(t-T)\,dt = x(T) \tag{17}$$

where the $x(t)$ is an arbitrary signal that is continuous at $t = T$. From this definition it follows immediately that

$$x(t) * \delta(t - t_0) = \int_{-\infty}^{\infty} x(\tau)\delta(t - t_0 - \tau)\, d\tau = x(t - t_0) \tag{18}$$

Hence, convolving a signal with a time-delayed delta function is equivalent to delaying the signal. The induced delay of the signal is equal to the delay t_0 of the delta function.

Analogous to the continuous-time case, when an arbitrary signal $x[n]$ is convolved with a delayed delta function $\delta[n - n_0]$, the result is a delayed signal $x[n - n_0]$. We have already seen this fact in Eq. (14) for the case $n_0 = 0$.

Convolving with the Unit-Step Function

An ideal integrator computes the "running" integral over an input signal $x(t)$. That is, the output $y(t)$ of the ideal integrator is given by

$$y(t) = \int_{-\infty}^{t} x(\tau)\, d\tau \tag{19}$$

With the unit-step function $u(t)$, we may rewrite this equality as

$$y(t) = x(t) * u(t) = \int_{-\infty}^{\infty} x(\tau)u(t - \tau)\, d\tau \tag{20}$$

The equality between the two expressions follows from the fact that $u(t - \tau)$ equals one for τ between $-\infty$ and t and $u(t - \tau)$ is zero for $\tau > t$.

The corresponding relationship for discrete-time signals is

$$y[n] = \sum_{k=-\infty}^{n} x[k] = \sum_{k=-\infty}^{\infty} x[k] \cdot u[n - k] = x[n] * u[n] \tag{21}$$

where $u[n]$ is equal to one for $n \geq 0$ and zero otherwise.

Transform Relationships

For both continuous- and discrete-time signals there exist transforms for computing the frequency domain description of signals. While these transforms may be of independent interest in the analysis of signals, they also exhibit a very important relationship to convolution.

Laplace and Fourier Transform. The Laplace transform of a signal $x(t)$ is denoted by $\mathcal{L}\{x(t)\}$ or $X(s)$ and is defined as

$$X(s) = \mathcal{L}\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-st}\, dt \tag{22}$$

where s is complex valued and can be written as $s = \sigma + j\omega$. We will assume throughout this section that signals are such that their region of convergence for the Laplace transform includes the imaginary axis (i.e., the preceding integral converges for $\Re\{s\} = \sigma = 0$). Hence, we obtain the Fourier transform, $\mathcal{F}\{x(t)\}$ or $X(f)$, of $x(t)$ by evaluating the Laplace transform for $s = j2\pi f$.

The Laplace transform can be interpreted as the complex-valued magnitude of the response by a linear, time-invariant system with impulse response $h(t)$ to an input $x(t) = \exp(st)$. Then the output is given by

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} h(\tau)x(t - \tau)\, d\tau \\ &= \int_{-\infty}^{\infty} h(\tau)e^{s(t-\tau)}\, d\tau \\ &= e^{st}\int_{-\infty}^{\infty} h(\tau)e^{-s\tau}\, d\tau \\ &= e^{st}H(s) \end{aligned} \tag{23}$$

where $H(s)$ denotes the Laplace transform of $h(t)$. $H(s)$ is commonly called the transfer function of the system. Notice, in particular, that the output $y(t)$ is an exponential signal with the same exponent as the input; the only difference between input and output is the complex-valued multiplicative constant $H(s)$. This observation is often summarized by the statement that (complex) exponential signals are eigenfunctions of linear, time-invariant systems.

The Laplace transform of the convolution of signals $x(t)$ and $y(t)$ can be written as

$$\begin{aligned} \mathcal{L}\{x(t) * y(t)\} &= \mathcal{L}\left\{\int_{-\infty}^{\infty} x(\tau)y(t - \tau)\, d\tau\right\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(\tau)y(t - \tau)e^{-st}\, d\tau\, dt \end{aligned} \tag{24}$$

Substituting $\sigma = t - \tau$ and $d\sigma = d\tau$ yields

$$\begin{aligned} \mathcal{L}\{x(t) * y(t)\} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(\tau)y(\sigma)e^{-s(\tau+\sigma)}\, d\tau\, d\sigma \\ &= \int_{-\infty}^{\infty} x(\tau)e^{-s\tau}\, d\tau \cdot \int_{-\infty}^{\infty} y(\sigma)e^{-s\sigma}\, d\sigma \\ &= X(s) \cdot Y(s) \end{aligned} \tag{25}$$

Hence, we have the very important relationship that the Laplace transform of the convolution of two signals, $x(t) * y(t)$, is the product of the respective Laplace transforms, $X(s) \cdot Y(s)$. Clearly, this property also holds for Fourier transforms. This property may be used to simplify the computation of the convolution of two signals. One would first compute the Laplace (or Fourier) transform of the signals to be convolved, then multiply the two transforms, and finally compute the inverse transform of the product to obtain the final result. This procedure is often simpler than direct evaluation of the convolution integral of Eq. (1) when the signals to be convolved have simple transforms (e.g., when the signals are exponentials, including complex exponentials and sinusoids).

Finally, let $x(t)$ be a periodic signal of period T. Then $x(t)$ can be represented by a Fourier series

$$x(t) = \sum_{k=-\infty}^{\infty} x_k \exp(j2\pi kt/T) \tag{26}$$

where the Fourier series coefficients x_k are given by

$$x_k = \frac{1}{T}\int_0^T x(t)\exp(-j2\pi kt/T)\, dt \tag{27}$$

A periodic signal is said to have a discrete spectrum.

If $x(t)$ is convolved with an aperiodic signal $y(t)$, then it is easily shown that the signal $z(t) = x(t) * y(t)$ is periodic and has a Fourier series representation

$$z(t) = \sum_{k=-\infty}^{\infty} z_k \exp(j2\pi kt/T) \tag{28}$$

with Fourier series coefficients z_k equal to the product $x_k \cdot Y(k/T)$, where $Y(f)$ is the Fourier transform of of $y(t)$.

When two periodic signals are convolved, the convolution integral generally does not converge unless the spectra of the two signals do not overlap, in which case the convolution equals zero.

***z*-Transform and Discrete-Time Fourier Transform.** For discrete-time signals, the z-transform plays a role equivalent to the Laplace transform for continuous-time signals. The z-transform $\mathcal{Z}\{x[n]\}$ or $X(z)$ of a discrete-time signal $x[n]$ is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \tag{29}$$

The variable z is complex valued, $z = A \cdot e^{j\omega}$. Analogous to our assumption for the Laplace transform, we will assume throughout that signals are such that their region of convergence includes the unit circle (i.e., the preceding sum converges for $|z| = A = 1$). Then the (discrete-time) Fourier transform $X(f)$ can be found by evaluating the z-transform for $z = \exp(j2\pi f)$. Notice that the discrete-time Fourier transform is periodic in f (with period 1); the continuous-time Fourier transform, in contrast, is not periodic.

Additionally, just as complex exponential signals are eigenfunctions of continuous-time, linear, time-invariant systems, signals of the form $x[n] = z^n$ are eigenfunctions of discrete-time, linear, time-invariant systems. Hence, if $x[n] = z^n$ is the input, then $y[n] = H(z)z^n$ is the output from a linear, time-invariant system with impulse response $h[n]$ and corresponding z-transform $H(z)$.

The z-transform of the convolution of sequences $x[n]$ and $y[n]$ is given by

$$\begin{aligned}\mathcal{Z}\{x[n] * y[n]\} &= \mathcal{Z}\left\{\sum_{k=-\infty}^{\infty} x[k] * y[n-k]\right\} \\ &= \sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} x[k]y[n-k]z^{-n}\end{aligned} \tag{30}$$

By substituting $l = n - k$ and thus $n = l + k$, we obtain

$$\begin{aligned}\mathcal{Z}\{x[n] * y[n]\} &= \sum_{l=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} x[k]y[l]z^{-l-k} \\ &= \sum_{k=-\infty}^{\infty} x[k]z^{-k} \cdot \sum_{l=-\infty}^{\infty} y[k]z^{-l} \\ &= X(z) \cdot Y(z)\end{aligned} \tag{31}$$

Therefore, the z-transform of the convolution of two signals $x[n]$ and $y[n]$ equals the product of the z-transforms $X(z)$ and $Y(z)$ of the signals. Again, the same property also holds for Fourier transforms.

The discrete-time equivalent of the Fourier series is the discrete Fourier transform (DFT). Like the Fourier series, the DFT provides a signal representation using discrete, harmonically related frequencies. Both the Fourier series and the DFT representations result in periodic time functions or signals. For a discrete-time signal of length (or period) N samples, the coefficients of the DFT are given by

$$X_k = \sum_{n=0}^{N-1} x[n]\exp(-j2\pi kn/N) \tag{32}$$

The signal $x[n]$ can be represented as

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X_k \exp(j2\pi kn/N) \tag{33}$$

When a periodic, discrete-time signal $x[n]$ with period N and with DFT coefficients X_k is convolved with a nonperiodic signal $y[n]$, the result is a periodic signal $z[n]$ of period N. Furthermore, the DFT coefficients of the result $z[n]$ are given by $X_k \cdot Y(k/N)$, where $Y(f)$ is the Fourier transform of $y[n]$.

Circular Convolution. An interesting problem arises when we ask ourselves which signal $z[n]$ has DFT coefficients $Z_k = X_k \cdot Y_k$, for $k = 0, 1, \ldots, N-1$. First, because all three signals have DFTs of length N, they are implicitly assumed to be periodic with period N. Further, $z[n]$ can be written as

$$z[n] = \frac{1}{N}\sum_{k=0}^{N-1} Z_k \exp(j2\pi kn/N) = \frac{1}{N}\sum_{k=0}^{N-1} X_k \cdot Y_k \exp(j2\pi kn/N) \tag{34}$$

We can replace X_k using the definition for the DFT and obtain

$$z[n] = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} x[l]\exp(-j2\pi kl/N) \cdot Y_k \exp(j2\pi kn/N) \tag{35}$$

Reversing the order of summation, $z[n]$ can be expressed as

$$z[n] = \sum_{l=0}^{N-1} x[l]\frac{1}{N}\sum_{k=0}^{N-1} Y_k \exp(j2\pi k(n-l)/N) \tag{36}$$

The second summation is easily recognized to be equal to $y[\langle n-l\rangle]$, where $\langle n-l\rangle$ denotes the residue of $n-l$ modulo N (i.e., the remainder of $n-l$ after division by N). The modulus of $n-l$ arises because of the periodicity of the complex exponential, specifically because $\exp(j2\pi k(n-l)/N)$ and $\exp(j2\pi k(\langle n-l\rangle)/N)$ are equal. Hence, $z[n]$ can be written as

$$z[n] = \sum_{k=0}^{N-1} x[k] \cdot y[\langle n-k\rangle] = x[n] \circledast y[n] \tag{37}$$

This operation is similar to convolution as defined in Eq. (8) and referred to as circular convolution. The subtle, yet important, difference from regular, or linear, convolution is the occurrence of the modulus in the index of the signal $y[n]$. An immediate consequence of this difference is the fact that the circular convolution of two length N signals is itself of length N. The linear convolution of two signals of length N, however, yields a signal of length $2N-1$.

Incidentally, a similar relationship exists in continuous time. Let $x(t)$ and $y(t)$ be periodic signals with period T and Fourier series coefficients X_k and Y_k, respectively. Then the signal

$$z(t) = \frac{1}{T}\int_0^T x(\tau)y(\langle t-\tau\rangle)\,d\tau \tag{38}$$

is periodic with period T and has Fourier series coefficients $Z_k = X_k \cdot Y_k$. This property can be demonstrated in a manner analogous to that used for discrete-time signals.

We will investigate the relationship between linear and circular convolution later. We will demonstrate that circular convolution plays a crucial in the design of computationally efficient convolution algorithms.

NUMERICAL CONVOLUTION

The continuous-time convolution integral is often not computable in closed form. Hence, numerical evaluation of the continuous-time convolution integral is of significant interest. When we are exploring means to compute the integral in Eq. (1) numerically, we will discover that the discrete-time convolution of sampled signals plays a key role. Furthermore, by employing ideal sampling arguments, we develop an understanding for the accuracy of numerical approximations to the convolution integral.

Riemann Approximation

Let us begin by considering a straightforward approximation to continuous-time convolution based on the Riemann approximation to the integral. First, we approximate $z(t)$ by a stair-step function such that

$$z(t) \approx z(nT) \quad \text{for } nT \le \tau < (n+1)T \tag{39}$$

where T is a positive constant. Consequently, the convolution integral needs to be evaluated only at discrete times $t = nT$, and for these times we have

$$z(nT) = \int_{-\infty}^{\infty} x(\tau)y(nT-\tau)\,d\tau \tag{40}$$

Next, we use the Riemann approximation to an integral as follows. The range of integration is broken up into adjacent, non-overlapping intervals of width T. On each interval, we approximate $x(\tau)$ and $y(nT-\tau)$ by

$$\begin{aligned} x(\tau) &\approx x(kT) \text{ for } kT \le \tau < (k+1)T \\ y(nT-\tau) &\approx y((n-k)T) \text{ for } iT \le \tau < (k+1)T \end{aligned} \tag{41}$$

If T is sufficiently small, this approximation will be very accurate. In the limit as T approaches zero, the exact solution $z(nT)$ is obtained. We will discuss the choice of T in more detail later.

The Riemann approximation to the convolution integral is

$$z(nT) \approx \sum_{k=-\infty}^{\infty}\int_{kT}^{(k+1)T} x(kT)y((n-k)T)\,d\tau \tag{42}$$

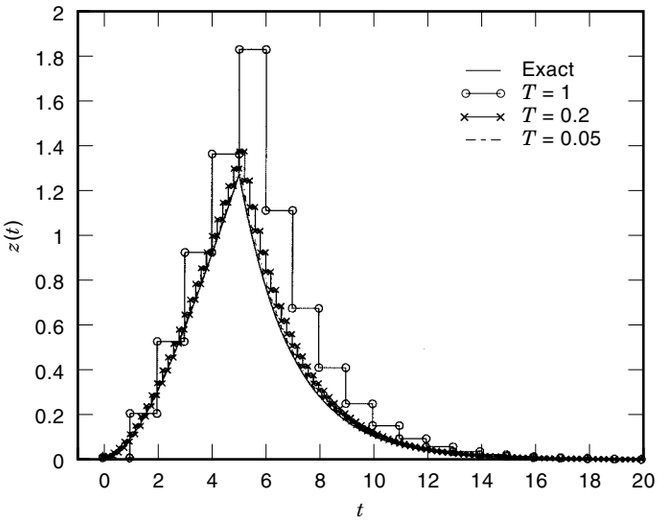


Figure 8. Numerical convolution.

Reversing the order of integration and summation and evaluating the (trivial) integral, we obtain

$$z(nT) \approx T \cdot \sum_{k=-\infty}^{\infty} x(kT)y((n-k)T) \tag{43}$$

Hence, apart from the constant T, this approximation is equal to the discrete-time convolution of sampled signals $x(t)$ and $y(t)$.

To illustrate, let us consider the two signals from the example given in the first section of this article. Figure 8 shows the exact result of the convolution together with approximations obtained by using $T = 1$, $T = 0.2$, and $T = 0.05$. Clearly, the accuracy of the approximation improves significantly with decreasing T. For $T = 0.05$, there is virtually no difference between the exact and the numerical solution.

How to select T remains an open question. Our intuition tells us that T must be small relative to the rate of change of the signals to be convolved. Then the error induced by approximating $x(\tau)$ and $y(nT-\tau)$ by the value of a nearby sample will be small. These notions can be made more precise by considering a system with ideal samplers.

Numerical Convolution via Ideal Sampling

Consider the system in Fig. 9. The signals $x(t)$ and $y(t)$ are sampled before they are convolved. We will see that the result of this convolution depends directly on the discrete-time convolution of the samples $x(nT)$ and $y(nT)$. Finally, the signal $z_p(t)$ is filtered to yield the signal $\hat{z}(t)$. The objective of this analysis is to derive conditions on the sampling rate T and the filter $h(t)$ such that $\hat{z}(t)$ is equal to $z(t)$.

A System with Ideal Samplers. The input signals $x(t)$ and $x(t)$ are first sampled using ideal samplers. Thus, the signals $x_p(t)$ and $y_p(t)$ are given by

$$\begin{aligned} x_p(t) &= \sum_{n=-\infty}^{\infty} x(nT)\delta(t-nT) \text{ and} \\ y_p(t) &= \sum_{n=-\infty}^{\infty} y(nT)\delta(t-nT) \end{aligned} \tag{44}$$

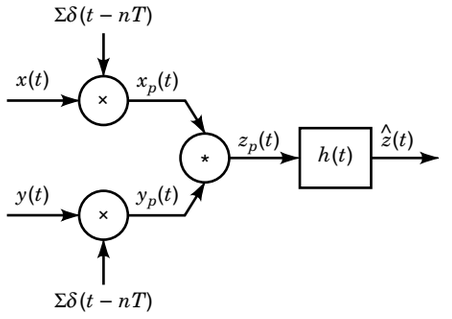


Figure 9. Convolution of ideally sampled signals. The input signals $x(t)$ and $y(t)$ are first sampled at rate $1/T$ and then convolved. The result $z_p(t)$ is then filtered [i.e., convolved with $h(t)$] to produce the approximation $\hat{z}(t)$ to $z(t)$.

Then $x_p(t)$ and $y_p(t)$ are convolved to produce the signal $z_p(t)$, which can be expressed as

$$\begin{aligned} z_p(t) &= x_p(t) * y_p(t) = \int_{-\infty}^{\infty} x_p(\tau) y_p(t-\tau)\, d\tau \\ &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(nT)\delta(\tau - nT) \sum_{k=-\infty}^{\infty} y(kT)\delta(t-\tau-kT)\, d\tau \\ &= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(nT)y(kT) \int_{-\infty}^{\infty} \delta(\tau - nT)\delta(t-\tau-kT)\, d\tau \end{aligned} \tag{45}$$

Based on our considerations regarding the delta function, we recognize that the integral in the last equation is given by

$$\int_{-\infty}^{\infty} \delta(\tau - nT)\delta(t-\tau-kT)\, d\tau = \delta(t-(n+k)T) \tag{46}$$

Substituting this result back into our expression for $z_p(t)$, we obtain

$$z_p(t) = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x(nT)y(kT)\delta(t-(n+k)T) \tag{47}$$

When we further substitute $l = n + k$, $z_p(t)$ becomes

$$z_p(t) = \sum_{l=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x((k-l)T)y(kT) \right) \delta(t-lT) \tag{48}$$

The term in parentheses is simply the discrete-time convolution of the samples $x(nT)$ and $y(nT)$. Hence, $z_p(t)$ is equal to

$$z_p(t) = \sum_{n=-\infty}^{\infty} (x(nT) * y(nT))\delta(t-nT) \tag{49}$$

In other words, $z_p(t)$ is itself an ideally sampled signal with samples given by $x(nT) * y(nT)$. It is important to realize, however, that in general the discrete time signal $x(nT) * y(nT)$ is not equal to the signal $z(nT)$ obtained by sampling $z(t) = x(t) * y(t)$ unless the sampling period T is chosen properly.

Selection of Sampling Rate *T*. To understand the impact of T, it is useful to consider the frequency domain representation of our signals. It is well known that the Fourier transform of an ideally sampled signal is obtained by periodic repetition and scaling of the Fourier transform of the original, nonsampled signal. Specifically, the Fourier transforms $X_p(f)$ and $Y_p(f)$ of the signals $x_p(t)$ and $y_p(t)$ are given by

$$X_p(f) = \frac{1}{T} \sum_{m=-\infty}^{\infty} X\left(f - \frac{m}{T}\right) \tag{50}$$

$$Y_p(f) = \frac{1}{T} \sum_{m=-\infty}^{\infty} Y\left(f - \frac{m}{T}\right) \tag{51}$$

where $X(f)$ and $Y(f)$ denote the Fourier transforms of $x(t)$ and $y(t)$, respectively. Since the Fourier transform of the convolution of $x_p(t)$ and $y_p(t)$ equals the product of $X_p(f)$ and $Y_p(f)$, it follows that the Fourier transform $Z_p(f)$ of $z_p(t)$ is

$$Z_p(f) = X_p(f) \cdot Y_p(f) = \frac{1}{T^2} \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} X\left(f - \frac{m}{T}\right) Y\left(f - \frac{k}{T}\right) \tag{52}$$

Recall that our objective is to obtain $\hat{z}(t)$ approximately equal to $z(t) = x(t) * y(t)$. On the other hand, we know that the Fourier transform $Z(f)$ of $z(t)$ equals $X(f) \cdot Y(f)$, and hence, we must seek to have $\hat{Z}(f)$ approximately equal to $X(f) \cdot Y(f)$. The simplest way to achieve this objective is to choose T small enough that $X(f - m/T)Y(f - k/T) = 0$ whenever $m \neq k$. In other words, T must be small enough that each replica $X(f - m/T)$ overlaps with exactly one replica $Y(f - k/T)$. Under this condition, the expression for $Z_p(f)$ simplifies to

$$Z_p(f) = \frac{1}{T^2} \sum_{m=-\infty}^{\infty} X\left(f - \frac{m}{T}\right) Y\left(f - \frac{m}{T}\right) \tag{53}$$

Notice that this is the Fourier transform of an ideally sampled signal with original spectrum $(1/T)\ X(f)Y(f)$. Expressed in the time domain, if T is chosen to meet the preceding condition, $z_p(t)$ is the ideally sampled version of $z(t) = x(t) * y(t)$.

To summarize these observations, when T is sufficiently small that $X(f - m/T)Y(f - k/T) = 0$ for all $m \neq k$, then

$$\begin{aligned} z_p(t) &= \sum_{n=-\infty}^{\infty} z(nT)\delta(t-nT) \\ &= \sum_{n=-\infty}^{\infty} (x(t) * y(t))|_{t=nT}\delta(t-nT) \\ &= \sum_{n=-\infty}^{\infty} (x(nT) * y(nT))\delta(t-nT) \end{aligned} \tag{54}$$

The convolution on the second line is in continuous time, while the one on the last line is in discrete time.

Most important, we may conclude that if T is chosen properly then the samples $z(nT)$ of $z(t) = x(t) * y(t)$ are equal to $x(nT) * y(nT)$ [i.e., the discrete-time convolution of samples $x(nT)$ and $y(nT)$]. In other words, the order of convolution and sampling may be interchanged provided that the sampling period is sufficiently small.

How do we select the sampling period T to be sufficiently small? Assume that both $x(t)$ and $y(t)$ are ideally band limited to f_x and f_y, respectively. Then $X(f) = 0$ for $|f| > f_x$ and $Y(f) = 0$ for $|f| > f_y$. The first replica of $Y(f)$ [i.e., $Y(f - 1/T)$] extends from $1/T - f_y$ to $1/T + f_y$. For this replica not to overlap with the zeroth replica of $X(f)$ [i.e., $X(f)$ itself], T must be such that

$$\frac{1}{T} - f_y > f_x \tag{55}$$

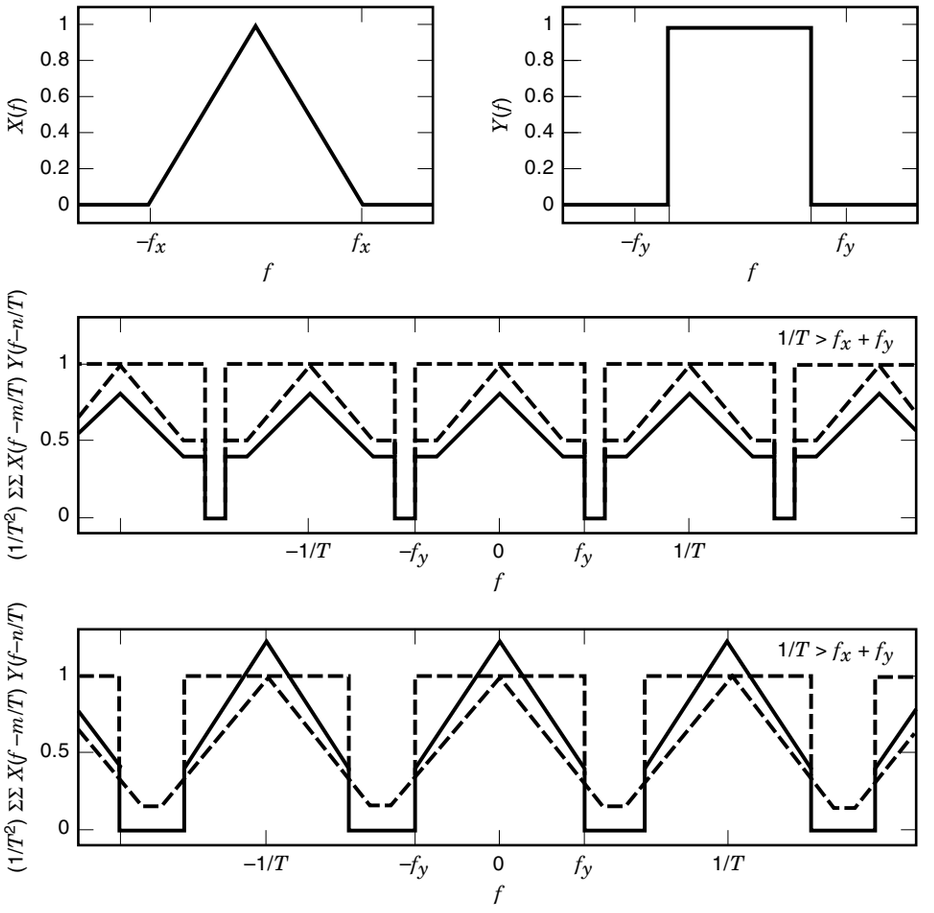


Figure 10. The influence of the sampling rate T on numerical convolution. The spectra of the signals $x(t)$ and $y(t)$ to be convolved are shown on the top row. The spectra in the second and third rows are the result of first sampling and then convolving $x(t)$ and $y(t)$. On the second row, the sampling rate is insufficient and the resulting spectrum is not equal to the spectrum that results from ideally sampling $z(t) = x(t) * Y(t)$. On the bottom row, the sampling rate is sufficient. This is evident because the product of $X(f)$ and $Y(f)$ is visible between $-f_y$ and f_y.

Equivalently, T must satisfy

$$T < \frac{1}{f_x + f_y} \tag{56}$$

Figure 10 illustrates these considerations. The first row shows the spectra $X(f)$ and $Y(f)$ of two strictly band-limited signals. The second and third rows contain plots that show the spectra resulting from first sampling signals $x(t)$ and $y(t)$ and then convolving the two sampled signals. An expression for the resulting expression is provided by Eq. (53). Both spectra are periodic with period $1/T$ and are thus spectra of ideally sampled signals.

However, the spectrum shown on the second row results from a violation of the condition of Eq. (56) on the sampling rate, while for the bottom plot this condition holds. Notice in particular that the segment between $-f_y$ and f_y in the bottom plot is exactly equal to $X(f) \cdot Y(f)$, except for a scale factor. No such segment exists in the middle plot. Hence, the spectrum shown in the bottom plot corresponds to an ideally sampled signal with samples $z(nT)$; the middle plot does not.

Finally, even for the bottom plot, the sampling rate violates the Nyquist criterion ($T < 1/2f_x$) for the signal $x(t)$. This is evident, for example, in the region between f_y and $1/T - f_y$, where aliasing is clearly evident.

Interpolation. We have demonstrated that the sampling rate T should be selected such that $1/T$ exceeds the sum of the bandwidths of the signals to be convolved. Let us turn our attention to the choice of the filter labeled $h(t)$ in Fig. 9. The principal function of this filter is to interpolate between the sample values. It produces the signal $\hat{z}(t)$ by convolving $z_p(t)$ and $h(t)$. Since $\delta(t - lT) * h(t)$ equals $h(t - lT)$, we have immediately

$$\hat{z}(t) = \sum_{n=-\infty}^{\infty} (x(nT) * y(nT))h(t - nT) \tag{57}$$

In particular, for the choice

$$h(t) = \begin{cases} T & 0 \le t < T \\ 0 & \text{else} \end{cases} \tag{58}$$

the same stair-step approximation as in the previous section is obtained.

The function of the interpolation filter is easily expressed in the frequency domain. Because of the transform properties discussed previously, the Fourier transform $\hat{Z}(f)$ of $\hat{z}(t)$ is given by

$$\hat{Z}(f) = Z_p(f)H(f) \tag{59}$$

Assuming that our condition on the sapling rate is met, $\hat{Z}(f)$ equals

$$\hat{Z}(f) = \frac{1}{T^2} H(f) \sum_{m=-\infty}^{\infty} X\left(f - \frac{m}{T}\right) Y\left(f - \frac{m}{T}\right) \tag{60}$$

This equation demonstrates that for $\hat{Z}(f)$ to be similar to $Z(f)$, the interpolation filter must reject all replicas $X(f - m/T)Y(f - m/T)$ for $m \neq 0$.

Furthermore, it should introduce the appropriate gain and no distortion in the passband such that $(1/T^2)\ H(f)X(f)Y(f)$ equals $X(f)Y(f)$. Thus, the ideal choice for $H(f)$ is an ideal lowpass filter. However, the ideal lowpass filter has an infinite impulse response and is therefore not practical.

Frequently used interpolation filters in practice include the simple "hold filter" with $h(t)$, given in Eq. (58), or a linear interpolator, which can be realized by using a filter with impulse response

$$h(t) = \begin{cases} T \cdot (1 - t) & 0 \leq t < T \\ T \cdot (1 + t) & -T \leq t \leq 0 \\ 0 & \text{else} \end{cases} \tag{61}$$

In particular, when T is much smaller than specified by the preceding condition, these simpler interpolators provide excellent results.

Our discussion of numerical convolution can be summarized as follows: Continuous-time convolution can be approximated with arbitrary accuracy through discrete-time convolution of sampled versions of the signals to be convolved as long as the sampling rate is sufficiently large. Specifically, the sampling period T must be chosen to exceed the sum of the bandwidths of the signals to be convolved. We have shown that under this condition, the discrete-time convolution produces a sequence of samples that is equal to samples of the original continuous-time convolution. Intermediate values may be produced via a suitable interpolation filter.

These considerations emphasize the practical importance of computationally efficient algorithms for discrete-time convolution. In the next section, we discuss convolution algorithms that rely heavily on ideas discussed in the context of transforms.

FAST ALGORITHMS FOR CONVOLUTION

Filtering signals with linear, time-invariant systems is probably the most common form of signal processing. Hence, there is enormous interest in algorithms for computationally efficient (discrete-time) convolution. We will see that such algorithms take advantage of the transform relationships that were discussed previously. In particular, the development of fast algorithms for computing the discrete Fourier transform in the late 1960s has been seminal for the field of digital signal processing.

Linear Convolution via Circular Convolution

The operation of linear, time-invariant filters is characterized by linear convolution. However, computationally attractive transform relationships exist for circular convolution. Previously, we showed that the DFT of two circularly convolved signals equals the product of the signals' DFTs. Furthermore, fast algorithms exist to compute the DFT of a signal. These algorithms are commonly called fast Fourier transforms (FFT).

Thus, the principal idea for a fast circular convolution algorithm is to compute the DFT of the signals to be convolved, to multiply the two DFTs, and finally to compute the inverse DFT of this product. All three DFTs can be computed efficiently using a suitable FFT algorithm.

We seek to take advantage of this approach for linear convolution. Toward this objective, let us take a closer look at the differences and similarities between linear and circular convolution.

Convolution via Matrix Multiplication. Both linear and circular convolution can be accomplished via matrix multiplication. This fact is of independent interest in many signal processing applications but will be used here to highlight the relationship between linear and circular convolution.

To fix ideas, consider the convolution of signals $x[n]$ and $y[n]$. Assume for the moment that both of these signals are of length N. The result of the linear convolution of $x[n]$ and $y[n]$ will be denoted $z_l[n]$ and the result of the circular convolution will be denoted $z_c[n]$. Recall that the length of $z_l[n]$ is $2N - 1$, while the length of $z_c[n]$ is N.

Both convolution operations can be expressed as the multiplication of a suitably chosen matrix and vector. Linear convolution can be written as

$$z_l[n] = x[n] * y[n] = X_l \cdot \boldsymbol{y} \tag{62}$$

where X_l is a $(2N - 1) \times N$ matrix and $\boldsymbol{y}$ is the length N vector with elements $y[n]$. The matrix X_l is constructed with columns equal to shifted and zero-padded replicas of $x[n]$, specifically

$$X_l = \begin{pmatrix} x[0] & 0 & 0 & \dots & 0 \\ x[1] & x[0] & 0 & \dots & 0 \\ x[2] & x[1] & x[0] & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & x[N-1] & x[N-2] \\ 0 & \dots & 0 & 0 & x[N-1] \end{pmatrix} \tag{63}$$

The equivalence between convolution and multiplication of X_l and $\boldsymbol{y}$ is easily verified. When the length of $y[n]$ is equal to L, X_l is an $(N + L - 1) \times L$ matrix constructed as previously and $\boldsymbol{y}$ is a length L vector.

Circular convolution can be written as

$$z_c[n] = x[n] * y[n] = X_c \cdot \boldsymbol{y} \tag{64}$$

where X_c is a $N \times N$ matrix and $\boldsymbol{y}$ is as before. In contrast to X_l, the construction of X_c does not involve zero padding. Instead, columns (and rows) are constructed from circular shifts of $x[n]$, specifically

$$X_c = \begin{pmatrix} x[0] & x[N-1] & x[N-2] & \dots & x[1] \\ x[1] & x[0] & x[N-1] & \dots & x[2] \\ x[2] & x[1] & x[0] & \dots & x[3] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x[N-1] & \dots & x[2] & x[1] & x[0] \end{pmatrix} \tag{65}$$

This form of matrix is called a circulant matrix, a special form of Toeplitz matrix (1).

Comparison of the two matrices shows that we can transform linear convolution into an equivalent circular convolution. For that purpose, we must first pad both $x[n]$ and $y[n]$ with zeros to make them length $2N - 1$. We will refer to the zero-padded signals as $x_p[n]$ and $y_p[n]$, respectively. The prod-

uct of the $(2N-1)\times(2N-1)$ circulant matrix X_p generated from $x_p[n]$ and the vector $\boldsymbol{y}_p$ with elements $y_p[n]$ is equivalent to linear convolution of $x[n]$ and $y[n]$. In other words, $X_p \cdot y_p = X_e \cdot y_e$. This is evident because the first N columns of this circulant matrix are equal to X_l and the remaining $N-1$ columns are multiplied by the appended zeros in $\boldsymbol{y}_p$.

Hence, linear convolution is equivalent to circular convolution of zero-padded sequences if the length of both padded sequences is equal to the length of the result of the linear convolution, in our case $2N-1$. Actually, it may be advantageous to append even more zeros to $x[n]$ and $y[n]$ to yield sequence lengths for which particularly good FFT algorithms exist. In this case, excess zeros can simply be removed from the result.

We can summarize our observation as

$$z[n] = x[n] * y[n] = x_p[n] \circledast y_p[n] \tag{66}$$

If the length of $y[n]$ is equal to $L \neq N$, then $x_p[n]$ and $y_p[n]$ must be padded to length $N+L-1$ (or greater) for this equality to hold.

Fast Convolution via the FFT

We are now in position to exploit the fact that fast algorithms for computing the DFT of a discrete-time signal exist. Again, we focus on the case of two signals of equal length N. The fundamental idea of fast convolution algorithms is to zero pad the signals to be convolved to at least length $2N-1$. Then the DFTs of the padded sequences are computed and multiplied. Finally, the inverse DFT of the product is computed and excess zeros are removed if necessary.

How does the computational efficiency of this algorithm compare to direct evaluation of the convolution sum? Let us consider both alternatives by first looking at the corresponding MATLAB code implementations.

The direct evaluation of the convolution sum can be programmed as

```
for n=1:2*N-1
  for m=max(1,n+1-N) :min(N,n)
    z(n) = z(n) + x(m)*y(n+1-m);
  end
end
```

A little thought reveals that the innermost statement is reached N^2 times and, hence, the direct computation of the convolution sum requires N^2 additions and multiplications.

A simple, FFT-based algorithm is given by

```
LEFT = 2^ceil(log2(2*N-1)); % choose FFT length
  as power of 2

xp = zeros(1,LFFT); % zero-padding
yp = zeros(1,LFFT);
xp(1:N) = x; % set first N samples to signal
yp(1:N) = y;

Xp = fft(xp); % forward FFTs
Yp = fft(yp);
Zp = Xp.*Yp; % multiplication of DFTs
zp = ifft(Zp); % inverse FFT

z = zp(1:2*N-1); % trim excess zeros
```

In the preceding program, the length L_{FFT} of the FFT is chosen to be a power of 2 such that an efficient radix-2 (or split-radix) FFT algorithm may be employed. Then the number of additions and multiplications for each FFT is approximately proportional to $L_{\text{FFT}} \log_2 L_{\text{FFT}}$ (2). Hence, the entire algorithm requires approximately $3cL_{\text{FFT}} \log_2 L_{\text{FFT}} + L_{\text{FFT}}$ computations, where c is a constant that depends on the specific FFT algorithm used.

To illustrate these ideas, we have conducted a simple numerical experiment using MATLAB. We generated signals varying in length between 10 and 10,000 and convolved these signals using three different algorithms: direct convolution, convolution via FFTs of length equal to the smallest power of 2 greater than $2N-1$, and convolution via FFTs of length $2N-1$. For each algorithm, the number of floating point operations, both additions and multiplications, was counted using the MATLAB built-in command `flops`.

The results of this experiment are shown in Fig. 11. The figure shows that the direct convolution requires nearly exactly $8N^2$ operations. For short sequences, $N \lesssim 50$, this algorithm is the most efficient. However, for longer sequences the algorithm using FFTs of length 2^m (m integer) performs better. Furthermore, the advantage of the FFT based algorithm increases with the length of the sequence and reaches 2 orders of magnitude as $N = 10{,}000$. Notice that we must take advantage of the existence of a fast algorithm to realize a computational advantage through the use of transform-based convolution. If we rely on FFTs of length $2N-1$, there are generally no highly efficient algorithms available and the

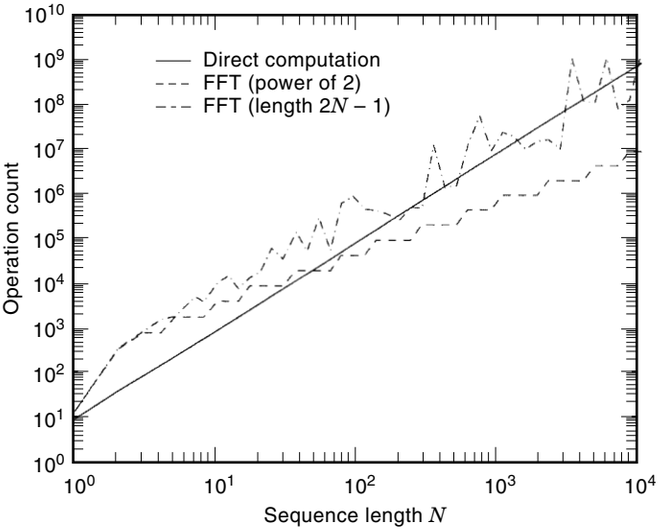


Figure 11. Computational complexity of convolution. The plot shows the number of floating point operations, additions, and multiplications, for three different convolutions algorithms as reported by the MATLAB command `flops`. All sequences are complex valued. The direct computation of the convolution sum requires nearly exactly $8N^2$ operations. For values of $N \gtrsim 50$ the number of operations for the (radix 2) FFT based algorithm is lower than for direct convolution. The third graph shows the number of operations if the length of the FFTs is not selected to be a power of 2. In this case, the transform-based algorithm is not efficient. The large variation in the operation count is related to MATLAB's FFT routine, which selects different FFT algorithms depending on the sequence length.

computational burden is often increased over direct computation of the convolution.

Sequences of Different Length

It is often necessary to convolve sequences of very different length. The impulse response of a filter $h[n]$ is typically of length 50 to 100, but the input signal $x[n]$ may consist of thousands of samples. In this case, it is possible to segment the input data into shorter blocks, perform convolution on these blocks, and combine the intermediate results. In light of our preceding discussion, the block length B of the input segments should be selected such that $B + L - 1$, where L is the filter length, provides the opportunity to employ a good FFT algorithm. For example, $B + L - 1$ can be selected to equal a power of 2.

To illustrate the reassembly of the intermediate results, let us consider the convolution of a length 3 filter $h[n]$ with a length 6 input sequence. We use two segments of block length three, such that intermediate results are of length $3 + 3 - 1 = 5$. These must be combined to yield the final result $y[n]$ of length $6 + 3 - 1 = 8$. For our illustration, we use the matrix formulation of linear convolution as in Eq. (63):

$$\begin{pmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \\ y[4] \\ y[5] \\ y[6] \\ y[7] \end{pmatrix} = \left(\begin{array}{ccc|ccc} h[0] & 0 & 0 & & & \\ h[1] & h[0] & 0 & & & \\ h[2] & h[1] & h[0] & & & \\ 0 & h[2] & h[1] & h[0] & 0 & 0 \\ 0 & 0 & h[2] & h[1] & h[0] & 0 \\ & & & h[2] & h[1] & h[0] \\ & & & 0 & h[2] & h[1] \\ & & & 0 & 0 & h[2] \end{array}\right) \cdot \left(\begin{array}{c} x[0] \\ x[1] \\ x[2] \\ \hline x[3] \\ x[4] \\ x[5] \end{array}\right) \tag{67}$$

In this example, two intermediate sequences are obtained by convolving $h[n]$ with the top and bottom half of $x[n]$, respectively. To assemble the final result requires that the two intermediate sequences are added such that they overlap by $L - 1 = 2$ samples. This method of convolving different length sequences is appropriately called overlap-add convolution.

More details on computational aspects of convolution are provided in the book by Burrus and Parks (3) or the recent book by Ersoy (4). An in-depth analysis and discussion of the state-of-the art in fast algorithms for DFT and convolution is presented in the tutorial article by Sorensen and Burrus (2).

APPLICATIONS AND EXTENSIONS

We conclude this article by looking at several applications beyond filtering and signal processing in which convolution arises naturally. Additionally, we provide pointers to several interesting extensions of the material presented herein.

Polynomial Multiplication

When two polynomials are multiplied, the result is another polynomial and the coefficients of the resulting polynomial are obtained from the original coefficients through convolution.

Let $p(x)$ and $q(x)$ be two polynomials with coefficients p_i and q_i, respectively. When these polynomials are multiplied, we obtain the polynomial $r(x)$ as

$$\begin{aligned} r(x) &= p(x) \cdot q(x) \\ &= \sum_{k=0}^{N_p} p_k x^k \cdot \sum_{l=0}^{N_q} q_l x^l \\ &= \sum_{k=0}^{N_p} \sum_{l=0}^{N_q} p_k q_l x^{k+l} \end{aligned} \tag{68}$$

Substituting $k + l = n$, the last expression can be simplified to

$$r(x) = \sum_{n=0}^{N_p+N_q} \left(\sum_{k=0}^{\min(n, N_p)} p_k q_{n-k} \right) x^n \tag{69}$$

The term in parentheses is the convolution of the sequences of coefficients of $p(x)$ and $q(x)$.

Hence, the resulting polynomial is of degree equal to the sum of the original polynomials, and its coefficients are obtained by convolving the original coefficient sequences.

Applications in Error-Correcting Coding

In all our discussions to this point, arithmetic operations were assumed to be based on real number arithmetic. Error-correcting coding relies on convolution with arithmetic over finite fields (e.g., binary arithmetic). Error-correcting coding is a field that has attracted considerable research efforts over the last 50 years, and we have to limit ourselves to simple examples here. Good introductions to the field and considerably more depth can be found in the classic book by Lin (5) or the more recent book by Wicker (6).

Cyclic Codes. Cyclic codes constitute an important class of practical error control codes and include such well-known representatives as Golay, BCH, and Reed-Solomon codes. An important reason for the continuing practical relevance of these codes is the fact that encoders and decoders can be implemented with simple, high-speed shift-register circuits. This is of great importance in high-speed communications applications.

Perhaps surprisingly, cyclic codes are based on ideas and concepts from mathematical algebra. The key to the structure of cyclic codes lies in the association of a code polynomial $c(x)$ with every code word $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1})$. Skipping many important details, we only mention that code words (i.e., information sequences with error-correction capabilities) are obtained from unprotected message words $\boldsymbol{m} = (m_0, m_1, \ldots, m_{k-1})$ through polynomial multiplication.

Specifically, a polynomial $m(x) = m_0 + m_1 x + m_2 x^2 + \cdots + m_{k-1} x^{k-1}$ is constructed and then multiplied by a suitably chosen generator polynomial $g(x)$. The selection of $g(x)$ is crucial and discussed in detail in Chapter 5 of Ref. 6 or Chapter 4 of Ref. 5. Then every code polynomial can be expressed as $c(x) = g(x) \cdot m(x)$. Since $c(x)$ is obtained by polynomial multiplication, the coefficients of $c(x)$ and, hence, the elements of the code word $\boldsymbol{c}$ are obtained by convolution

of the coefficients of $g(x)$ and $m(x)$. However, all arithmetic operations are defined over a finite algebraic field. For example, when binary arithmetic is used, the underlying field is referred to as the Galois field of size 2 and modulo 2 arithmetic is used.

To fix ideas, let us consider a well-known (7, 4) cyclic code with generator polynomial $g(x) = 1 + x + x^3$. To encode the message block $\boldsymbol{m} = (1110)$, we construct first the polynomial $m(x) = 1 + x + x^2$. Then the code polynomial $c(x)$ is computed as

$$\begin{aligned} c(x) = g(x) \cdot m(x) &= (1 + x + x^3) \cdot (1 + x + x^2) \\ &= 1 + (1 + 1)x + (1 + 1)x^2 \\ &\quad + (1 + 1)x^3 + x^4 + x^5 \\ &= 1 + x^4 + x^5 \end{aligned} \tag{70}$$

The last two lines are equal because in modulo 2 arithmetic $1 + 1 = 0$. The resulting code word is $\boldsymbol{c} = (1000110)$.

To verify that a code word has been transmitted without error, a decoder checks if the polynomial associated with a received word $\boldsymbol{r}$ is a valid code polynomial by verifying that it is divisible by $g(x)$.

While the operation of the encoder and decoder may appear awkward at first sight, they are implementable with very simple, high-speed digital hardware. Both encoder and decoder hardware can be implemented as feedback shift register circuits.

Convolutional Coding. As its name suggests, a convolutional encoder employs convolution to insert error-correction information into a sequence of information symbols. As in cyclic coding, all arithmetic operations are carried out over finite fields. While linear block codes, including cyclic codes, operate on message sequences of fixed length, convolutional codes can be used to encode message sequences that are not necessarily bounded in length.

Specifically, a convolutional encoder can be built around K shift registers with m_k memory elements, $k = 1, 2, \ldots, K$, into which the message sequence is fed. Let the message sequence be given by

$$\boldsymbol{x} = (x_0^1, x_0^2, \ldots, x_0^K, x_1^1, x_1^2, \ldots, x_1^K, \ldots, x_n^k, \ldots) \tag{71}$$

Then in symbol interval n the information symbol x_n^k is fed into the kth shift register. Also, at the beginning of symbol interval n, the kth shift register contains the information symbols $(x_{n-1}^k, x_{n-2}^k, \ldots, x_{n-m_k}^k)$. A rate K/L convolutional encoder generates L output sequences $\boldsymbol{y}^l$ by convolving (over a finite field) the information subsequences $\boldsymbol{x}^k = (x_0^k, x_1^k, \ldots, x_n^k, \ldots)$ with KL generator sequences $\boldsymbol{g}^{k,l}$ of length $m_k + 1$. Hence, the nth symbol in the lth output sequence is given by

$$y_n^l = \sum_{k=1}^{K} \left(\sum_{m=0}^{m_k} x_{n-m}^k \cdot g_m^{k,l} \right) \tag{72}$$

where all arithmetic operations are performed over a finite field (e.g., in modulo 2 arithmetic).

To fix ideas, let us consider a rate $\frac{1}{2}$ convolutional code with generator sequences $\boldsymbol{g}^{1,1} = (1011)$ and $\boldsymbol{g}^{1,2} = (1101)$. All arithmetic operations are modulo 2. Then the input sequence $\boldsymbol{x}$ yields output sequences $\boldsymbol{y}^1$ and $\boldsymbol{y}^2$ with elements

$$y_n^1 = \sum_{m=0}^{m_1} x_{n-m} g_m^{1,1} = x_n + x_{n-1} + x_{n-3} \pmod 2 \tag{73}$$

and

$$y_n^2 = \sum_{m=0}^{m_2} x_{n-m} g_m^{1,2} = x_n + x_{n-2} + x_{n-3} \pmod 2 \tag{74}$$

The operation of this convolutional encoder can be summarized by the block diagram in Fig. 12.

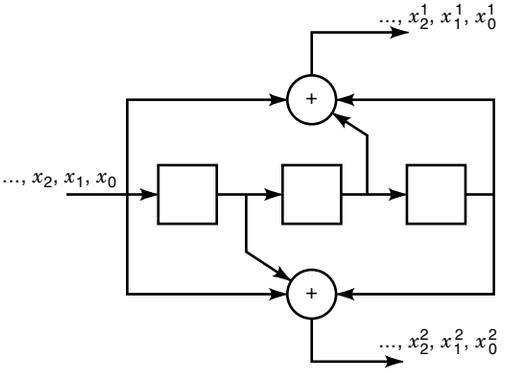


Figure 12. A rate-$\frac{1}{2}$ convolutional encoder.

Clearly, we have only scratched the surface on the topic of error-correcting codes. The inclined reader is referred to the Refs. 5 and 6 for further details.

Convolution in Statistics

While filtering of random signals and the associated convolution operation are important operations in statistical signal processing processing, the convolution operation appears in other problems of statistics as well. Background information on the concepts discussed in this section are contained in Ref. 7.

Sum of Independent Random Variables. The probability density function of the sum of two independent random variables is related to the density function of the original random variables through convolution. To begin, let X and Y denote two independent, continuous random variables with probability density functions $f_X(x)$ and $f_Y(y)$. When we form a third random variable Z as the sum of X and Y, we can determine the probability density function of Z as follows.

The distribution function $F_Z(z)$ of Z is defined as

$$F_Z(z) = \Pr(Z \le z) = \Pr(X + Y \le z) \tag{75}$$

The last term can be expressed via the joint density function $f_{XY}(x, y)$ of X and Y as

$$F_Z(z) = \Pr(X + Y \le z) = \int\int_{x+y \le z} f_{XY}(x, y)\, dx\, dy \tag{76}$$

For an independent random variable, $f_{XY}(x, y) = f_X(x)f_Y(y)$. Furthermore, the range of integration can be rewritten such that we obtain

$$F_Z(z) = \int_{-\infty}^{\infty} f_X(x) \int_{-\infty}^{z-x} f_Y(y)\, dy\, dx \tag{77}$$

Since we assumed continuous random variables, the density function $f_Z(z)$ is obtained by differentiation of $F_Z(z)$:

$$f_Z(z) = \frac{dF_Z(z)}{dz} = \int_{-\infty}^{\infty} f_X(x) f_Y(z-x)\, dx = f_X(z) * f_Y(z) \tag{78}$$

Hence, the sum of two independent random variables yields a density function that equals the convolution of the original densities. An analogous result can be derived for discrete random variables.

Finally, a transform relationship, very similar to those presented earlier, exists that captures the preceding result. For a random variable with density function $f(x)$, the moment generating function $M(j\nu)$ is defined as

$$M(j\nu) = \int_{\infty}^{\infty} f(x)e^{j\nu x}\, dx \tag{79}$$

Hence, the moment-generating function is essentially equal (except for the sign of the exponent) to the Fourier transform of the density function $f(x)$. If we denote the characteristic functions of our independent random variables X and Y by $M_X(j\nu)$ and $M_Y(j\nu)$, the characteristic function of $Z = X + Y$ is given by

$$M_Z(j\nu) = M_X(j\nu)M_Y(j\nu) \tag{80}$$

Correlation. The empirical autocorrelation of certain random processes is computed through an operation virtually identical and closely related to convolution. To be specific, let X_t denote a real-valued, wide-sense stationary random process such that its autocorrelation function $R_X(\tau)$ is given by

$$R_X(\tau) = \mathbf{E}[X_t \cdot X_{t+\tau}] \tag{81}$$

where $\mathbf{E}[\,\cdot\,]$ denotes statistical expectation.

In practice, one is often faced with the problem of estimating the autocorrelation function from a single realization $x(t)$ of X_t (observed for $0 \le t \le T$). If the random process is ergodic, we may estimate the statistical average in Eq. (81) via the time average

$$\hat{R}_X(\tau) = \frac{1}{T - |\tau|} \int_0^{T-|\tau|} x(t)x(t + |\tau|)\, dt \tag{82}$$

for $|\tau| < T$ In this expression, we have taken advantage of the symmetry property $R_X(\tau) = R_X(-\tau) = R_X(|\tau|)$.

It is easily shown that $\hat{R}_X(\tau)$ is an unbiased estimate of $R_X(\tau)$. However, the variance of $\hat{R}_X(\tau)$ becomes infinite as τ approaches T. To alleviate this problem, a weighting function, or window, may be used to ensure that the variance remains finite as τ approaches T. Further details can be found in Ref. 7, Chapter 13.

Notice that Eq. (82) bears a striking resemblance to the convolution integral of Eq. (1). In fact, it is easily verified that we can express $\hat{R}_X(\tau)$ as

$$\hat{R}_X(\tau) = w(\tau) \cdot x(\tau) * x(-\tau) \tag{83}$$

where $w(\tau) = (T - |\tau|)^{-1}$.

Thus, the empirical autocorrelation function is essentially equal to the convolution of a signal with a time-reversed version of itself. Consequently, our discussion on properties, numerical evaluation, and efficient computation of the convolution integral applies equally to the empirical autocorrelation function.

Analogous arguments can be made for discrete-time random processes, or random sequences, and for the cross-correlation function of two random processes or sequences.

Signal Spaces

Modern signal processing and control theory rely extensively on the concept of linear spaces from the mathematical field of functional analysis. The results presented here are compiled mainly from Refs. 8–10. Also, we restrict ourselves to scalar signals; most of the referenced literature treats the more general case of vector signals.

Though most of our discussion is aimed at more abstract spaces of functions (signals), it may be useful for the reader to consider the space $\mathbb{R}^N$ of length N real-valued vectors throughout our exposition for illustrative purposes.

Linear Spaces, Norms, and Inner Products. A linear space consists of a set $\mathcal{S}$ (of signals, functions, or vectors), a scalar field $\mathcal{F}$, usually the real or complex numbers, and rules that addition of elements of $\mathcal{S}$ as well as scalar multiplication of elements of $\mathcal{F}$ and $\mathcal{S}$ obey. More specifically, both the addition $x + y$ of two elements of the space $\mathcal{S}$ and the multiplication αx, $\alpha \in \mathcal{F}$, satisfy the usual laws of commutativity and associativity. Also, inverse and neutral elements exist for addition over $\mathcal{S}$, and a neutral element for scalar multiplication is contained in $\mathcal{F}$. These properties of a linear space lend a well-behaved algebraic structure to $\mathcal{S}$.

By means of a norm on the elements of $\mathcal{S}$, we can provide a topological structure for our space $\mathcal{S}$. A linear space with a norm is called a normed, linear space. A norm is simply a real-valued functions defined for all elements of S. The norm of an element $x \in \mathcal{S}$ is denoted as $\|x\|$ and must satisfy the following conditions:

1. $\|x\| \ge 0$
2. $\|x + y\| \le \|x\| + \|y\|$ (triangle inequality)
3. $\|\alpha x\| = \alpha\|x\|$, $\alpha \in \mathcal{F}$
4. $\|x\| = 0$ if and only if $x = 0$

From an engineering perspective, the most important norm is defined for the space of finite-energy signals. For reasons that will become apparent shortly, this space is conventionally denoted $\mathcal{L}_2$. The norm of a signal $x(t)$ in $\mathcal{L}_2$ is defined by

$$\|x(t)\|^2 = \int_{-\infty}^{\infty} |x(t)|^2\, dt \tag{84}$$

Hence, $\|x(t)\|^2$ equals the energy of signal $x(t)$. It is easily verified that this norm meets all four of the preceding requirements. In general, norms are useful primarily for quantifying the difference between two elements x and y of a space S through $\|x - y\|$.

Even more topological structure is induced if a space $\mathcal{S}$ also possesses an inner product. Fundamentally, an inner product introduces important geometrical concepts such as orthogonality. We may even say that an inner product space is more or less the generalization of Euclidean geometry to infinite dimensions. This leads directly to useful geometrical interpretations of many problems in signal processing and control.

The inner product is a function that associates with each pair x and y of elements of $\mathcal{S}$ a scalar. We denote the inner product of x and y as (x, y). An inner product must satisfy the following properties:

1. $(x + y, z) = (x, z) + (y, z)$ (additivity)
2. $(\alpha x, y) = \alpha(x, y)$ (homogeneity, $\alpha \in \mathcal{F}$)
3. $(x, y) = (y, x)^*$ (symmetry, $*$ denotes the complex conjugate)
4. $(x, x) > 0$, unless $x = 0$

Two observations are useful. First, any inner product space is also normed, linear space because $\|x\|^2 = (x, x)$ satisfies all requirements for a norm. However, there are many norms that cannot be expressed through an inner product. Hence, inner product spaces form a subset of normed, linear spaces. Second, the following inequality is often extremely useful, particularly in optimization:

$$|(x, y)| \leq \|x\|\|t\| \tag{85}$$

Furthermore, equality holds if and only if x and y are collinear. This result is known as the Schwarz inequality.

For the space $\mathcal{L}^2$ introduced previously, the inner product is given by

$$(x(t), y(t)) = \int_{-\infty}^{\infty} x^*(t)y(t)\,dt \tag{86}$$

The final concept we introduce is completeness. Completeness becomes important when we consider infinite sequences x_n of elements of our abstract space $\mathcal{S}$. Specifically, a normed, linear space is complete when the limit of all sequences x_n in $\mathcal{S}$ is itself an element of $\mathcal{S}$. Hence, in a complete space we may consider limits without fear that the result maybe outside of the space $\mathcal{S}$. Complete spaces play a crucial role, prompting the following terminology. A complete normed, linear space is called a Banach space and a complete inner product space is called a Hilbert space.

For example, the space $\mathbb{R}^N$ with the inner product

$$(x, y) = \sum_{k=1}^{N} x_k y_k \tag{87}$$

is a Hilbert space. The space $\mathbb{R}^N$ with norm

$$\|x\| = \sum_{k=1}^{N} |x_k| \tag{88}$$

is a Banach space.

Lebesgue and Hardy Spaces. The following Banach and Hilbert spaces are of frequent practical interest. These spaces consist of signals with certain properties and are distinct through their norm (or inner product).

The Lebesgue Space $\mathcal{L}_2$**.** As indicated previously, the space of all finite energy signals is denoted $\mathcal{L}_2$. Formally, we may say

$$\mathcal{L}_2 = \{f : \|f\|_2 < \infty\} \tag{89}$$

in which

$$\|f\|_2 = \left(\int_{-\infty}^{\infty} |f(t)|^2\,dt\right)^{\frac{1}{2}} \tag{90}$$

The space $\mathcal{L}_2$ is a Hilbert space with inner product defined by Eq. (86). Two signals are said to be orthogonal if $(x(t), y(t)) = 0$. This provides a natural extension of orthogonality in $\mathbb{R}^N$.

Alternatively, we can consider the norm and inner product of the Fourier transforms of signals. Hence, the inner product of Fourier transforms $X(f)$ and $Y(f)$ is defined completely analogously to the time domain counterpart of Eq. (86) as

$$(X(f), Y(f)) = \int_{-\infty}^{\infty} X^*(f)Y(f)\,df \tag{91}$$

Using the definition of the Fourier transform in Eq. (22), we can rewrite the last expression as

$$\begin{aligned}(X(f), Y(f)) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x*(t)e^{j2\pi ft}\,dt \int_{-\infty}^{\infty} y(u)e^{j2\pi fu}du\,df \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(t)y(u)\left(\int_{-\infty}^{\infty} e^{-j2\pi f(u-t)}\,df\right)dt\,du\end{aligned} \tag{92}$$

The expression in parentheses equals $\delta(u - t)$, and hence the entire expression simplifies to

$$\begin{aligned}(X(f), Y(f)) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x(t)y(u)\delta(u-t)\,dt\,du \\ &= \int_{-\infty}^{\infty} x(t)y(t)\,dt \\ &= (x(t), y(t))\end{aligned} \tag{93}$$

This result is known as Parseval's theorem. It establishes that there exists a so-called isomorphism between the time-domain and frequency-domain versions of $\mathcal{L}_2$.

The Hardy Space $\mathcal{H}_2$**.** Hardy spaces in general contain only signals whose Laplace transforms $X(s)$ are analytic in the right half-plane $\Re(s) > 0$ [i.e., if $X(s)$ is rational it does not have poles in the right half plane]. The Hardy space $\mathcal{H}_2$ is formally defined as the set of signals with Laplace trans-

forms $X(s)$ such that $X(s)$ is analytic for $\Re(s) > 0$ and the following norm is finite:

$$\|X(s)\|_2 = \left(\sup_{\alpha>0} \int_{-\infty}^{\infty} X^*(\alpha + j2\pi f) X(\alpha + j2\pi f)\, df \right)^{\frac{1}{2}} < \infty \tag{94}$$

It can be shown that the norm always assumes its supremum as α approaches zero. If we define $X_b(f) = \lim_{\alpha \downarrow 0} X(\alpha + j2\pi f)$, we may replace this norm by the simpler $\mathscr{L}_2$ norm

$$\|X(s)\|_2 = \left(\int_{-\infty}^{\infty} X_b^*(f) X_b(f)\, df \right)^{\frac{1}{2}} \tag{95}$$

Thus, we may regard $\mathscr{H}_2$ as a proper subspace of $\mathscr{L}_2$. Furthermore, by the Paley–Wiener criterion we can conclude that $\mathscr{H}_2$ is isomorphic to the subspace of $\mathscr{L}_2$ that contains only right-sided signals [i.e., signals such that $x(t) = 0$ for $t < 0$]. $\mathscr{H}_2$ is a Hilbert space.

The Hardy Space $\mathscr{H}_\infty$. The Banach space $\mathscr{H}_\infty$ is the space of all signals whose Laplace transform is not only analytic in the right half plane but bounded. The norm for $\mathscr{H}_\infty$ is given by

$$\|X(s)\|_\infty = \sup_f |X_b(f)| \tag{96}$$

where, as before, $X_b(f) = \lim_{\alpha \downarrow 0} X(\alpha + j2\pi f)$.

As we will see shortly, the space $\mathscr{H}_\infty$ plays a crucial role in robust control theory.

Examples. To illustrate the usefulness of the concepts introduced previously, we consider two representative examples from the areas of signal processing and control.

Optimum, Binary Detection. In a simple binary communication system, one of two equally likely signals $s_0(t)$ or $s_1(t)$ is transmitted to convey one bit of information. We assume that each signal is of finite duration T and during transmission the signal is corrupted by white Gaussian noise with autocorrelation function $N_0/2\ \delta(\tau)$.

A crucial aspect in the receiver is the design of a linear filter that maximizes the ability to distinguish which of the two possible signals was transmitted. If we denote the impulse response of the filter by $h(T - t)$ and sample the output of the filter at time $t = T$, then a random variable R with conditional Gaussian distribution is obtained.

Specifically, if $s_0(t)$ was transmitted, then the mean μ_0 and variance σ_0^2 of R are given by

$$\begin{aligned} \mu_0 &= s_0(t) * h(T-t)|_{t=T} = \int_0^T s_0(t) h(t)\, dt = (s_0(t),\ h(t)) \\ \sigma_0^2 &= \frac{N_0}{2} \int_0^T |h(t)|^2\, dt = \frac{N_0}{2} \|h(t)\|^2 \end{aligned} \tag{97}$$

Similarly, if $s_1(t)$ was transmitted, we obtain

$$\begin{aligned} \mu_1 &= s_1(t) * h(T-t)|_{t=T} = \int_0^T s_1(t) h(t)\, dt = (s_1(t),\ h(t)) \\ \sigma_1^2 &= \frac{N_0}{2} \int_0^T |h(t)|^2\, dt = \frac{N_0}{2} \|h(t)\|^2 \end{aligned} \tag{98}$$

It is straightforward to demonstrate that for equally likely signals, the probability of error is given by

$$P_e = Q\left(\frac{(s_0(t) - s_1(t),\ h(t))}{\sqrt{2N_0} \cdot \|h(t)\|} \right) \tag{99}$$

where

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}\, dy \tag{100}$$

Thus, to minimize the probability of error of the receiver, we must choose $h(t)$ to maximize the ratio

$$\frac{(s_0(t) - s_1(t),\ h(t))}{\sqrt{2N_0}\|h(t)\|} \tag{101}$$

However, because of the Schwarz inequality, we know immediately that we must select $h(t) = s_0(t) - s_1(t)$ and the resulting probability of error equals

$$P^e = Q\left(\frac{\|s_0(t) - s_1(t)\|}{\sqrt{2N_0}} \right) \tag{102}$$

The filter with impulse response $h(T - t)$ is known as the matched filter for the signals set $s_0(t)$ and $s_1(t)$.

Robust Control. A fundamental problem in robust control is to ensure that a system is designed such that its output in response to any finite energy signal has itself finite energy. This is crucial in systems with noise or other disturbances that can only be bounded in energy.

Let $x(t)$ denote the input to a system with impulse response $g(t)$ and let $y(t)$ be the resulting output. Then we would like to ensure that the ratio of output energy to input energy remains finite for all possible inputs. In terms of the norms defined previously, we can formulate this problem by defining the gain G as

$$\begin{aligned} G &= \sup_{u(t) \neq 0} \frac{\|g(t) * u(t)\|_2}{\|u(t)\|_2} \\ &= \sup_{U(f) \neq 0} \frac{\|G(f)U(f)\|_2}{\|U(f)\|_2} \end{aligned} \tag{103}$$

Here we have employed the transform property of convolution. We can invoke the Schwarz inequality again and further simplify the last expression to obtain

$$G = \sup_f |G(f)| = \|G(s)\|_\infty \tag{104}$$

Thus, the $\mathscr{H}_\infty$ norm measures the maximum possible increase in signal energy for all possible finite energy inputs. Because of the preceding considerations, we say that the $\mathscr{H}_\infty$ norm is induced by the cal H_2 norm on the input and output signals.

SUMMARY

In this article, we have examined in some detail the convolution operation. We have seen that convolution is an operation

that is fundamental for all linear, time-invariant systems. After examining important properties of convolution, including several transform properties, we turned our attention to the numerical evaluation of the continuous-time convolution integral. Then we discussed possible approaches for computationally efficient convolution algorithms, emphasizing algorithms based on the fast Fourier transform.

We concluded by examining several applications in which convolution or related operations arise, including error correcting coding and correlation. Finally, we gave a brief introduction to the concept of abstract signal spaces.

BIBLIOGRAPHY

1. R. Gray, *Toeplitz and circulant matices: A review* [Online], Technical report, Stanford University, 1971 (revised 1977, 1993, 1997). (Available www:http://www-isl.stanford.edu/~gray/toeplitz.pdf)
2. H. V. Sorensen and C. S. Burrus, Fast dft and convolution algorithms, in S. K. Mitra and J. F. Kaiser (eds.), *Handbook for Digital Signal Processing,* New York: Wiley, 1993, pp. 491–610.
3. C. S. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms—Theory and Implementation,* New York: Wiley, 1985.
4. O. Ersoy, *Fourier-Related Transforms, Fast Algorithms and Applications,* Upper Saddle River, NJ: Prentice Hall, 1997.
5. S. Lin, *An Introduction to Error-Correcting Codes,* Englewood Cliffs, NJ: Prentice-Hall, 1970.
6. S. B. Wicker, *Error Control System for Digital Communiations and Storage,* Englewood Cliffs, NJ: Prentice-Hall, 1995.
7. A. Papoulis, *Probability, Random Variables, and Stochastic Processes,* 3rd ed., New York: McGraw-Hill, 1991.
8. J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory,* New York: Macmillan, 1992.
9. D. G. Luenberger, *Optimization by Vector Space Methods,* New York: Wiley, 1969.
10. A. W. Naylor and G. R. Sell, *Linear Operator Theory in Engineering and Science,* vol. 40 of *Applied Mathematical Sciences,* New York: Springer-Verlag, 1982.

BERND-PETER PARIS
George Mason University

/ /

Correlation Theory
Standard Article

David J. Edelblute[1]
[1]SPAWAR Systems Center San Diego, San Diego, CA

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2407
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (242K)

Abstract

The sections in this article are

Physical Measurements and the Decibel Scale

Transformations

The Detection Problem

Least-Squares Prediction and Estimation

Maximum Likelihood, Cramér–Rao, and Fisher's Information Matrix

Fourier Transforms and Spectrum Estimation

Stationarity Issues

Bandwidth and Time–Bandwidth Products

Single-Waveform Testing and Square Law Detectors

Two-Channel Detection

Gaussian Distributions

Likelihood Detectors for Gaussian Noise

Other Distributions

Future Trends

Acknowledgments

| | |

Copyright © 1999-2008 All Rights Reserved.

CORRELATION THEORY

In the vernacular, if two variables are correlated, then they are somehow related. In scientific discussion the term correlation has a more limited and specific meaning. If two variables are said to be correlated, this means that both variables can be characterized by real or complex numbers. It also implies that either of the variables can be used to predict the other. More specifically, the prediction can be accomplished by a linear function. For example, the variable θ and the function $\cos\theta$ are uncorrelated. The function can be predicted exactly from the variable, but the prediction method is not linear.

These (usually unspoken) assumptions are implicit in all correlation analyses: numerical representation, linear prediction, and judgment of the prediction by a least-squares, or root mean square (rms), criterion.

The problem of finding a numerical representation for the data is not always easy. Most of the following discussion will assume that the data originate as time-dependent waveforms. However, these long strings of numbers are rarely immediately useful. There are too many of them, and most of them contain little or no useful information. The first task is usually to extract parameters from these waveforms. Often correlation analysis is used first to extract the parameters and then to analyze the parameters.

It is always legitimate to question whether only linear functions should be considered. In many cases there is good reason to believe that a nonlinear function is appropriate. The problem of fitting a nonlinear function to a data set is not always more difficult than fitting a linear function. The difficulty is that there are no standard techniques that lend themselves to routine use. Often it is necessary to devise a new approach for each problem. For example, suppose that one needs to fit a nonlinear function of x to a data set y over a specific interval by adjusting parameters a, b, and c. It may happen that a dominates the function over a portion of the interval and then becomes unimportant over the rest of the interval. Similarly, b may have little or no effect on the peak value of the function but control the rate of decay of the function. One must first examine the function to see what effect each parameter has, and then perhaps adjust them separately. This may be fairly easy to do by inspection, but difficult to automate. By contrast, linear functions give equal importance to each variable and to all parts of the interval (although it is possible to weight some parts of the interval more heavily than others in many cases).

The insistence on linear functions for prediction is sometimes compromised. For example, a polynomial may be treated as a linear sum of powers of the variable. Other nonlinear functions may be inserted into the summation with linear multipliers.

The choice of an rms criterion for judging the quality of a predictor is not necessarily obvious and should not always be taken for granted. In some cases it may be more important to minimize the worst-case error than the rms error. This criterion leads to a minimax problem. In a few cases the average of the absolute value of the error may be a better criterion. This may lead to median estimators.

However, the rms criterion has proven to be by far the most fruitful assumption in the majority of cases. This is due largely to the ease with which second moments of rms solutions can be followed through linear transformations of the variables.

It is important to understand that correlation analysis can never prove a cause-and-effect relationship. If two variables are causally related, correlation analysis cannot determine which variable is the cause. Often both variables are caused by a third variable that is not even known. About the only thing that can be said with certainty is that if two variables are independent then they are uncorrelated.

However, correlation analysis is an important tool to study relationships of many types. That two variables are uncorrelated is not reason enough to dismiss the possibility that they are related. But if they are correlated it is reasonable to try to figure out why. Also, the absence of a predicted correlation can lead to important discoveries.

PHYSICAL MEASUREMENTS AND THE DECIBEL SCALE

In this discussion, the key quantities of interest are often ratios of averages of squares of variables. In physical systems, the variables often are voltages or pressures or flux densities. In these cases the physical power is proportional to these mean squared values. Discussion of these power levels often involves several problems.

First, the quantities often vary over ranges that are difficult to imagine. The quietest sound that a human can hear corresponds to a pressure on the order of 0.00002 Pa (rms), while the sound level on a jet plane can be over 5 Pa. Second, the uncertainty of the measurement often varies with the level. A good acoustic measurement may have an uncertainty

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

of 20%, making measurements at low levels appear much better than measurements at higher levels.

The usual solution to these problems is the decibel (dB) scale. The decibel scale is a logarithmic scale. However, a common logarithmic scale is felt to be a bit too coarse, so the logarithm is multiplied by 10. The key to understanding the decibel scale is to remember that by convention it is always a ratio of power, or energy values. Suppose the rms acoustic pressure in a room is 0.2 Pa. An acoustician, remembering that acoustic power goes as the square of the rms pressure, might compute $10 \log(0.2^2/0.00002^2) = 80$ and say, "The sound level in the room is 80 dB relative to 0.00002 Pa." Of course, he or she could get the same answer by computing $20 \log(2/0.0002)$, so it is often said that the sound level goes as 20 times the logarithm of the pressure.

In some instrumentation problems it can be tricky to keep track of whether the values should be plotted as 10 log or 20 log. The key to keeping it straight is to ask, "How would the quantity behave if the power were doubled?"

In the same vein, the decibel scale says nothing about the units of measurement. The reference to 0.00002 Pa is a specification of a physical state, not a system of units. If one cannot relate the variables to a power level, then the decibel scale is not appropriate. Engineers will occasionally make comments like "His salary went up by 1 dB when he got the promotion." The implied humor is that the speaker is also saying "Money is power."

TRANSFORMATIONS

Modern data collection problems tend to involve great amounts of data, most of which have no value. It is important to select the small subset of the data that is of potential value. The solution is to attempt to transform the data in such a way as to concentrate the important information into a small number of parameters. The most effective way to do this is usually with the Fourier transform.

In its most common form, the Fourier transform represents the data as a summation of sine and cosine waves, or complex exponentials. Other function sets are sometimes used, (e.g., Walsh functions or Bessel functions), but not often. The reason for the preeminence of complex exponentials as basis functions is the ease with which time translations are handled. Often the data look the same from one time to another and absolute time has no physical significance in interpreting the waveform. (This assumption is referred to as *time stationarity*. Various types of stationarity are defined, depending on how rigorous a concept of time stationarity is needed, but the general idea is that it is impossible to infer absolute time from the waveform.) Even if the waveform is impulsive, it is confusing if its representation changes drastically with arbitrary shifts of the time origin, as happens with some of the alternatives to the complex exponentials.

This invariance with respect to the start time of the signal gives rise to another important aspect of the complex exponentials. They look the same after they have been operated on by a linear filter (i.e., the complex exponentials are eigenfunctions of linear differential equations). This is the key idea. If a summation of complex exponentials is passed through the linear filter, the filter may amplify or delay each component by a different amount, but it does not mix the different frequencies. The output at each frequency depends only on the input at that frequency and is independent of any other frequency.

Thus, the most useful approach known for studying how a waveform will change as it passes through a linear filter is as follows: First the waveform is represented as a summation of complex exponentials at various frequencies. Then the effect on each frequency is calculated separately to see how its amplitude and phase will change as it passes through the filter. Finally, the altered exponentials are summed to give the waveform that emerges from the filter. Since so much of our world is governed by linear differential equations, the importance of understanding waveforms in terms of their Fourier representation is difficult to exaggerate.

The Fourier transform is often best understood by thinking of it as a frequency shift followed by a low-pass filter. For a frequency of interest, the waveform is frequency-shifted by multiplying it by the sine and cosine waves, or the complex exponential wave. This shifts the information of interest to the band around zero frequency. The waveform is then low-pass-filtered to eliminate information at other frequencies, and the result is the Fourier coefficient that describes the waveform at the frequency of interest.

It is tempting to believe that the complex exponentials are only mathematical artifacts, while reality is restricted to the wave as it is represented in time. Common experience refutes this. For example, an AM radio receives an electrical signal that is mostly meaningless noise. The circuitry then separates out the real information for the station of interest and sends the resulting Fourier coefficients to the speaker to produce meaningful sounds (plus advertisements and political commentary).

The Fourier coefficients are usually represented by complex numbers, so the theory of complex numbers is intimately connected with many engineering problems.

The terminology of complex variables is misleading and unfortunate. The name "complex numbers" suggests that they are difficult to handle. In fact, complex variables are popular because their use greatly simplifies many problems. The terminology also suggests that the "imaginary" part of the complex variable is somehow less intimately connected with reality than the "real" part. This idea is dangerously wrong. For example, electrical circuits sometimes develop large imaginary voltages. These imaginary voltages can cause arcing between supposedly isolated parts of the circuit. They can break down circuit components, and they can kill unwary handlers.

In the following discussion, the term *analytic* will be used to describe a function that is analytic in the sense of complex analysis theory. Any of several equivalent definitions may be used. For example, a function is analytic if it has a derivative in the ordinary sense, or if it is the derivative of another function, or if it has a power series (Taylor series) representation. In the same vein, an *analyticity* is a point at which a function is analytic. When a function is analytic that fact has profound implications, most of which are far beyond the scope of this article.

THE DETECTION PROBLEM

Correlators are often used to decide the presence or absence of a particular signal. The investigator begins with a wave-

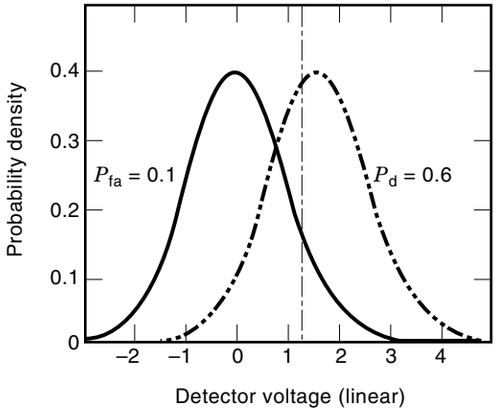


Figure 1. Probability density curves for noise only (left) and signal plus noise (right) for matched filter detector. The threshold is chosen for a false-alarm rate of 10%. The detector characteristics are determined entirely by the ratio of the horizontal separation (the signal strength) and the standard deviation.

form that may or may not contain the signal. By correlating the waveform with the signal, a value is obtained whose statistics depend on the amount of signal energy in the waveform. The analytic details will be discussed below, but the reasoning used for the test is illustrated in Fig. 1.

Figure 1 shows the probability density function for the correlator output when the signal is absent and when it is present. In each case, the probability density function is bell-shaped. Sometimes the function is truly Gaussian, and sometimes it can be approximated as Gaussian. (This approximation is sometimes dangerous, as will be discussed below.) A threshold has been established, which is indicated by the vertical line, and if the correlator output is above this threshold the equipment is to issue an alarm signal.

The probability of a false alarm is the area under the left curve that is to the right of the threshold. In the illustration, a threshold has been set to provide a 10% probability of a false alarm (P_{fa} = 10%). That is, when no signal is present, the correlator will produce an alarm 10% of the time. The signal strength is represented by the horizontal separation of the two curves. When the signal is present, the probability of detecting it is the area under the right curve that is to the right of the threshold. In this case, the signal energy is strong enough that, when present, it will be correctly detected 60% of the time. This is the probability of detection (P_d = 60%). The areas under the curves to the left of the threshold give the probability of a correct dismissal (P_{cd} = 90%) and the probability of a miss (P_m = 40%).

Figure 1 also illustrates several other important concepts. The noise is measured not by the average level, but by the standard deviation of the noise-only distribution. The important measure of a signal is the ratio of the signal strength to the standard deviation of the noise. The detector performance is characterized by this ratio and the threshold.

In most problems, a 10% false-alarm rate is too high. In standard statistical testing one often talks about "confidence" values of 5% or 1%. However, in most signal-processing applications the false-alarm rate must be several orders of magnitude lower for the system to be useful. That is because the rate is an individual-detection value. Consider, for example, a multibeam radar. Returns from a single pulse may come in from 100 directions. In each direction there may be on the order of 10,000 range bins. This means that on each pulse, there are on the order of 10^6 opportunities for a false alarm. In order to avoid overloading the operator, it may be necessary to keep the system false-alarm rate below about one per 100 pulses.

Figure 2 shows another way to analyze the performance of a detector. This is the same detector discussed in Fig. 1, but the signal strength is now treated as a variable. This means that for a given threshold setting the probability of detection depends on the signal strength. Figure 2 shows how the probability of detection varies with the signal-to-noise ratio. In this case, the threshold has been set for a false-alarm rate of 10^{-6}. Figure 2 illustrates an important point that is common with most detectors operating at these low false-alarm rates: The transition from a very low probability of detection to a very high probability of detection occurs over a fairly narrow decibel range. This is consistent with experience in auditory testing. Initially, the investigator sets the signal strength very low and the subject hears nothing. As the signal strength is increased, at some point the subject begins to hear the signal very faintly and with much uncertainty. By the time the signal strength has increased 3 dB beyond that point, the subject hears the signal clearly and calls it with no hesitation.

For this reason, there is usually no need to measure the probability of detection very accurately. Once the signal-to-noise ratio necessary for a 50% probability of detection is established, the 10% and 90% values are not far away.

The lower asymptote in Fig. 2 is not zero. It is the false-alarm rate. This is easy to see in Fig. 1, where the distributions become identical as the signal strength goes to zero.

LEAST-SQUARES PREDICTION AND ESTIMATION

Minimum-mean-squared-error estimation is among the most fruitful problems that have been investigated. Other criteria for goodness of fit have often been suggested, and in some cases may be more appropriate. However, they have not proven as rich in implications.

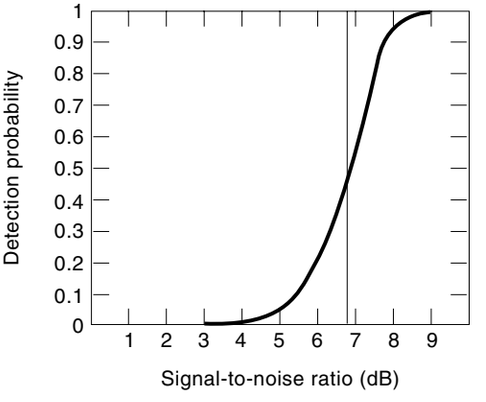


Figure 2. Probability of detection versus signal-to-noise ratio for the detector in Fig. 1. The false-alarm rate has been reduced to one per million. The transition from low probability of detection to high probability of detection occurs over a range of about 2 dB.

In the following discussion, the expectation of a variable, x, will be denoted by $\mathrm{E}[x]$, while the average of the available data will be denoted by $\langle x \rangle$. The transpose of a matrix or vector $\boldsymbol{x}$ will be denoted by $\boldsymbol{x}^{\mathrm{T}}$, while the complex conjugate of the transpose will be denoted by $\boldsymbol{x}^{\mathrm{H}}$. The trace of a matrix, A, will be denoted by $\mathrm{tr}\, A$.

In its simplest form the problem is to estimate variable y from another variable, x. Since mean values are easy to add or remove, usually nothing is lost by assuming that both x and y are zero-mean. This means that the estimate $\hat{y}$ is equal to ax. The error criterion, or goodness-of-fit criterion, is $\epsilon = \mathrm{E}[(y - \hat{y})^2] = \mathrm{E}[y^2] - 2a\mathrm{E}[xy] + a^2\mathrm{E}[x^2]$, which is easily reduced to

$$\epsilon = \mathrm{E}[y^2]\left(1 - \frac{\mathrm{E}[xy]^2}{\mathrm{E}[x^2]\mathrm{E}[y^2]}\right) + \mathrm{E}[x^2]\left(a - \frac{\mathrm{E}[xy]}{\mathrm{E}[x^2]}\right)^2$$

This equation illustrates several common terms. Obviously the error ϵ is a minimum when the second term is zero, so the optimum coefficient is $a_{\mathrm{o}} = \mathrm{E}[xy]/\mathrm{E}[x^2]$. The quantity $\mathrm{E}[y^2]$ is called the *variance* of y, while the quantity $\mathrm{E}[xy]$ is called the *covariance* between x and y. The covariance can be normalized to give a *correlation coefficient* between x and y, $\mathrm{E}[xy]/\sqrt{\mathrm{E}[x^2]\mathrm{E}[y^2]}$, which is limited to the range between -1 and 1. The square of this quantity, $\mathrm{E}[xy]^2/(\mathrm{E}[x^2]\mathrm{E}[y^2])$, is called the *coherence* between x and y. The important point is that the coherence is the fraction of the variance of y that can be removed by the linear predictor. If the problem were turned around, so that y was used to predict x, the same coherence would still predict the fraction of the variance of x that could be removed.

This pattern of analysis also works when multiple variables are involved. In this case, it is convenient to group the coefficients and the independent variables, which may be complex, into column vectors, $\boldsymbol{a}$ and $\boldsymbol{x}$. The variable, y is then estimated by a scalar product $\hat{y} = \boldsymbol{a}^{\mathrm{H}}\boldsymbol{x}$. The error criterion is $\epsilon = \mathrm{E}[(y - \boldsymbol{a}^{\mathrm{H}}\boldsymbol{x})^*(y - \boldsymbol{a}^{\mathrm{H}}\boldsymbol{x})] = \mathrm{E}[y^*y] - \boldsymbol{a}^{\mathrm{H}}\mathrm{E}[y^*\boldsymbol{x}] - \mathrm{E}[y\boldsymbol{x}^{\mathrm{H}}]\boldsymbol{a} + \boldsymbol{a}^{\mathrm{H}}\mathrm{E}[\boldsymbol{x}\boldsymbol{x}^{\mathrm{H}}]\boldsymbol{a}$. Again, it is convenient to define correlation quantities. The covariance matrix of $\boldsymbol{x}$ is $C_x = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^{\mathrm{H}}]$. If $\boldsymbol{v} = \mathrm{E}[y^*\boldsymbol{x}]$ and $\sigma_y = \mathrm{E}[y^*y]$, then

$$\epsilon = \sigma_y - \boldsymbol{v}^{\mathrm{H}}C_x^{-1}\boldsymbol{v} + (\boldsymbol{a} - C_x^{-1}\boldsymbol{v})^{\mathrm{H}}C_x(\boldsymbol{a} - C_x^{-1}\boldsymbol{v})$$

Since a covariance matrix is necessarily positive definite, the last term is greater than or equal to zero, and can only be zero if $\boldsymbol{a}_{\mathrm{o}} = C_x^{-1}\boldsymbol{v}$, in which case the minimum mean squared error is $\epsilon_{\mathrm{o}} = \sigma - \boldsymbol{v}^{\mathrm{H}}C^{-1}\boldsymbol{v}$. This takes a more interesting form if one uses the total covariance matrix

$$C = \mathrm{E}\left[\begin{bmatrix} y \\ \boldsymbol{x} \end{bmatrix}\begin{bmatrix} y^* & \boldsymbol{x}^{\mathrm{H}} \end{bmatrix}\right] = \begin{bmatrix} \sigma_y & \boldsymbol{v}^{\mathrm{H}} \\ \boldsymbol{v} & C_x \end{bmatrix}$$

because the inverse is

$$C^{-1} = \begin{bmatrix} \dfrac{1}{\epsilon_{\mathrm{o}}} & -\dfrac{1}{\epsilon_{\mathrm{o}}}\boldsymbol{a}_{\mathrm{o}}^{\mathrm{H}} \\ -\dfrac{1}{\epsilon_{\mathrm{o}}}\boldsymbol{a}_{\mathrm{o}} & \left(C_{\boldsymbol{x}} - \dfrac{1}{\sigma_y}\boldsymbol{v}\boldsymbol{v}^{\mathrm{H}}\right)^{-1} \end{bmatrix}$$

By interchanging rows and columns it is easy to see that, in terms of the total covariance matrix and its inverse, it is arbitrary which element is being predicted. It follows that if a variable becomes extremely predictable, the corresponding diagonal element in the inverse will become very large. If any variable becomes completely predictable, the covariance matrix becomes singular.

This also provides a way to study multiple coherence. The multiple coherence of y with respect to a set of variables $x_1 \ldots x_n$, denoted $\mathrm{Coh}_{y;x_1,\ldots,x_n} = 1 - \epsilon_{\mathrm{o}}/\sigma_y$, is the fraction of the variance of σ_y that can be removed by a linear predictor based on the x's.

This forms the basis of some valuable methods for screening data. By computing the covariance matrix of experimental variables one can look for large correlations. This may be easier if the matrix is normalized so that the diagonal elements are all one, that is, $C_{i,j} \to C_{i,j}/\sqrt{C_{i,i}C_{j,j}}$. When the matrix is inverted, the diagonal elements will indicate if any variable is predicted especially well or poorly. If one diagonal element in the inverse is unusually small, it may be an indication that that variable somehow does not belong with the others. If one diagonal element of the inverse is unusually large, it can indicate that a variable is almost completely predicted from the others and therefore contributes little information to the data set. In the same vein, the optimum linear predictor for any variable can be extracted from the row or column of the inverse containing the corresponding diagonal element. The rows or columns of the inverse matrix can also be interpreted as a data-whitening filter.

The above pattern holds when the problem is generalized. A vector $\boldsymbol{y}$ can be estimated by a linear transformation $\hat{\boldsymbol{y}} = A^{\mathrm{H}}\boldsymbol{x}$. The important correlation matrices are $C_x = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^{\mathrm{H}}]$, $C_y = \mathrm{E}[\boldsymbol{y}\boldsymbol{y}^{\mathrm{H}}]$, and $V = \mathrm{E}[\boldsymbol{x}\boldsymbol{y}^{\mathrm{H}}]$. In this case, the error quantity is $\boldsymbol{\epsilon} = \mathrm{E}[(\boldsymbol{y} - \hat{\boldsymbol{y}})^{\mathrm{H}}(\boldsymbol{y} - \hat{\boldsymbol{y}})] = \mathrm{tr}\ \mathrm{E}[(\boldsymbol{y} - A^{\mathrm{H}}\boldsymbol{x})(y - A^{\mathrm{H}}x)^{\mathrm{H}}] = \mathrm{tr}[(A - C_x^{-1}V)^{\mathrm{H}}C_x(A - C_x^{-1}V)] + \mathrm{tr}(C_y - V^{\mathrm{H}}C_x^{-1}V)$. This, of course, immediately gives $\epsilon_{\mathrm{o}} = \mathrm{tr}(C_y - V^{\mathrm{H}}C_x^{-1}V)$ and $A_{\mathrm{o}} = C_x^{-1}V$. The total covariance matrix and its inverse take the form

$$\begin{bmatrix} C_y & V^{\mathrm{H}} \\ V & C_x \end{bmatrix}^{-1} = \begin{bmatrix} R_y & -(C_y^{-1}V^{\mathrm{H}})R_x \\ -A_{\mathrm{o}}R_y & R_x \end{bmatrix}$$

where $R_y = (C_y - V^{\mathrm{H}}C_x^{-1}V)^{-1}$ and $R_x = (C_x - VC_y^{-1}V^{\mathrm{H}})^{-1}$.

At first glance it may seem that the introduction of complex variables is an unnecessary complication. After all, the complex numbers can be treated as pairs of real numbers, so by doubling the size of the matrices we can solve the problem using only real variables. However, we cannot easily solve quite the same problem. The form $\hat{y} = ax$ carries an analytic assumption. For example, the solution can never take on the form $\mathrm{Re}\ y = \mathrm{Re}\ x + \mathrm{Im}\ x$, $\mathrm{Im}\ y = \mathrm{Re}\ x + \mathrm{Im}\ x$, because this would not be an analytic function. In order to make the problem equivalent one would have to pose the estimation as $\hat{y} = ax + bx^*$, giving up the analytic assumption. Usually the choice of complex variables for the original problem statement is determined by the physics of the problem. Thus, the use of complex variables injects certain *a priori* assumptions. If the complex functions seem appropriate to the problem definition, one should be careful about assuming that real variables could produce a sensible solution. Only if a nonanalytic solution can easily be given a physical interpretation should real variables be considered.

For the same reason, the most general form of the linear estimation procedure is rarely seen. It is $\hat{\boldsymbol{y}} = A\boldsymbol{x} + B\boldsymbol{x}^*$. However, even if an analytic solution is not necessary, it may be best to work the problem with complex variables in order to more easily give a physical interpretation to the problem definition or the solution.

MAXIMUM LIKELIHOOD, CRAMÉR–RAO, AND FISHER'S INFORMATION MATRIX

The predictors in the previous section are based on the constraint that a linear function is to be used. An obvious question is whether some nonlinear predictor could do better. As will be seen below, if the data are Gaussian, the answer is no. In non-Gaussian cases the minimum-mean-squared-error predictor is often difficult or impossible to find. However, even when this predictor cannot be found, it is sometimes possible to obtain bounds on how well any predictor can perform. Any candidate prediction function can then be compared with those bounds.

The most popular method to find such performance bounds is to use the Cramér–Rao inequality. The reasoning goes as follows.

An unknown quantity, A, is to be estimated. Here, A is a real number. Although A cannot be directly observed, an experiment is run that produces a real-number result, R, which depends in part on A. That is, the probability density function of R depends on A and can be written as $\text{prob}_{R|A}(r|A)$. The investigator must now make an estimate of A. The estimate, which depends on R, is denoted by $\hat{a}(R)$. The estimate may have a bias, $\beta(A) = \text{E}[\hat{a}(R) - A]$. The question is "How good, in a mean-squared-error sense, can $\hat{a}(R)$ be?" The Schwarz inequality can be used to show that

$$\text{E}\left[[\hat{a}(R) - A]^2\right] \geq \frac{\left(1 + \frac{d}{dA}\beta(A)\right)^2}{\text{E}\left[\left(\frac{\partial}{\partial A}\ln \text{prob}_{R|A}(R|A)\right)^2\right]} = \frac{\left(\frac{d}{dA}\text{E}[\hat{a}(R)]\right)^2}{\text{E}\left[\left(\frac{\partial}{\partial A}\ln \text{prob}_{R|A}(R|A)\right)^2\right]}$$

or, equivalently,

$$\text{E}\left[[\hat{a}(R) - A]^2\right] \geq \frac{-\left(\frac{d}{dA}\text{E}[\hat{a}(R)]\right)^2}{\text{E}\left[\frac{\partial^2}{\partial A^2}\ln \text{prob}_{R|A}(R|A)\right]}$$

This is of most interest when the estimator is unbiased, that is $\beta(A) = 0$. In this case the numerator of the right side of the above equations becomes one, and the right side of the equations is independent of the estimating procedure, $\hat{a}(R)$. Thus, for any unbiased estimator, one can arrive at bound on the goodness of the estimator in a mean-squared-error sense. Any estimator that gives equality with the bound is said to be *efficient.* No unbiased estimator can do better.

The above argument may give a good lower bound on the error, but it gives no help in finding a way to achieve that bound. One of the most intuitive lines of reasoning leads to the maximum likelihood estimator. It seems unreasonable to assume that the observation R was extremely improbable, given the true value of A. The extension of that idea is that the best guess for A is the one that would have made R seem most likely. The maximum likelihood estimate is the value of A that would maximize the probability density function $\text{prob}_{R|A}(R|A)$—in other words, the value of A that solves

$$\frac{d}{dA}\text{prob}_{R|A}(R|A) = 0$$

Often it is easier to solve

$$\frac{d}{dA}\ln \text{prob}_{R|A}(R|A) = 0$$

It turns out that if the maximum likelihood estimator exists, and if it is unbiased, then the maximum likelihood estimator is efficient.

Maximum likelihood estimators are often used with good results. For example, suppose that R is a Gaussian variable with unit variance and unknown mean. In order to estimate the mean from a single sample, consider that the logarithm of the probability density function is

$$\ln \text{prob}_{R|A}(R|A) = -\tfrac{1}{2}\ln(2\pi) - \tfrac{1}{2}(R - A)^2$$

so the maximum likelihood estimator is $\hat{A} = R$. The mean squared error is

$$-\left(\frac{\partial^2}{\partial A^2}\left[-\tfrac{1}{2}\ln 2\pi - \tfrac{1}{2}(R - A)^2\right]\right)^{-1} = 1$$

This is the best possible unbiased estimator in a mean-squared-error sense.

This idea can be generalized to the multivariable problem through the use of Fisher's information matrix. In this case a vector $\boldsymbol{A}$ is to be estimated after observing another vector $\boldsymbol{R}$ by use of an estimating function $\hat{\boldsymbol{a}}(\boldsymbol{R})$. The elements of Fisher's information matrix, J, can be defined in either of two equivalent ways:

$$J_{i,j} = \text{E}\left[\frac{\partial \ln \text{prob}_{R|A}(R|A)}{\partial A_i}\frac{\partial \ln \text{prob}_{R|A}(R|A)}{\partial A_j}\right]$$

or

$$J_{i,j} = -\text{E}\left[\frac{\partial^2 \ln \text{prob}_{R|A}(R|A)}{\partial A_i\,\partial A_j}\right]$$

To see how this works, consider the estimation error of the ith component of A. It has a bias error of $\beta_i(\boldsymbol{A}) = \text{E}[\hat{a}_i(\boldsymbol{R}) - A_i]$ and a mean squared error of $\epsilon_i = \text{E}[(\hat{a}_i(\boldsymbol{R}) - A_i)^2]$. It is convenient to define the vector $\boldsymbol{b}(i) = (\partial/\partial A_j)\text{E}[\hat{a}_i(\boldsymbol{R})]$. (Of course, if the estimator is unbiased, then $\boldsymbol{b}(i)$ has a 1 in the ith position and zeros elsewhere.) Then

$$\epsilon_i \geq \boldsymbol{b}(i)^T J^{-1}\boldsymbol{b}(i)$$

It is important not to confuse the concept of efficiency with optimality. Arguments that an estimator is optimum must be based on game theory or decision theory. This mistake is tempting partly because it seems intuitively that an unbiased estimator should be better than a biased estimator. However, this is not necessarily true. The following problem illustrates the difficulty.

Suppose one needs to estimate the variance of a zero mean Gaussian variable.

$$\ln \text{prob}_{R|A}(R|A) = -\tfrac{1}{2}\ln 2\pi - \tfrac{1}{2}\ln A - \frac{R^2}{2A}$$

The Cramér–Rao bound is

$$\frac{-1}{\mathrm{E}\left[\dfrac{\partial^2}{\partial A^2}\left(-\tfrac{1}{2}\ln 2\pi - \tfrac{1}{2}\ln A - \dfrac{R^2}{2A}\right)\right]} = 2A^2$$

and the maximum likelihood estimator is $\hat{A} = R^2$. The following observations follow easily:

1. $\hat{A} = R^2$ is an unbiased maximum likelihood estimator of A.
2. $\hat{A} = R^2$ is an efficient estimator of A, that is, it meets the Cramér–Rao bound with equality.
3. $\hat{A} = R^2$ is obviously not an optimal, or even a good, estimator for A. In fact, if one were to ignore R and simply make $\hat{A} = 0$ the average mean squared error would only be half of that given by the efficient estimator.

In fact, a better estimator would be $\hat{A} = R^2/3$. It would have a mean squared error of only one-third that of the efficient estimator.

In many cases, especially those involving small sample sizes, it may be worthwhile to investigate the possibilities of biased estimators. Little seems to be known about how a good bias function can be chosen.

FOURIER TRANSFORMS AND SPECTRUM ESTIMATION

Fourier transforms can be viewed as the solution to a least-squared-error estimation problem. This is useful for analysis of existence, convergence, uniqueness, and so on. However, when designing analysis procedures it is much easier to think of them as a frequency translation and filtering process. Let $x(n)$ denote a sequence of data values sampled at regular intervals at a rate of f_s samples per second. Then $x(n)e^{-i2\pi nf/f_s}$ is a time sequence that has the same structure as $x(n)$ except that it is shifted so the information that was at frequency f is now at 0 Hz. The spectral coefficient for frequency f is now found by low-pass filtering with a summation filter function to get

$$\sum_{n=0}^{M-1} x(n)e^{-i2\pi nf/f_s}$$

For a time period $T = M/f_s$, the complex exponentials at $f = f_s m/M$ are uncorrelated, where m is any integer. So the Fourier transform components are defined as

$$X(m) = \eta \sum_{n=0}^{M-1} x(n)e^{-i2\pi mn/M}$$

This formula is often referred to as the discrete Fourier transform (DFT). Since the original data sequence can be recovered by

$$x(n) = \frac{1}{\eta M}\sum_{m=0}^{M-1} X(m)e^{i2\pi mn/M}$$

the transformation has lost no information.

The choice of η is arbitrary and usually depends on the software package used. Most standard programming packages use $\eta = 1$, and the reader can safely assume this for the following discussion. However, a few packages [e.g., MathCad (MathSoft, Inc.)] use $\eta = 1/\sqrt{M}$, which makes the above formulas symmetrical.

It is important to keep track of the exact form of the Fourier transform used, because it determines the form of Parseval's theorem. With the above definitions, Parseval's theorem says that

$$\sum_{n=0}^{M-1} x^*(n)x(n) = \frac{1}{\eta^2 M}\sum_{m=0}^{M-1} X^*(m)X(m)$$

This is the key to computing the power spectral density.

The *power spectral density* $S_P(m)$ of a waveform is a function that, when integrated over a frequency band, will give the power of the waveform in that band. The equations must be calibrated in order to make the integral over the total frequency band come out right. Since the frequency resolution of the analysis is f_s/M, the approximations to Riemann integrals look like

$$\text{power} = \frac{1}{M}\sum_{n=0}^{M-1} x^*(n)x(n) = \sum_{m=0}^{M-1} S_P(m)\frac{f_s}{M}$$

so Parseval's theorem gives

$$S_P(m) = \frac{1}{\eta^2 f_s M} X^*(m)X(m)$$

This gives a procedure for stationary waveforms. However, if one needs to analyze impulsive functions a different line of thought is necessary. An *impulse* will be defined here as a function that takes on nonzero values only within the interval $0 \le n < M$. In this case, power is not an interesting quantity and the energy in the waveform becomes important. The energy spectral density is found by first noting the energy in the waveform is

$$\text{energy} = \sum_{n=0}^{M-1} x^*(n)x(n)\frac{1}{f_s} = \sum_{m=0}^{M-1} S_E(m)\frac{f_s}{M}$$

In this case, Parseval's theorem gives

$$S_E(m) = \frac{1}{\eta^2 f_s^2} X^*(m)X(m)$$

In either case, when the results are plotted, the usual procedure is to plot the spectral density versus frequency on a decibel scale. If the original level of E[x^*x] was specified in decibels relative to a reference level, the spectral data should be plotted in decibels per hertz relative to the reference level. The spectrum should not be labeled as "per $Hz^{1/2}$" unless the author really intends that the function is to be integrated with respect to the square root of the frequency. This mistake is made by a remarkable number of authors.

When the waveform contains pure tonals (defined as signals whose bandwidth is less than the analysis resolution), special problems arise. A pure sinusoid would have an infinite power spectral density, and be properly modeled as a Dirac delta function in frequency. This cannot be sensibly plotted on the same scale as power that is distributed over an identifiable frequency band. In this case, the total power in the tonal should be estimated. Then the peak should be deleted from the plot, and replaced by a line indicating the sinusoidal power. For example, suppose the spectral density in the neighborhood of 60 Hz is 150 dB/Hz, while the indicated power spectral density for the 60 Hz bin is 160 dB/Hz. If the frequency resolution from the Fourier transform were 1/50 Hz, this would mean that the power in the line was 160 dB − 17 dB = 143 dB (since 10 log 1/50 = −17). When the data are reported, the plot should show a smooth spectrum at 150 dB/Hz through the 60 Hz region and a vertical line rising to a level of 143 dB.

The above formulas are usually considered to be a bad way to estimate a spectrum, because of sidelobe leakage. Therefore a window function, $w(n)$, is usually used. To see the effect, it is easiest to think of the equivalent low-pass filter.We can write the Fourier transform as a convolution filter:

$$X(f,n) = \sum_{L=0}^{M-1} w(L)x(n-L)e^{-i2\pi(n-L)f/f_s}$$

Then the Fourier transformation consists in sampling this function at regular intervals. The intervals are not necessarily simply related to the length of the Fourier transform.

In the first case, $w(n)$ was a *boxcar* filter. That is, $w(n) = 1$ if $0 \le n < M$, and $w(n) = 0$ otherwise. Many good window functions are known.

When analyzing a window, it helps to compare it with the boxcar window. Relative to a boxcar window, the more popular window functions widen the main-lobe frequency response, reducing the frequency resolution, in order to lower the sidelobes. A window is usually judged by two criteria: How much does it broaden the main lobe, and how much does it lower the sidelobes? The Dolph–Chebyshev window has the lowest worst-case sidelobe for a given main-lobe width. Although this window is rarely used, it is a quick way to see what can be done. The window shape, in the frequency domain, is controlled by a parameter β. For an M-point window,

$$W(f) = T_{M-1}\left(\frac{\cos(\pi f/f_s)}{\cos(\pi\beta/M)}\right)$$

where T_{M-1} is a Chebyshev polynomial of order $M - 1$. (This works because the Chebyshev polynomials are themselves solutions of a minimax problem.) The first zero of a boxcar window of the same length would be at f_s/M. The first zero of a Dolph–Chebyshev window is at approximately $\beta f_s/M$. If the window width were measured to the points 3 dB or 6 dB down, the width would be about $\sqrt{\beta}$ times the width of the boxcar window of the same length. The sidelobes would be about $27.3\beta - 6$ dB down from the main lobe. Thus, if a given level of sidelobe rejection is specified, one can immediately see how narrow a main lobe is possible (i.e., what the best possible frequency resolution is). Or if the frequency resolution of the window is specified, one can see how much sidelobe rejection is possible.

The Dolph–Chebyshev window is rarely used, for two reasons. First, although the worst sidelobes are well down, all of the other sidelobes are equally high. They do not taper off. Second, the endpoints of the window are often quite high. These problems are sometimes alleviated by convolving the Dolph–Chebyshev window with a short binomial window. The binomial window has a very broad main lobe but no sidelobes at all. When the two windows are convolved in the time domain, they are multiplied in the frequency domain. The time domain convolution smooths out the spikes at the end of the window, while the frequency domain multiplication reduces the distant sidelobes.

The Kaiser–Bessel window is a more popular choice. It is obtained by sampling the function

$$w(t) = \begin{cases} \dfrac{1}{T} I_0\left(\pi\beta\sqrt{4\left(\dfrac{t}{T}\right)\left(1-\dfrac{t}{T}\right)}\right), & 0 \le t \le T \\ 0 & \text{otherwise} \end{cases}$$

where T is the time duration of the window and I_0 is a Bessel function (1) computed by

$$I_0(z) = 1 + \frac{z^2/4}{(1!)^2} + \frac{(z^2/4)^2}{(2!)^2} + \frac{(z^2/4)^3}{(3!)^2} + \cdots$$

The first (and worst) sidelobe of the filter is approximately $27.3\beta - 20 \log \beta - 2.5$ dB down from the peak response. The first null occurs at approximately $\beta(1 + 0.333/\beta^2)/T$ Hz. If a boxcar filter is used, the first null occurs at a frequency of $1/T$ Hz.

The actual process of computing the Fourier transforms is usually done using an algorithm called the *fast Fourier transform* (FFT). It provides a much faster computation with less rounding error than one would get from a DFT. For present purposes it is important only to understand that the FFT has no theoretical significance. It is simply a quick way to compute the same result that could otherwise be obtained with a DFT.

To be efficient the FFT requires that M be a highly composite number, usually a power of 2. Since the length of the available data string is unlikely to be a power of 2, this might seem to be a problem. However, the problem is easily solved by padding the data with zeros to fill out the input vector. The effect of this is to overresolve the spectrum. This turns out to be very beneficial if the spectrum contains sharp features that might otherwise be difficult to resolve. Of course, one could get the same effect by interpolation, but it would be much more difficult. This works out so well that often the analyst may make M much longer than the size of the data set in order to get a smooth spectrum that is easy to interpret.

The use of a window and zero padding requires a modification of the equation calibration. Parseval's theorem again provides guidance.

STATIONARITY ISSUES

Most signal-processing theory assumes time-stationary processes, at least for the noise. All physical systems are ultimately not time-stationary. It is often important to arrive at some clear opinion about how nearly stationary the data are. A time sequence is time-stationary if, for any set of values $x(n)$, $x(n + 1)$, . . ., $x(n + M - 1)$ and any function $a(x(n), x(n + 1), \ldots, x(n + M - 1))$ of those points, the average of a, or $\mathrm{E}[a(x(n), x(n + 1), \ldots, x(n + M - 1))]$, is not a function of n. In other words, absolute time has no meaning for the sequence. This condition is usually impossible to test and stronger than is needed for most analyses. Therefore, it is much more common to assume that the data are *wide-sense stationary,* or *second-order stationary.* This means simply that all of the first and second moments of the data stream are independent of time. In this case it is possible to identify an autocorrelation function, $A(n) = \mathrm{E}[x(t)x(t + n)]$, where $A(n)$ is independent of t.

When the data sequence is not second-order stationary, it is often possible to choose the Fourier transform lengths so that the spectrum changes slowly relative to the Fourier transform interval. Then sequential spectra can be compared and peaks in the spectrum followed as they change in time. When several spectra are plotted, one above another, on a single display, these peaks often follow characteristic paths down the display. Since the peaks trace out a visible line, narrowband components of a spectrum are often referred to as *lines.* Often a great deal of study and experience is required to interpret these lines. For example, lines at frequencies that are harmonics of power frequencies (50 Hz or 60 Hz, depending on the country) are apt to be a symptom of instrumentation problems.

Assuming that the data sequence is stationary, two assumptions are usually made that are only approximately true. The first is that Fourier coefficients corresponding to different frequencies are uncorrelated, that is, $\mathrm{E}[X^*(m_1)X(m_2)] = 0$ for all $m_1 \neq m_2$. The second is that the real and imaginary parts of the Fourier coefficients are uncorrelated and of equal variance, that is, $\mathrm{E}[X_r(m)X_i(m)] = 0$ and $\mathrm{E}[X_r^2(m)] = \mathrm{E}[X_i^2(m)]$ for all m. Another way to state the condition is that $\mathrm{E}[X^2(m)] = 0$. As will be seen below, this second condition means that for Gaussian data the probability density function of $X(m)$ depends only on the magnitude of $X(m)$. Equal-probability contours of $X(m)$ then are circles in the complex plane, so the variables are called *circular.*

The Fourier coefficients from different frequency bins usually have a small but nonzero correlation because of sidelobe leakage in the window function. The amount of correlation is controlled by the choice of window and the extent to which the data have been whitened prior to study.

The circularity issue has not been fully explored. In order to do so, it is probably useful to define a *circularity anomaly,*

$$\alpha_c(m) = -\eta^2 \sum_{n-0}^{M-1} A(n) \sin \frac{2\pi nm}{M}$$

that is, the sine transform of the autocorrelation function. Then

$$\mathrm{E}[X^2(m)] = \frac{2\alpha_c(m)}{\sin(2\pi m/M)} e^{i2\pi m/M}$$

The phase angle is independent of the spectrum. Under some circumstances, it is possible that this phase angle might be used as a test of stationarity. However, no such test procedures have been worked out.

If one suspects that circularity might not hold, it may be a good precaution to multiply each Fourier coefficient by $e^{-i2\pi m/M}$. This will have the effect of decorrelating the real and imaginary components. However, it will also maximize the difference in their magnitudes.

Usually $\alpha_c(m)$ is small enough to ignore safely. Therefore, the following sections will assume that the Fourier coefficients are circular. However, it is not clear when the rare exceptions occur. They are associated with steep changes in the spectrum. It is possible, in situations where a narrowband signal is on a steep spectral slope, that the signal will be more detectable on looking only at one part of the Fourier coefficients. This is not commonly done.

BANDWIDTH AND TIME–BANDWIDTH PRODUCTS

The entire frequency range available for analysis is usually wider than the signals of interest. Often, the signal energy is confined between two frequencies, f_1 and f_2. Then it is convenient to define a frequency bandwidth $W = f_2 - f_1$. Recalling that the integration time of the Fourier transform is $T = M/f_s$, the frequency resolution of the analysis is $1/T = f_s/M$, so there are $K = WT$ Fourier transform bins that contain the signal. K is referred to as the *time–bandwidth product* of the signal. It is often important to know what the duration and the bandwidth of the signal are.

Curiously, there is no generally agreed-upon way to define the bandwidth of a signal. Indeed, a similar problem may exist in defining the duration of a signal. Sometimes the nature of the problem may dictate a definition that is appropriate only to that problem. For example, when considering the uncertainty principle, the bandwidth and time duration of a signal are defined by

$$W^2 = \int_{-\infty}^{\infty} f^2 S(f)\, df \quad \text{and} \quad T^2 = \int_{-\infty}^{\infty} t^2 |x(t)|^2\, dt$$

when the signal is normalized so that

$$1 = \int_{-\infty}^{\infty} |x(t)|^2\, dt = \int_{-\infty}^{\infty} S(f)\, df$$

This leads to the uncertainty principle (2)

$$WT \geq 1/4\pi$$

However, these definitions for W and T seem not to be used for any other purpose than proving the uncertainty principle.

More frequently, the edges of the frequency band are defined as the points at which the spectrum is 3 dB down from the peak. This is especially appropriate when working with Butterworth filters. In this case the 3 dB down frequency is known as the *corner frequency.* This identity of the corner frequency and the 3 dB down frequency is not true for most other filter types, but they drop off fast enough that the error is small.

Square law detection theory can provide another useful definition of bandwidth. As will be seen below, in this case

the detectability of a random signal increases as the square root of the time–bandwidth product and the average signal-to-noise ratio across the frequency band. Thus, for maximum detectability one would want to choose f_1 and f_2 to maximize

$$\frac{1}{\sqrt{f_2 - f_1}} \int_{f_1}^{f_2} \frac{S(f)}{N(f)}\, df$$

This prescription leads to

$$\frac{S(f_1)}{N(f_1)} = \frac{S(f_2)}{N(f_2)} = \frac{1}{2} \frac{1}{f_2 - f_1} \int_{f_1}^{f_2} \frac{S(f)}{N(f)}\, df$$

In other words, the edges of the frequency band should be chosen so that the signal-to-noise ratio at each edge is 3 dB below the average signal-to-noise ratio over the band.

SINGLE-WAVEFORM TESTING AND SQUARE LAW DETECTORS

One of the most instructive and fundamental detection problems involves a waveform that may or may not be present in Gaussian noise. The two possibilities are denoted as H_0 (no signal present) and H_1 (one signal present). The noise is assumed to be from a time-stationary random process, and is known only by its spectrum. This can be denoted by $\nu(m) = \mathrm{E}[X^*(m)X(m)|H_0]$.

The problem usually takes one of two different forms, depending on what *a priori* information about the signal is assumed. In the first case the signal is assumed to be known exactly. This is appropriate for study of active sonar or radar. The signal is then a waveform that takes on nonzero values only for a limited time. The Fourier transform of the signal will be denoted by $S(m)$, with the signal power designated as $\sigma(m) = S(m)^*S(m)$. Several lines of thought lead to use of a correlator or a convolution operator. This detector may be implemented in either the time domain or the frequency domain. However, it is easier to analyze in the frequency domain. The detector uses a linear filter described by $H(m)$ and is equivalent to forming a test statistic

$$U = \sum \mathrm{Re}[H(m)X(m)]$$

where the sum is taken over the $K = WT$ frequency bins that contain significant signal energy. The issue is the statistics of U,

$$\mathrm{E}[U|H_0] = 0 \quad \text{and} \quad \mathrm{E}[U|H_1] = \sum \mathrm{Re}[H(m)S(m)]$$

The only other quantity of interest is the variance,

$$\mathrm{E}[U^2|H_0] = \frac{1}{2} \sum H^*(m)H(m)\nu(m)$$

It can be easily shown that the optimum choice of filter function is $H(M) = S^*(m)/\nu(m)$. However, nothing is lost by scaling $H(M)$ so that $\mathrm{E}[U^2|H_0] = 1$. It also helps to designate the average signal-to-noise ratio over the band as $\langle \sigma/\nu \rangle_\mathrm{f}$. This is a different type of average than used above. It enables one to separate the effect of averaging, or time–bandwidth product, from the effect of the energy ratios. If this is done, then

$$\mathrm{E}[U|H_1] = \sqrt{2K\langle \sigma/\nu \rangle_\mathrm{f}}$$

If the noise is white (i.e., the spectrum is flat over the band) the above detector is called a *matched filter*. In this case, the signal-to-noise ratio reduces to the ratio of the total signal energy to the noise power spectral density.

The detection process consists in selecting a threshold value, U_th, and comparing it with the filter output. The false-alarm rate, or probability of false alarm, can be found from the usual Gaussian distribution

$$P_\mathrm{F} = Q(U_\mathrm{th})$$

where $Q(\ )$ is characterized by Eq. (26.2.3) of Ref. 1. Possibly useful values are

$$Q(3.72) = 10^{-4}$$
$$Q(4.75) = 10^{-6}$$
$$Q(5.61) = 10^{-8}$$
$$Q(6.36) = 10^{-10}$$

The probability of detection is at least 50% if $2K\langle \sigma/\nu \rangle_\mathrm{f} \geq U_\mathrm{th}^2$. If $a = \log U_\mathrm{th}$, then the signal excess can be defined as

$$\mathrm{SE_c} = 10 \log \langle \sigma/\nu \rangle_\mathrm{f} + 10 \log K + 3 - 2a$$

Two points may surprise the knowledgeable reader. First, SE is not a simple function of integrated signal power and integrated noise power. The averaging is done only after the ratio has been taken. The second is the 10 log dependence on K. This is an important difference between detection of a known signal and of an unknown signal (discussed below).

If the noise is white,

$$\mathrm{SE_c} = 10 \log(\text{total signal power}) - 10 \log(\text{noise power spectral density}) + 3 - 2a$$

The quantity $3 - 2a$ is sometimes referred to as the "recognition differential." However, this term is used in a confusing variety of different ways, so one is usually better off to avoid using it altogether.

The second important variant on this problem is the unknown signal. In this case, the signal is assumed to be a time-stationary Gaussian signal known only by its spectrum, which will be denoted by $\sigma(m) = \mathrm{E}[X^*(m)X(m)|H_1] - \mathrm{E}[X^*(m)X(m)|H_0]$. Several arguments lead to a square law detector,

$$V = \sum X^*(m)X(m)H(m)$$

Assuming the signal spectrum is accurately known, the best choice of the frequency weights is

$$H(m) = \frac{\sigma(m)}{\nu(m)[\nu(m) + \sigma(m)]}$$

This is approximated by the Eckart filter, $H(m) = \sigma(m)/\nu^2(m)$ (3). However, for various reasons, including difficulty in knowing the signal spectrum, it is often more practical to use

a noise-whitening filter followed by a band-pass filter,

$$H(m) = \frac{1}{K\nu(m)}$$

This means that

$$\mathrm{E}[V|H_0] = 1 \quad \text{and} \quad \mathrm{E}[V|H_1] = 1 + \left\langle \frac{\sigma}{\nu} \right\rangle_{\mathrm{f}}$$

At this point it is tempting to use the central limit theorem (CLT) to argue that the distribution of V is Gaussian and the detection statistics can be estimated as above. However, the CLT works poorly on the tails of the distribution, and fortunately this approximation is not necessary. In fact, V has the form of a chi-square variable, and the distribution of V is the gamma distribution. If $\Gamma(\ ,\ )$ denotes the incomplete gamma function, then the probability density function of V is

$$G(K, KV) = \frac{\Gamma(K, KV)}{\Gamma(K)}$$

This equation differs slightly from Eq. (26.4.19) of Ref. 1 because of different normalization and because K is only half the number of degrees of freedom.

Then if V_{th} denotes the threshold,

$$P_{\mathrm{F}} = G(K, KV_{\mathrm{th}})$$

Exact evaluation of this equation is cumbersome. However, it is easily approximated by

$$\frac{(KV)^{K-1}}{(K-1)!}e^{-KV} < G(K, KV) < \frac{1}{1 - \dfrac{K-1}{KV}} \frac{(KV)^{K-1}}{(K-1)!} e^{-KV}$$

In fact, $G(K, KV)$ stays much closer to the upper bound.

This gives an easy way to estimate P_{F} for various values of K and V_{th}. However, it is also useful to be able to turn the problem around and find the required V_{th} for a given P_{F} and K. In most cases this problem cannot be solved in closed form. It has been found empirically that, for realistic false-alarm rates, a good approximating equation is

$$10\log(V_{\mathrm{th}} - 1) = a - 5\log K + 10\log\left(1 + \frac{b}{\sqrt{K}}\right)$$

For this purpose, the following table may be adequate:

$$\begin{aligned} P_F &= 10^{-4}, \quad a = 5.705, \quad b = 1.2 \\ P_F &= 10^{-6}, \quad a = 6.77, \quad b = 1.65 \\ P_F &= 10^{-8}, \quad a = 7.49, \quad b = 2 \\ P_F &= 10^{-10}, \quad a = 8.03, \quad b = 2.4 \end{aligned}$$

As above, one can define a signal excess equation as

$$\mathrm{SE_s} = 10\log\left\langle \frac{\sigma}{\nu} \right\rangle_{\mathrm{f}} + 5\log WT - a - 10\log\left(1 + \frac{b}{\sqrt{K}}\right)$$

The last term may be interpreted as the error that would have resulted if a Gaussian distribution assumption had been made for V.

For detection of tonals, this equation takes a somewhat different form because a different definition of σ is used. When investigating tonals, or nearly pure sinusoids, instead of specifying the power spectral density of the signal, only the total signal power is specified. The spectrum of the signal is then treated as a Dirac delta function times that signal power. The key assumption is that the total width of the signal is less than the width of a Fourier bin. Then the apparent signal power spectral density depends on the bin width, which is now W. With this new different definition of σ,

$$\mathrm{SE_s} = 10\log\left\langle \frac{\sigma}{\nu} \right\rangle + 5\log T - 5\log W - a - 10\log\left(1 + \frac{b}{\sqrt{K}}\right)$$

As suggested above, it is often difficult to obtain good *a priori* information about the signal. However, similar problems occur for the noise. If the absolute level of the noise is unknown, then it must be measured before thresholds can be set. When attempting to detect narrowband signals, this process is referred to as *noise spectrum equalization,* or NSE. In its simplest form, NSE can be analyzed as follows.

When attempting to detect a narrowband signal, a common approach is to plot the power spectral density and look for sharp narrow peaks. The eye can then easily identify the average level of the noise and judge whether a particular spike is significantly higher than that average level. To quantify this, assume that L bin levels around the signal bin are averaged. If K is the time–bandwidth product for each bin, then the average noise level is being estimated with a time–bandwidth product of KL. What the eye actually sees, especially if the spectrum is plotted on a decibel scale, is the ratio of the estimated power in the signal bin to the estimated power in the surrounding noise bins. This is a ratio of two powerlike variables. This ratio will have an F distribution. Let $\rho = \sigma(m)/\nu(m)$ denote the signal-to-noise ratio in the signal bin, and assume that the noise spectrum is flat over the L frequency bins around the signal. The probability density function of the ratio is

$$\mathrm{prob}_Z(z) = \frac{(K + KL - 1)!(KL)^{KL}}{(K-1)!(KL-1)!} \times \left(\frac{K}{1+\rho}\right)^K \frac{z^{K-1}}{\left(\dfrac{zK}{1+\rho} + KL\right)^{K+KL}}$$

The cumulative probability function can be written as

$$\mathrm{Prob}_Z(z) = \frac{B_{z/z+L(1+\rho)}(K, KL)}{B(K, KL)} = I_{z/(z+L(1+\rho))}(K, KL)$$

where $B_{\xi}(K, KL)$ is the incomplete beta function (1). To compute the false-alarm rate, simply set $\rho = 0$.

This type of detector can work very well when the time–bandwidth products are large. However, for small time–bandwidth products the price of having to estimate the normalization factor is severe. The extreme case occurs when $K = L = 1$. In this case, the false-alarm rate, for a threshold value of th, is 1/(th + 1). This means that if one wanted a false-alarm rate of 10^{-4}, it would take a signal-to-noise ratio of 40 dB to give a 50% probability of detection. As the time–bandwidth product of the detector increases, the detection

performance improves rapidly, approaching the performance of a square law detector as L becomes large. This pattern—the importance of large time–bandwidth product when a normalization factor is estimated from the data—will reappear below in the discussion of two-channel detectors.

The general problem of detection of Gaussian signals in Gaussian noise, or even of sinusoids in Gaussian noise, is far from solved. For example, effects of sidelobe leakage in the Fourier transforms have been ignored. More importantly, if the noise power spectral density is significantly far from white, the resulting detection statistic is a sum of unequal chi-square variables. The probability distribution of such a variable is so cumbersome as to be nearly useless. A good method for approximating it is needed.

TWO-CHANNEL DETECTION

Detection or estimation of a signal that is believed to be common to two different waveforms may be done in several different ways, depending on the *a priori* information available and the type of information to be extracted. In the following discussion, $X(n)$ and $Y(n)$ are the two complex data sequences. Usually they are Fourier coefficients from successive transform intervals. In the following equations, $\langle\ \rangle$ denotes the average over K data samples. It will also be initially assumed that the signal-to-noise ratio in both sequences is the same. That is, $\mathrm{E}[X^*X|H_0] = \mathrm{E}[Y^*Y|H_0] = \nu$ and $\mathrm{E}[X^*X|H_1] = \mathrm{E}[Y^*Y|H_1] = \nu + \sigma$. The noise will be assumed to be Gaussian, uncorrelated between the two sequences, and independent of the signal.

There are four principal functions from which one may choose:

$$u_1 = \langle X^*X\rangle \overset{?}{>} T_1\nu \qquad \text{square law}$$

$$u_2 = \mathrm{Re}\langle X^*Y)\rangle \overset{?}{>} T_2\nu \qquad \text{correlator}$$

$$u_3 = \frac{\mathrm{Re}\langle X^*Y\rangle}{\sqrt{\langle X^*X\rangle\langle Y^*Y\rangle}} \overset{?}{>} T_3 \qquad \text{correlation coefficient}$$

$$u_4 = \frac{|\langle X^*Y\rangle|^2}{\langle X^*X\rangle\langle Y^*Y\rangle} \overset{?}{>} T_4 \qquad \text{coherence}$$

The first function is included as a reference. It is the simple square law detector, analyzed previously. It forms a baseline for judgement of the other detectors, since it simply uses one of the two sequences. The comparison gives an indication of the value of having two sequences instead of one. An important case that is not considered here is $\langle (X + Y)^*(X + Y)\rangle$. This is because it does not really constitute a separate case. It is simply the square law detector with a 3 dB increase in signal-to-noise ratio.

In each case, the quantity u is compared with a threshold. (In the first two cases it is necessary to know ν in order to set the threshold.) It is important to know how the false-alarm rate will be determined by the threshold. However, this is only part of the story, since the probability of detection is also important. In each case, it is possible to associate a signal-to-noise ratio with the threshold that will produce approximately a 50% probability of detection. The critical signal-to-noise ratios are

$$1 + \left(\frac{\sigma}{\nu}\right)_1 = T_1$$

$$\left(\frac{\sigma}{\nu}\right)_2 = T_2$$

$$\frac{(\sigma/\nu)_3}{(\sigma/\nu)_3 + 1} = T_3, \quad \text{or} \quad \left(\frac{\sigma}{\nu}\right)_3 = \frac{T_3}{1 - T_3}$$

$$\frac{(\sigma/\nu)_4^2}{[(\sigma/\nu)_4 + 1]^2} = T_4, \quad \text{or} \quad \left(\frac{\sigma}{\nu}\right)_4 = \frac{\sqrt{T_4}}{1 - \sqrt{T_4}}$$

These four signal-to-noise ratios will be called threshold signal-to-noise ratios. However, they are actually bin signal-to-noise ratios. To reconcile the following discussion with standard detection equations, one would have to at least correct for the bandwidth of the frequency bins. Further, due to asymmetries in the distribution functions, the threshold signal-to-noise ratios do not correspond precisely with a 50% probability of detection. The errors from this asymmetry are usually very small.

The false-alarm rates are, of course, determined by the threshold values. For the correlator the false-alarm rate is

$$P_F(T_2) = \frac{1}{2^{2K-1}(K-1)!}\sum_{n=1}^{K-1}\frac{(2K-n-2)!2^n}{n!(K-n-1)!}\Gamma(n+1, 2KT_2)$$

For the correlation coefficient detector the false-alarm rate is (4)

$$P_F(T_3) = \frac{(K-1)!}{\sqrt{\pi}\left(\dfrac{2K-3}{2}\right)!}\int_{T_3}^{1}(1-t^2)^{(2\kappa-3)/2}dt$$

Although this formula can be integrated in closed form, the solution is very cumbersome. However, it lends itself to numerical integration. For the coherence detector the false-alarm rate is (5)

$$P_F(T_4) = (1 - T_4)^{K-1}$$

These formulas were used to compute Figs. 3, 4, 5, and 6. In each case the plot was designed to answer the question, "If

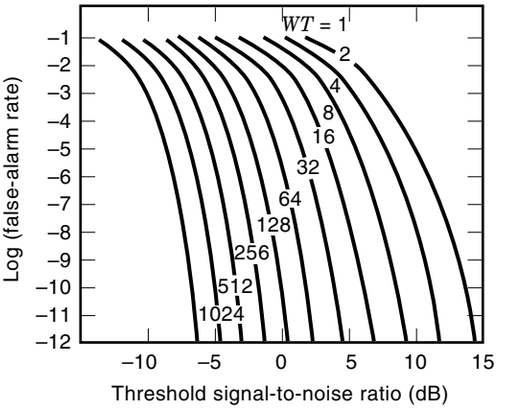


Figure 3. Log of false-alarm rate versus threshold signal-to-noise ratio for a square law detector. The curves are separated by about 1.5 dB for large WT products, but performance deteriorates more rapidly for small WT products.

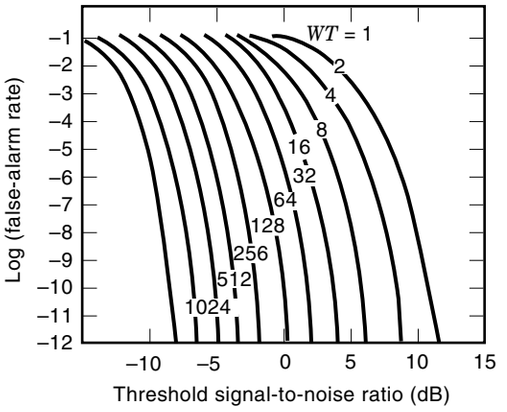


Figure 4. Log of false-alarm rate versus threshold signal-to-noise ratio for a correlator. The curves approximate those in Fig. 3 with a doubling of the *WT* product.

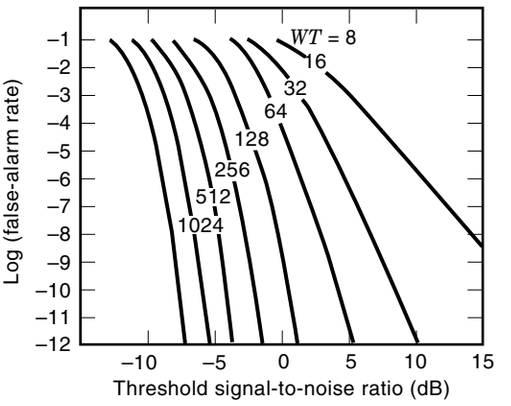


Figure 6. Log of false-alarm rate versus threshold signal-to-noise ratio for a coherence. Again, the performance deteriorates rapidly for small *WT* products.

the detector is set up to detect a signal at a given signal-to-noise ratio, what will the false-alarm rate of the detector be?" In each case, a threshold signal-to-noise ratio was chosen and the corresponding threshold value calculated. Then the probability of a noise-only false alarm was calculated and plotted. This was done for several values of $K = WT$, the time–bandwidth product. Since in general low false-alarm rates are necessary, the curves are mainly useful for the region of $P_{fa} < 10^{-4}$. The following discussion will address only this region.

In Fig. 3, for a given false-alarm probability, the curves are separated by about 1.5 dB in the large-*WT* cases. This agrees with the general rule that the integration gain of a detector is 5 log *WT*. However, for small *WT* values the separation increases to about 2.5 dB. This is because the 5 log *WT* is based on application of the CLT, which breaks down for small *WT*. In some cases this can lead to a difference of 3 or 4 dB in minimum detectable signal.

The curves in Fig. 4 nearly overlie those in Fig. 3, with a shift in *WT*. For example, the curve for $WT = 1$ in Fig. 4 closely overlays the curve for $WT = 2$ in Fig. 3. In other words, the advantage in having a second waveform and using a correlator over using a square law detector on one waveform is a factor of 2 in the integration time needed.

For large *WT*, the curves in Figs. 4, 5, and 6 nearly coincide. In other words, for large *WT*, all three of these techniques give nearly the same performance. Selection among these formulas can be made on the basis of considerations other than detection performance, such as ease of implementation.

For small *WT*, the performance of the normalized detectors deteriorates so rapidly that curves for *WT* less than 8 were not even plotted. This is consistent with the previous observations about normalized spectra. Normalized detection formulas work well only with large sample sizes. For *WT* less than 128, the normalized formulas do not work as well as a square law detector using only one sequence.

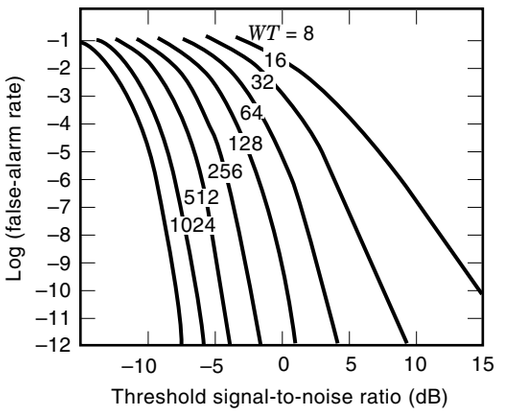


Figure 5. Log of false-alarm rate versus threshold signal-to-noise ratio for a correlation coefficient. The curves approximate those in Fig. 4 for large *WT* products but deteriorate rapidly for small *WT* products. This illustrates the difficulty of estimating a normalization factor from local data unless the *WT* factor is very large.

GAUSSIAN DISTRIBUTIONS

Most theoretical work on signal-processing problems assumes a Gaussian noise distribution. This assumption rests on two points of practical experience. First, much of the noise encountered in operating systems is approximately Gaussian. Second, data-processing systems based on Gaussian noise assumptions have a good track record in a wide range of problems. (This record is partly due to the coincidence between solutions based on Gaussian noise theory and solutions based on least-squares theory, as will be seen below.)

From a theoretical viewpoint the key feature of the Gaussian distribution is that a sum of Gaussian variables has a Gaussian distribution. (Other distributions with this property, called *alpha stability,* exist. One example is the Cauchy distribution. However, their role has yet to be established.) The importance of this fact is difficult to exaggerate. It means, among other things, that when Gaussian noise is passed through a linear filter, the output will still be Gaussian. (Unfortunately, little is known about what happens to the distribution of non-Gaussian noise when it is filtered. It is often said that because of the CLT the output of the filter can be assumed to be Gaussian. However, many important

counterexamples are known, e.g., AM radio.) Partly because of this, the Gaussian distribution is almost the only distribution for which the extension to multiple variables or complex variables is understood.

The CLT is often cited as another reason to assume a Gaussian distribution. The CLT says that if a variable y is an average of a large number of variables, $x_1, x_2, \ldots, x_N$, then the distribution of y is approximately Gaussian and that this approximation improves as N increases, that is, y is asymptotically Gaussian. The necessary and sufficient conditions for this theorem are not known. However, several sets of sufficient conditions are known, and they seem to cover most reasonable situations. For example, one set of sufficient conditions is that the x_i's are independent and have equal variance.

The reader should, however, use some caution in invoking the CLT. It is an asymptotic result that is only approximately true for finite N. Further, the accuracy of this approximation is often very difficult to test. It tends to come into play fairly quickly in the central portions of the distribution, so when the experimental distribution is plotted the data look deceptively close to a Gaussian curve. However, detection and estimation problems tend to depend on the tails of the distribution, which may be very slow to converge to a Gaussian limit and cause large errors that are poorly understood. The investigator should always be alert for the possibility that a Gaussian distribution is not appropriate and should therefore consider alternatives.

Let $\boldsymbol{x}$ denote a column vector of real variables, $\boldsymbol{x}^T = [x_1 \; x_2 \; \cdots \; x_n]$, and let $C = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}]$ denote the covariance matrix of $\boldsymbol{x}$. Then the statement that $\boldsymbol{x}$ is Gaussian means that

$$\mathrm{prob}_{\boldsymbol{x}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n |C|}} e^{-\frac{1}{2}\boldsymbol{x}^{\mathrm{T}} C^{-1} \boldsymbol{x}}$$

If the variables are complex, it is possible to define two important square matrices, $\Gamma = \langle \boldsymbol{x}\boldsymbol{x}^{\mathrm{H}} \rangle$ and $C = \langle \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}} \rangle$. It is customary to assume that $C = 0$, which is the circularity assumption. This custom will be adopted later. In this case, it is convenient to define the accent vector

$$\acute{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}^* \end{bmatrix}$$

The moment matrices and their inverses take the form

$$\mathrm{E}[\acute{\boldsymbol{x}}\acute{\boldsymbol{x}}^{\mathrm{H}}] = E = \begin{bmatrix} \Gamma & C \\ C^* & \Gamma^* \end{bmatrix} = \begin{bmatrix} A & B \\ B^* & A^* \end{bmatrix}^{-1}$$

The probability density function of $\acute{\boldsymbol{x}}$ is

$$\mathrm{prob}(\boldsymbol{x}) = \frac{1}{\pi^n \sqrt{|E|}} e^{-\boldsymbol{x}^{\mathrm{H}} A \boldsymbol{x} - (\boldsymbol{x}^{\mathrm{H}} B \boldsymbol{x}^* + \boldsymbol{x}^{\mathrm{T}} B^* \boldsymbol{x})/2}$$

For a single complex Gaussian variable x, this simplifies. Let $\gamma = \mathrm{E}[xx^*]$, let $c = \mathrm{E}[x^2]$, and let $\rho = c/\gamma$. Then

$$\mathrm{prob}(x) = \frac{1}{\pi\gamma\sqrt{1-\rho^*\rho}} \exp\left(\frac{-[x^* x - \frac{1}{2}(x^2\rho^* + z^{*2}\rho)]}{\gamma(1-\rho^*\rho)} \right)$$

Using the accent notation for the variables, the joint and conditional distributions take simple forms. If $\boldsymbol{x}$ and $\boldsymbol{y}$ are jointly Gaussian vectors of length n and m respectively, the total covariance matrix can be defined as

$$E_{\mathrm{total}} = \mathrm{E}\left[\begin{bmatrix} \acute{\boldsymbol{x}} \\ \acute{\boldsymbol{y}} \end{bmatrix} [\acute{\boldsymbol{x}}^{\mathrm{H}} \quad \acute{\boldsymbol{y}}^{\mathrm{H}}] \right] = \begin{bmatrix} E_{xx} & E_{xy} \\ E_{yx} & E_{yy} \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}^{-1}$$

Then the joint distribution of $\boldsymbol{x}$ and $\boldsymbol{y}$ is

$$\frac{1}{\pi^{n+m}\sqrt{|E_{\mathrm{total}}|}} \exp\left(-\frac{1}{2} [\acute{\boldsymbol{x}}^{\mathrm{H}} \quad \acute{\boldsymbol{y}}^{\mathrm{H}}] E_{\mathrm{total}}^{-1} \begin{bmatrix} \acute{\boldsymbol{x}} \\ \acute{\boldsymbol{y}} \end{bmatrix} \right)$$

while the conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{y}$ is

$$\mathrm{prob}(\boldsymbol{x}|\boldsymbol{y}) = \frac{\sqrt{|F_{11}|}}{\pi^n} e^{-\frac{1}{2}(\acute{\boldsymbol{x}} - E_{xy}E_{yy}^{-1}\acute{\boldsymbol{y}})^{\mathrm{H}} F_{11} (\acute{\boldsymbol{x}} - E_{xy}E_{yy}^{-1}\acute{\boldsymbol{y}})}$$

The moment generating function of a complex vector $\boldsymbol{s}$ is

$$\mathrm{mgf}(\boldsymbol{s}) \equiv \mathrm{E} e^{-\boldsymbol{s}^{\mathrm{H}}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{H}}\boldsymbol{s}} = e^{\frac{1}{2}\acute{\boldsymbol{s}}^{\mathrm{H}} E \acute{\boldsymbol{s}}} = e^{\boldsymbol{s}^{\mathrm{H}}\Gamma\boldsymbol{s} + (\boldsymbol{s}^{\mathrm{H}} C \boldsymbol{s}^* + \boldsymbol{s}^{\mathrm{T}} C^* \boldsymbol{s})/2}$$

For a single variable, this simplifies to

$$\mathrm{E} e^{-s^* x + x^* s} = e^{s^*\gamma + (s^{*2} c + s^2 c^*)/2}$$

Matching up coefficients for the fourth moments gives a little-known result,

$$\mathrm{E}[(x^* x)^2] = \gamma^2 (2 + \rho^* \rho)$$

In other words, the *kurtosis,* defined here as the ratio of the fourth moment to the square of the second moment, varies between 2 and 3 depending on the degree of circularity of the variable. For real variables, $\rho = 1$, so the kurtosis is 3. For circular Gaussian variables, the most commonly used complex distribution, $\rho = 0$, so the kurtosis is 2. (Some authors subtract 3 from the ratio in their definition of kurtosis, so that for real Gaussian variables the kurtosis is zero. For formulations that include complex variables this is not a simplification.)

LIKELIHOOD DETECTORS FOR GAUSSIAN NOISE

Assuming a known signal, $\boldsymbol{s}$, in Gaussian noise the likelihood ratio for a sample variable $\boldsymbol{x}$ is

$$\frac{\frac{1}{\pi^n\sqrt{|E|}} e^{-\frac{1}{2}(\acute{\boldsymbol{x}} - \acute{\boldsymbol{s}})^{\mathrm{H}} E^{-1} (\acute{\boldsymbol{x}} - \acute{\boldsymbol{s}})}}{\frac{1}{\pi^n\sqrt{|E|}} e^{-\frac{1}{2}\acute{\boldsymbol{x}}^{\mathrm{H}} E^{-1} \acute{\boldsymbol{x}}}}$$

Isolating the terms that depend on $\boldsymbol{x}$, the likelihood ratio depends only on the expression

$$\acute{\boldsymbol{x}}^{\mathrm{H}} E^{-1} \acute{\boldsymbol{s}}$$

This provides a justification for the correlation structure discussed above.

The Gaussian signal assumption leads to a more complicated structure. In its simplest form, the signal is modeled as a random complex amplitude times a signal model vector $\boldsymbol{v}$

that is normalized so that $\boldsymbol{v}^{\mathrm{H}}\boldsymbol{v} = n$. If we admit that the signal may be noncircular, the signal covariance matrix takes a rank-two form:

$$\begin{aligned} P &= \begin{bmatrix} \sigma\boldsymbol{v}\boldsymbol{v}^{\mathrm{H}} & c\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}} \\ c^*\boldsymbol{v}^*\boldsymbol{v}^{\mathrm{H}} & \sigma\boldsymbol{v}^*\boldsymbol{v}^{\mathrm{T}} \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{c}\boldsymbol{v} & \sqrt{c}\boldsymbol{v} \\ \sqrt{c^*}\boldsymbol{v}^* & -\sqrt{c^*}\boldsymbol{v}^* \end{bmatrix} \\ &\quad \begin{bmatrix} \dfrac{\sigma/\sqrt{c^*c}+1}{2} & 0 \\ 0 & \dfrac{\sigma/\sqrt{c^*c}-1}{2} \end{bmatrix} \begin{bmatrix} \sqrt{c^*}\boldsymbol{v}^{\mathrm{H}} & \sqrt{c}\boldsymbol{v}^{\mathrm{T}} \\ \sqrt{c^*}\boldsymbol{v}^{\mathrm{H}} & -\sqrt{c}\boldsymbol{v}^{\mathrm{T}} \end{bmatrix} \end{aligned}$$

This notation can be simplified by introducing matrices V and D so that the above equation becomes $P = VDV^{\mathrm{H}}$. Ignoring terms that are independent of $\boldsymbol{x}$, the log of the likelihood ratio becomes

$$\acute{\boldsymbol{x}}^{\mathrm{H}}E^{-1}\acute{\boldsymbol{x}} - \acute{\boldsymbol{x}}^{\mathrm{H}}(E + VDV^{\mathrm{H}})^{-1}\acute{\boldsymbol{x}}$$

This simplifies to a quadratic form

$$\acute{\boldsymbol{x}}^{\mathrm{H}}E^{-1}VTV^{\mathrm{H}}E^{-1}\acute{\boldsymbol{x}} = \acute{\boldsymbol{x}}^{\mathrm{H}}W\acute{\boldsymbol{x}}$$

where T is a 2×2 matrix defined by

$$T^{-1} = D^{-1} + V^{\mathrm{H}}E^{-1}V$$

and W is a $2n \times 2n$ nonnegative matrix of rank 2. This provides justification for the square law detector discussed above.

OTHER DISTRIBUTIONS

As signal-processing applications become more sophisticated, other functions of complex variables come into play. For example, in the above discussions products of complex variables have already been encountered. In some deconvolution problems, quotients also arise.

The extension of standard probability theory to complex variables is an interesting exercise. The reason is that probability density functions are not analytic functions. (Obviously, they cannot be, since they always take on only real values.) Thus, standard theory of analytic continuation is not helpful. It seems that the easiest way to deal with this is simply to modify the basic definitions to accommodate the complex numbers and then do a set of derivations that parallel those already familiar for real variables. The following table shows some of the parallel formulas. (In the Gaussian case, only cir-

Real Variables	Complex Variables
	$dA_x = dx_r\, dx_i$
Probability density function	
$P_X(X < x) = \int_{-\infty}^{x} p_X(t)\, dt$	$P_X(X \in A) = \int\int_A p_X(x)\, dA_x$
Average	
$\mathscr{E}[X] = \int_{-\infty}^{\infty} x\, p_X(x)\, dx$	$\mathrm{E}[X] = \int\int_{\infty} x p_X(x)\, dA_x$
Gaussian	
$p_X(x) = \dfrac{1}{\sqrt{2\pi\nu}} e^{-x^2/2\nu}$	$p_X(x) = \dfrac{1}{\pi\nu} e^{-x^*x/\nu}$
Gaussian (multivariable)	
$p_{\boldsymbol{X}}(x) = \dfrac{1}{(2\pi)^{n/2}\sqrt{\lvert C\rvert}} e^{-1/2\boldsymbol{x}^{\mathrm{T}}C^{-1}\boldsymbol{x}}$	$p_{\boldsymbol{X}}(x) = \dfrac{1}{\pi^n\lvert C\rvert} e^{-\boldsymbol{x}^{\mathrm{H}}C^{-1}\boldsymbol{x}}$
Sum $Z = X + Y$	
$p_Z(z) = \int_{-\infty}^{\infty} p_{X,Y}(z - y, y)\, dy$	$p_Z(z) = \int\int_{\infty} p_{X,Y}(z - y, y)\, dA_y$
Product (general) $Z = XY^*$	
$p_Z(z) = \int_{-\infty}^{\infty} \dfrac{p_{X,Y}(z/y, y)}{\lvert y\rvert} dy$	$p_Z(z) = \int\int_{\infty} \dfrac{p_{X,Y}(z/y^*, y)}{y^*y} dA_y$
Product (Gaussian) $Z = XY^*$	
$p_Z(z) = \dfrac{1}{\pi\sqrt{1-\rho}} \exp\left(\dfrac{\rho z}{1-\rho^2}\right) K_0\left(\dfrac{z}{1-\rho^2}\right)$	$p_Z(z) = \dfrac{2}{\pi(1-\rho^*\rho)} \exp\left(\dfrac{\rho^*z + \rho z^*}{1-\rho^*\rho}\right) K_0\left(\dfrac{2\sqrt{z^*z}}{1-\rho^*\rho}\right)$
Quotient (general) $Z = X/Y$	
$p_Z(z) = \int_{-\infty}^{\infty} \lvert y\rvert\, p_{X,Y}(zy, y)\, dy$	$p_Z(z) = \int\int_{\infty} y^*y\, p_{X,Y}(zy, y)\, dA_y$
Quotient (Gaussian) $Z = X/Y$	
$p_Z(z) = \dfrac{\sqrt{1-\rho^2}}{\pi[(z-\rho)^2 + (1-\rho^2)]}$	$p_Z(z) = \dfrac{(1-\rho^*\rho)}{\pi[(z-\rho)^*(z-\rho) + (1-\rho^*\rho)]^2}$
Moment generating function	
$m_X(s) = \int_{-\infty}^{\infty} e^{-sx} p_X(x)\, dx$	$m_X(s) = \int\int_{\infty} e^{-(s^*x + x^*s)} p_X(x)\, dA_x$
Gaussian moments	
$\mathrm{E}[X^{2i}] = \dfrac{(2i)!\,\nu^i}{2^i(i)!}$	$\mathrm{E}[(X^*X)^i] = i!\ \nu^i$
Fourth moment (Gaussian)	
$\mathrm{E}[X_1X_2X_3X_4] = \mathrm{E}[X_1X_2]\mathrm{E}[X_3X_4] + \mathrm{E}[X_1X_4]\mathrm{E}[X_3X_2] + \mathrm{E}[X_1X_3]\mathrm{E}[X_2X_4]$	$\mathrm{E}[X_1X_2^*X_3X_4^*] = \mathrm{E}[X_1X_2^*]\mathrm{E}[X_3X_4^*] + \mathrm{E}[X_1X_4^*]\mathrm{E}[X_3X_2^*]$

cular variables are considered.) For the real variable case, $\rho = \mathrm{E}[xy]$. For the complex case $\rho = \mathrm{E}[xy^*]$.

FUTURE TRENDS

As the above discussion indicates, there are numerous points where the current understanding is inadequate. The field is rich in opportunities for investigation of improved theory and techniques.

If one wants to improve on the methods described above, probably the best place to start will be to find ways to better incorporate *a priori* information into the procedure. A clear understanding of the problem and the nature of the data will often make the difference between a valuable and a useless analysis.

The use of higher-order cumulants as functions of higher-order moments which have the properties of correlations is increasing. Since cumulants above the second order are zero for Gaussian data, they may be a good way to filter out Gaussian noise in order to study non-Gaussian components. This use is handicapped by two problems. First, the probability distributions for the estimators are not as well understood. This makes testing of estimates, and estimation of false-alarm rates, difficult. This is aggravated by the fact that unless the sample size is very large, the random variability of the cumulant estimators is very large. Second, it is often not clear which cumulants to use. To date, the best innovations in this area seem to have consisted in clever identification of cumulants of interest.

The most useful data analysis techniques tend to be based on arguments from decision theory and/or game theory. Information theory has also played a role, primarily in the use of ideas about entropy. In the future, information theory will probably play a more important role. From this viewpoint, the binary decision problem, that is, the detection problem, seems well supported by convincing theoretical arguments. This is much less true for the multiple-hypothesis problem, that is the estimation problem. Occasionally, the basic ideas here should be carefully revisited.

ACKNOWLEDGMENTS

Much of the material on probability theory for complex variables was worked out on funds from the U.S. Office of Naval Research In-house Laboratory Independent Research program.

BIBLIOGRAPHY

1. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Function,* New York: Dover, 1972.
2. R. W. Hamming, *Digital Filters,* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1977.
3. C. Eckart, *Optimal Rectifier Systems for the Detection of Steady Signals,* La Jolla, CA: University of California Marine Physical Laboratory of the Scripps Institution of Oceanography, 1952.
4. A. M. Mood and F. A. Graybill, *Introduction to the Theory of Statistics,* New York: McGraw-Hill, 1963.
5. N. R. Goodman, *On the joint estimation of the spectra, cospectrum, and quadrature spectrum of a two-dimensional stationary gaussian process,* Technical Report, Engineering Statistics Laboratory, New York University, 1957.

Reading List

A. Bertlesen, On non-null distributions connected with testing that a real normal distribution is complex, *J. Multivariate Anal.,* **32**: 282–289, 1990.

R. Fortet, *Elements of Probability Theory,* London: Gordon and Breach, 1977.

C. G. Khatri and C. D. Bhavsar, Some asymptotic inferential problems connected with complex elliptical distribution, *J. Multivariate Anal.,* **35**: 66–85, 1990.

C. L. Nikias and M. Shao, *Signal Processing with Alpha-Stable Distributions and Applications,* New York: Wiley, 1995.

B. Picinbono, On circularity, *IEEE Trans. Signal Process.,* **42**: 3473–3482, 1994.

A. K. Saxena, Complex multivariate statistical analysis: An annotated bibliography, *Int. Statist. Rev.,* **46**: 209–214, 1978.

R. A. Wooding, The multivariate distribution of comlpex normal variables, *Biometrika,* **43**: 212–215, 1956. Historical interest aside, this paper is interesting for the connection with Hilbert transforms.

DAVID J. EDELBLUTE
SPAWAR Systems Center San Diego

Describing Functions
Standard Article
James H. Taylor[1]
[1]University of New Brunswick
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2409
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (657K)

Abstract

The sections in this article are

Basic Concepts and Definitions

Traditional Limit Cycle Analysis Methods: One Nonlinearity

Frequency Response Modeling

Methods for Analyzing the Performance of Nonlinear Stochastic Systems

Limit Cycle Analysis: Systems with Multiple Nonlinearities

Methods for Designing Nonlinear Controllers

Describing Function Methods: Concluding Remarks

Keywords: mathematical analysis; nonlinear systems; nonlinear control techniques; higher-order nonlinear systems; nonlinear oscillations; nyquist polar plots; nonlinear input/output characterizations; nonlinear stochastic systems; systems with multiple nonlinearities

| | |

Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

DESCRIBING FUNCTIONS

In this article we define and overview the basic concept of a describing function, and then we look at a wide variety of usages and applications of this approach for the analysis and design of nonlinear dynamical systems. More specifically, the following is an outline of the contents:

(1) Concept and definition of describing functions for sinusoidal inputs (*sinusoidal-input describing functions*) and for random inputs (*random-input describing functions*)
(2) Traditional application of sinusoidal-input describing functions for limit cycle analysis, for systems with one nonlinearity
(3) Application of sinusoidal-input describing function techniques for determining the *frequency response* of a nonlinear system
(4) Application of random-input describing functions for the performance analysis of nonlinear stochastic systems
(5) Application of sinusoidal-input describing functions for limit cycle analysis, for systems with multiple nonlinearities
(6) Application of sinusoidal-input describing function methods for the design of nonlinear controllers for nonlinear plants
(7) Application of sinusoidal-input describing functions for the implementation of linear self-tuning controllers for linear plants and of nonlinear self-tuning controllers for nonlinear plants

Basic Concepts and Definitions

Describing function theory and techniques represent a powerful mathematical approach for understanding (analyzing) and improving (designing) the behavior of nonlinear systems. In order to present describing functions, certain mathematical formalisms must be taken for granted, most especially differential equations and concepts such as step response and sinusoidal input response. In addition to a basic grasp of differential equations as a way to describe the behavior of a system (circuit, electric drive, robot, aircraft, chemical reactor, ecosystem, etc.), certain additional mathematical concepts are essential for the useful application of describing functions—Laplace transforms, Fourier expansions, and the frequency domain being foremost on the list. This level of mathematical background is usually achieved at about the third or fourth year in undergraduate engineering.

The main motivation for describing function (*DF*) techniques is the need to understand the behavior of nonlinear systems, which in turn is based on the simple fact that every system is nonlinear except in very limited operating regimes. Nonlinear effects can be beneficial (many desirable behaviors can only be achieved by a nonlinear system, e.g., the generation of useful periodic signals or oscillations), or they can be detrimental

(e.g., loss of control and accident at a nuclear reactor). Unfortunately, the mathematics required to understand nonlinear behavior is considerably more advanced than that needed for the linear case.

The elegant mathematical theory for linear systems provides a unified framework for understanding all possible linear system behaviors. Such results do not exist for nonlinear systems. In contrast, different types of behavior generally require different mathematical tools, some of which are exact, some approximate. As a generality, exact methods may be available for relatively simple systems (ones that are of low order, or that have just one nonlinearity, or where the nonlinearity is described by simple relations), while more complicated systems may only be amenable to approximate methods. Describing function approaches fit in the latter category: approximate methods for complicated systems.

One way to deal with a nonlinear system is to linearize it. The standard approach, often called *small-signal linearization*, involves taking the derivative (slope) of each nonlinear term and using that slope as the gain of a linear term. As a simple example, an important cause of excessive fuel consumption at higher speeds is drag, which is often modeled with a term $Bv^2\ \mathrm{sgn}(v)$ (a constant times the square of the velocity times the sign of v). One may choose a nominal velocity, e.g., $v_0 = 100$ km/h, and approximate the incremental effect of drag with the linear term $2Bv_0(v - v_0) \triangleq 2Bv_0\ \delta v$. For velocity in the vicinity of 100 km/h (perhaps $|\delta v| \leq 5$ km/h), this may be reasonably accurate, but for larger variations it becomes a poor model—hence the term small-signal linearization.

The strong attraction of small-signal linearization is that the elegant theory for linear systems may be brought to bear on the resulting linear model. However, this approach can only explain the effects of small variations about the linearization point, and, perhaps more importantly, it can only reveal linear system behavior. This approach is thus ill suited for understanding phenomena such as nonlinear oscillation or for studying the limiting or detrimental effects of nonlinearity.

The basic idea of the DF approach for modeling and studying nonlinear system behavior is to replace each nonlinear element with a (quasi)linear descriptor, the DF, whose gain is a function of the input amplitude. The functional form of such a descriptor is governed by several factors: the type of input signal, which is assumed in advance, and the approximation criterion (e.g., minimization of mean square error). This technique is dealt with very thoroughly in a number of texts for the case of nonlinear systems with a single nonlinearity (1,2); for systems with multiple nonlinearities in arbitrary configurations, the most general extensions may be attributed independently to Kazakov 3 and Gelb and Warren 4 in the case of random-input DFs (*RIDFs*) and jointly to Taylor (5,6) and Hannebrink et al. (7) for sinusoidal-input DFs (*SIDFs*). Developments for multiple nonlinearities have been presented in tutorial form in Ramnath, Hedrick, and Paynter (8).

Two categories of DFs have been particularly successful: sinusoidal-input DFs and random-input DFs, depending, as indicated, upon the class of input signals under consideration. The primary texts cited above (1,2) and some other sources make a more detailed classification (e.g., SIDF for pure sinusoidal inputs, sine-plus-bias DFs if there is a nonzero dc value, RIDF for pure random inputs, random-plus-bias DFs); however, this seems unnecessary, since sine-plus-bias and random-plus-bias can be treated directly in a unified way, so we will use the terms SIDF and RIDF accordingly. Other types of DFs also have been developed and used in studying more complicated phenomena (e.g., two-sinusoidal-input DFs may be used to study effects of limit cycle quenching via the injection of a sinusoidal "dither" signal), but those developments are beyond the scope of this article.

The SIDF approach generally can be used to study periodic phenomena. It is applied for two primary purposes: limit cycle analysis (see the two sections below with that phrase in their titles) and characterizing the input/output behavior of a nonlinear plant in the frequency domain (see the section "Frequency Response Modeling"). This latter application serves as the basis for a variety of control system analysis and design methods, as outlined in the section "Methods for Designing Nonlinear Controllers." RIDF methods, on the other hand, are used for the analysis and design of stochastic nonlinear systems (systems with random signals), in analogous ways to the corresponding SIDF approaches, although SIDFs may be said to be more general and versatile, as we shall see.

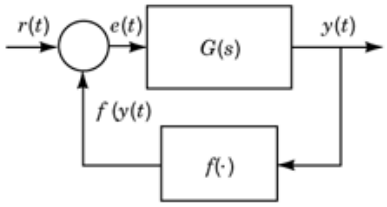


Fig. 1. System configuration with one dominant nonlinearity.

There is one additional theme underlying all the developments and examples in this article: Describing function approaches allow one to solve a wide variety of problems in nonlinear system analysis and design via the use of direct and simple extensions of linear systems analysis machinery. In point of fact, the mathematical basis is generally different (not based on linear systems theory); however, the application often results in conditions of the same form, which are easily solved. Finally, we note that the types of nonlinearity that can be studied via the DF approach are very general—nonlinearities that are discontinuous and even multivalued can be considered. The order of the system is also not a serious limitation. Given software such as MATLAB (trademark of The MathWorks, Inc.) for solving problems that are couched in terms of linear system mathematics (e.g., plotting the polar or Nyquist plot of a linear system transfer function), one can easily apply DF techniques to high-order nonlinear systems. The real power of this technique lies in these factors.

Introduction to Describing Functions for Sinusoids. The fundamental ideas and use of the SIDF approach can best be introduced by overviewing the most common application, limit cycle analysis for a system with a single nonlinearity. A *limit cycle* (*LC*) is a periodic signal, $x_{\mathrm{LC}}(t+T)=x_{\mathrm{LC}}(t)$ for all t and for some T (the *period*), such that perturbed solutions either approach x_{LC} (a stable limit cycle) or diverge from it (an unstable one).

The study of LC conditions in nonlinear systems is a problem of considerable interest in engineering. An approach to LC analysis that has gained widespread acceptance is the frequency-domain–SIDF method. This technique, as it was first developed for systems with a single nonlinearity, involved formulating the system in the form shown in Fig. 1, where $G(s)$ is defined in terms of a ratio of polynomials, as follows:

$$\begin{aligned} Y(s) \triangleq \mathcal{L}(y(t)) &= \frac{p(s)}{q(s)}E(s) \\ &\triangleq G(s)\mathcal{L}(e(t)) \\ e(t) &= r(t) - f(y(t)) \end{aligned} \tag{1}$$

where $(\cdot)$ denotes the Laplace transform of a variable and $p(s)$, $q(s)$ represent polynomials in the Laplace complex variable s, with order $(p) <$ order $(q) \triangleq n$. An alternative formulation of the same system is the state-space description,

$$\begin{aligned} \dot{x} &= Ax + be(t) \\ y(t) &= c^{\mathrm{T}}x \\ e(t) &= r(t) - f(y(t)) \end{aligned} \tag{2}$$

where x is an n-dimensional state vector. In either case, the first two relations describe a linear dynamic subsystem with input e and output y; the subsystem input is then given to be the external input signal $r(t)$ minus a nonlinear function of y. There is thus one single-input single-output (*SISO*) nonlinearity, $f(y)$, and linear

dynamics of arbitrary order. The state-space description is seen to be equivalent to the SISO transfer function on identifying $G(s) = c^{\mathrm{T}}(sI - A)^{-1}b$. Thus either system description is a formulation of the conventional "linear plant in the forward path with a nonlinearity in the feedback path" depicted in Fig. 1. The single nonlinearity might be an actuator or sensor characteristic, or a plant nonlinearity—in any case, the following LC analysis can be performed using this configuration.

In order to investigate LC conditions with no excitation, $r(t) = 0$, the nonlinearity is treated as follows: First, we assume that the input y is essentially sinusoidal, i.e., that a periodic input signal may exist, $y \approx a \cos \omega t$, and thus the output is also periodic. Expanding in a Fourier series, we have

$$f(a \cos \omega t) = \sum_{k=1}^{\infty} \mathrm{Re}\left[b_k(a) e^{jk\omega t}\right] \tag{3}$$

By omitting the constant or dc term from Eq. (3) we are implicitly assuming that $f(y)$ is an odd function, $f(-y) = -f(y)$ for all y, so that no rectification occurs; cases when $f(y)$ is not odd present no difficulty, but are omitted to simplify this introductory discussion. Then we make the approximation

$$\begin{aligned} f(a \cos \omega t) &\approx \mathrm{Re}\left[b_1(a) e^{j\omega t}\right] \\ &\triangleq \mathrm{Re}\left[N_{\mathrm{s}}(a) \cdot a\, e^{j\omega t}\right] \end{aligned} \tag{4}$$

This approximate representation for $f(a \cos \omega t)$ includes only the first term of the Fourier expansion of Eq. (3); therefore the approximation error (difference between $f(a \cos \omega t)$ and $\mathrm{Re}[N_{\mathrm{s}}(a) a\, e^{\mathrm{j}\omega t}]$) is minimized in the mean squared error sense (9). The Fourier coefficient b_1 [and thus the *gain* $N_{\mathrm{s}}(a)$] is generally complex unless $f(y)$ is single-valued; the real and imaginary parts of b_1 represent the in-phase (cosine) and quadrature (sine) fundamental components of $f(a \cos \omega t)$, respectively. The so-called describing function $N_{\mathrm{s}}(a)$ in Eq. (4) is, as noted, amplitude-dependent, thus retaining a basic property of a nonlinear operation.

By the principle of harmonic balance, the assumed oscillation—if it is to exist—must result in a loop gain of unity (including the summing junction), that is, substituting $f(y) \approx N_{\mathrm{s}}(a) y$ in Eq. (1) yields the requirement $N_{\mathrm{s}}(a)\, G(j\omega) = -1$, or

$$G(j\omega) = -1/N_{\mathrm{s}}(a) \tag{5}$$

For the state-space form of the model, using $X(j\omega)$ to denote the Fourier transform of x [$X(j\omega) = F(x)$], and thus $j\omega X = F(x)$, and again substituting $f(y) \approx N_{\mathrm{s}}(a) y$ in Eq. (2) yields the requirement

$$\left| j\omega I - A + N_{\mathrm{s}}(a) b c^{\mathrm{T}} \right| X(j\omega) = 0 \tag{6}$$

for some value of ω and $X(j\omega) \neq 0$. This is exactly equivalent to the condition (5).

The condition in Eq. (5) is easy to verify using the polar or Nyquist plot of $G(j\omega)$; in addition the LC amplitude a_{LC} and frequency ω_{LC} are determined in the process. More of the details of the solution for LC conditions are exposed in the following section. Note that the state-space condition, Eq. (6), appears to represent a quasilinearized system with pure imaginary eigenvalues; this is merely the first example showing how the describing function approach gives rise to conditions that seem to involve linear systems-theoretic concepts.

It is generally well understood that the classical SIDF analysis as outlined above is only approximate, so caution is always recommended in its use. The standard caveat that $G(j\omega)$ should be "low pass to attenuate higher harmonics" (so that the first harmonic in Eq. (3) is dominant) indicates that the analyst has to be

cautious. Nonetheless, as demonstrated by a more detailed example in the following section, this approach is simple to apply, very informative, and in general quite accurate. The main circumstance in which SIDF limit cycle analysis may yield poor results is in a borderline case, that is, one where the DF just barely cuts the Nyquist plot, or just barely misses it.

The next step in this brief introductory exposition of the SIDF approach involves showing a few elementary SIDF derivations for representative nonlinearity types. The basis for these evaluations is the well-known fact that a truncated Fourier series expansion of a periodic signal achieves minimum mean square approximation error (9), so we define the DF $N_s(a)$ as the first Fourier coefficient divided by the input amplitude [we divide by a so that $N_s(a)$ is in the form of a quasilinear gain]:

(1) *Ideal Relay.* $f(y) = D$ sgn $(y\ )$, where we assume no dc level, $y(t) = a \cos \omega t$. We set up and evaluate the integral for the first Fourier coefficient divided by a as follows:

$$\begin{aligned} N_s(a) &= \frac{1}{\pi a} \int_0^{2\pi} f(a \cos x) \cos x \, dx \\ &= \frac{2D}{\pi a} \int_0^{\pi} \cos x \, dx \qquad \text{(by symmetry)} \\ &= \frac{4D}{\pi a} \end{aligned} \tag{7}$$

(2) *Cubic Nonlinearity.* $f(y) = Ky^3(t)$; again, assuming $y(t) = a \cos \omega t$, we can directly write the Fourier expansion using the trigonometric identity for $\cos^3 x$ as follows:

$$\begin{aligned} f(a \cos \omega t) &= K(a \cos \omega t)^3 \\ &= K\, a^3 \left(\frac{3}{4} \cos \omega t + \frac{1}{4} \cos 3\omega t \right) \\ &\approx \frac{3\, K\, a^2}{4} \cdot a \cos \omega t \end{aligned} \tag{8}$$

so the SIDF is $N_s(a) = 3\ K\ a^2/4$. Note that this derivation uses trigonometric identities as a shortcut to formulating and evaluating Fourier integrals; this approach can be used for any power-law element.

The plots of $N_s(a)$ versus a for these two examples are provided in Fig. 2. Note the sound intuitive logic of these relations: a relay acts as a very high gain for small input amplitude but a low gain for large inputs, while just the opposite is true for the function $f(y) = K\, y^3(t)$. One more example is provided in the following section, to show that a multivalued nonlinearity (relay with hysteresis) in fact leads to a complex-valued SIDF. Other examples and an outline of useful SIDF properties are provided in a companion article, Nonlinear Control Systems, Analytical Methods. Observe that the SIDFs for many nonlinearities can be looked up in Refs. 1 and 2 (SIDFs for 33 and 51 cases are provided, respectively).

Finally, we demonstrate that the condition in Eq. (5) is easy to verify, using standard linear system analytic machinery and software:

Example 1. The developments so far provide the basis for a simple example of the traditional application of SIDF methods to determine LC conditions for a system with one dominant nonlinearity, defined in Eq. (1).

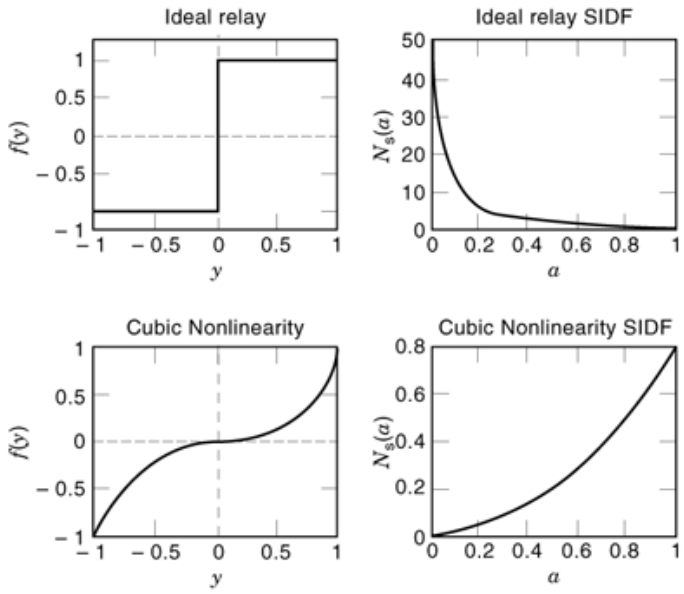


Fig. 2. Illustration of elementary SIDF evaluations.

Assume that the plant depicted in Fig. 1 is modeled by

$$G(s) = \frac{100\ (s+1)(s+70)}{(s+0.4)\ (s+2)(s+5+7j)\ (s+5-7j)\ (s+20)} \quad (9)$$

This transfer function might represent a servo amplifier and dc motor driving a mechanical load with friction level and spring constant adjusted to give rise to the lightly damped complex conjugate poles indicated, and the question is: will this cause limit cycling?

The Nyquist plot for this fifth-order linear plant is portrayed in Fig. 3, and the upper limit for stability is $K_{\max} = 2.07$; in other words, a linear gain k in the range [0, 2.07) will stabilize the closed-loop system if $f(y)$ is replaced by ky. According to Eq. (5), limit cycles are predicted if there is a nonlinearity $f(y)$ in the feedback path such that $-1/N_s(a)$ cuts the Nyquist curve, or in this case if $N_s(a)$ takes on the value 2.07. For the two nonlinearities considered so far, the ideal relay and the cubic characteristic, the SIDFs lie in the range [0, ∞), so limit cycles are indeed predicted in both cases. Furthermore, one can immediately determine the corresponding amplitudes of the LCs, namely setting $N_s(a) = 2.07$ in Eqs. (7), (8) yields an amplitude of $a_{LC} = 4\ D/(2.07\ \pi)$ for the relay and $a_{LC} = 8.28/(3\ K)$ for the cubic. In both cases $-1/N_s(a)$ cuts the Nyquist curve at the standard cross-over point on the real axis, so the LC frequency is $\omega_{LC} = \omega_{CO} = 8.11$/rad/s.

The LCs predicted by the SIDF approach in these two cases are fundamentally different, however, in one important respect: *stability* of the nonlinear oscillation. An LC is said to be stable if small perturbations from the periodic solution die out, that is, the waveform returns to the same periodic solution. While the general analysis for determining the stability of a predicted LC is complicated and beyond the scope of this article (1,2), the present example is quite simple: If points where $N_s(a\ )$ slightly exceeds $K_{\max}$ correspond to $a > a_{LC}$, then the LC is unstable, and conversely. Therefore, the LC produced by the ideal relay would be stable, while that produced by the cubic characteristic is unstable, as can be seen referring to Fig. 2. Note again that this

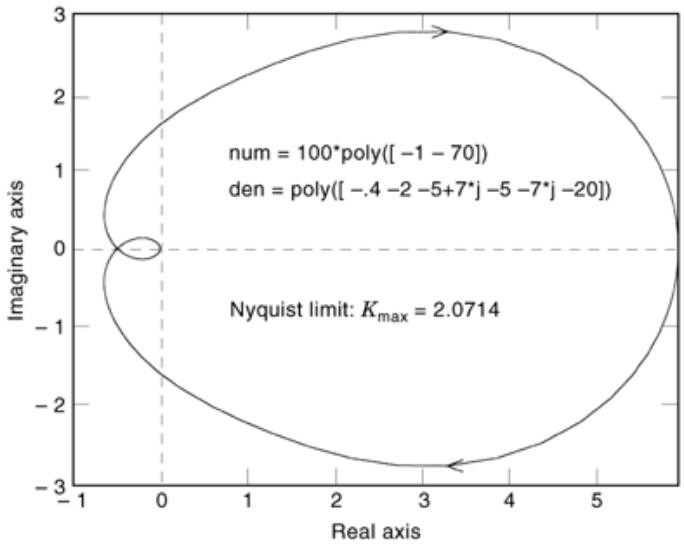


Fig. 3. Nyquist plot for plant in Example 1.

argument appears to be based on linear systems theory, but in fact the significance is that the loop gain for a periodic signal should be less than unity for $a > a_{LC}$ if a is not to grow.

To summarize this analysis, we first noted the simple approximate condition that allows a periodic signal to perpetuate itself: Eq. (5), that is, the loop gain should be unity for the fundamental component. We then illustrated the basis for and calculation of SIDFs for two simple nonlinearities. These elements came together using the well-known Nyquist plot of $G(j\omega)$ to check if the assumption of a periodic solution is warranted and, if so, what the LC amplitude would be. We also briefly investigated the stability question, i.e., for which nonlinearity the predicted LC would be stable.

Formal Definition of Describing Functions for Sinusoidal Inputs. The preceding outline of SIDF analysis of LC conditions illustrates the factors mentioned previously, namely the dependence of DFs upon the type of input signal and the approximation criterion. To express the standard definition of the SIDF more completely and formally,

- The nonlinearity under consideration is $f(y(t))$, and is quite unrestricted in form; for example, $f(y)$ may be discontinuous and/or multivalued.
- The class of input signals is $y(t) = y_0 + a \cos \omega t$, and the input amplitudes are quantified by the parameters (values) y_0, a.
- The SIDFs are denoted $N_s(y_0, a)$ for the sinusoidal component and $F_0(y_0, a)$ for the constant or dc part; that is, the nonlinearity output is approximated by

$$f(y(t)) \approx F_0(y_0, a) + \mathrm{Re}\left[N_s(y_0, a) \cdot a e^{j\omega t}\right] \qquad (10)$$

- The approximation criterion in Eq. (10) is the minimization of mean squared error,

$$E_{\text{ms}} \triangleq \int_0^{2\pi/\omega} \Big[f(y_0 + a\cos\omega t) - F_0(y_0, a) - \text{Re}\Big(N_s(y_0, a) \cdot a e^{j\omega t}\Big)\Big]^2 dt \qquad (11)$$

Again, under the above conditions it can be shown that $F_0(y_0,a)$ and $a\,N_s(y_0, a)$ are the constant (dc) and first harmonic coefficients of a Fourier expansion of $f(y_0 + a\cos\omega t)$ (9).

Note that this approximation of a nonlinear characteristic actually retains *two* important properties: amplitude dependence and the coupling between the dc and first harmonic terms. The latter property is a result of the fact that superposition does not hold for nonlinear systems, so, for example, N_s depends on both y_0 and a.

Describing Functions for General Classes of Inputs. An elegant unified approach to describing function derivation is given in Gelb and Vander Velde 1, using the concept of *amplitude density functions* to put all DF formulae on one footing. Here we will not dwell on the theoretical background and derivations, but just present the basic ideas and results.

Assume that the input to a nonlinearity comprises a bias component b and a zero mean component $z(t)$, in other words, $y(t) = b + z(t)$. In terms of random variable theory, the expectation of z, denoted $E(z)$, is zero and thus $E(y) = b$. The nonlinearity input y may be characterized by its amplitude density function, $p(\alpha)$, defined in terms of the amplitude distribution function $P(\alpha)$ as follows:

$$p(\alpha) = \frac{dP(\alpha)}{d\alpha}, \qquad P(\alpha) = \text{Prob}\,[y(t) < \alpha] \qquad (12)$$

A well-known density function corresponds to the Gaussian or normal distribution,

$$p_n(\alpha) = \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{(\alpha - b)^2}{2\sigma_n^2}\right) \qquad (13)$$

Other simple amplitude density functions are called *uniform*, where the amplitude of y is assumed to lie between $b - A$ and $b + A$ with equal likelihood $1/2A$,

$$p_u(\alpha) = \begin{cases} 0, & \alpha < b - A \\ 1/2A, & b - A \le \alpha \le b + A \\ 0, & \alpha > b + A \end{cases} \qquad (14)$$

and *triangular*,

$$p_t(\alpha) = \begin{cases} 0, & \alpha < b - A \\ \frac{1}{A}\left(1 - \frac{|\alpha - b|}{A}\right), & b - A \le \alpha \le b + A \\ 0, & \alpha > b + A \end{cases} \qquad (15)$$

Note that these three density functions are formulated so that the expected value of the variable is b in each case, and the area under the curve is unity (Prob $[y(t) < \infty] = \text{i}1^{\infty}{}_{-\infty}\, p(\alpha)\, d\alpha = 1$). From the standpoint of random variable theory the next most important expectation is $\Sigma = E((y(t) - b)^2)$, the *variance* or mean square value. In the normal case this is simply $\Sigma_{\text{n}} = \sigma^2{}_{\text{n}}$, where σ_{n} thus represents the *standard deviation*; for the other amplitude density functions we have $\Sigma_{\text{u}} = A^2/3$ and $\Sigma_{\text{t}} = A^2/6$. To express the corresponding DFs in terms of amplitude, the most commonly accepted measure is the standard deviation, $\sigma_{\text{u}} = A/\text{r1br3er}$ and $\sigma_{\text{t}} = A/\text{r1br6}$

Now, in order to unify the derivation of DFs for sinusoids as well as the types of variables mentioned above, we need to express the amplitude density function of such signals. A direct derivation, given $y(t) = b + a \cos \omega t$, yields

$$p_{\text{s}}(\alpha) = \begin{cases} 0, & \alpha < b - a \\ \dfrac{1}{\pi\sqrt{a^2 - (\alpha - b)^2}}, & b - a \le \alpha \le b + a \\ 0, & \alpha > b + a \end{cases} \tag{16}$$

Again, this density function is written so that $E(y) = b$; it is well known that the root mean square (rms) value of $a \cos \omega t$ is $\sigma_{\text{s}} = a/\text{r1br2}$

The above notation and terminology has been introduced primarily so that random-input describing functions (*RIDFs*) can be defined. We have, however, put signals with sinusoidal components into the same framework, so that one definition fits all cases. This leads to the following relations:

$$F_0(\sigma, b) = \int_{-\infty}^{\infty} f(b + z) p(z)\, dz \tag{17}$$

$$N_z(\sigma, b) = \frac{1}{\Sigma} \int_{-\infty}^{\infty} z f(b + z) p(z)\, dz \tag{18}$$

These relations again provide a minimum mean square approximation error. For separable processes (2), this amounts to assuming that the amplitude density function of the nonlinearity output is of the same class as the input, e.g., the RIDF for the normal case provides the gain that fits a normal amplitude density function to the actual amplitude density function of the output with minimum mean square error. There is only one restriction compared with the Fourier-series method for deriving SIDFs: multivalued nonlinearities (such as a relay with hysteresis) cannot be treated using the amplitude density function approach. The case of evaluating SIDFs for multivalued nonlinearities is illustrated in Example 2 in the following section; it is evident that the same approach will not work for signals defined only in terms of amplitude density functions.

Describing Functions for Normal Random Inputs. The material presented in the preceding section provides all the machinery needed for defining the usual class of RIDFs, namely those for Gaussian, or normally distributed, random variables:

$$F_0(\sigma_{\text{n}}, b) = \frac{1}{\sqrt{2\pi}\, \sigma_{\text{n}}} \int_{-\infty}^{\infty} f(b + z) \exp\left(-\frac{(z - b)^2}{2\sigma_n^2}\right) dz \tag{19}$$

$$N_{\text{n}}(\sigma_{\text{n}}, b) = \frac{1}{\sqrt{2\pi}\, \sigma_{\text{n}}^3} \int_{-\infty}^{\infty} z f(b + z) \exp\left(-\frac{(z - b)^2}{2\sigma_n^2}\right) dz \tag{20}$$

Considering the same nonlinearities discussed in the first section, the following results can be derived:

(1) *Ideal Relay.* $f(y) = D$ sgn (y), where we assume no bias $(b = 0)$, $y(t) = z(t)$ a normal random variable. We set up and evaluate the expectation in Eq. (20) as follows:

$$\begin{aligned} N_{\mathrm{n}}(\sigma_{\mathrm{n}}) &= \frac{1}{\sqrt{2\pi}\,\sigma_{\mathrm{n}}^3} \int_{-\infty}^{\infty} z f(b+z) \exp\left(-\frac{z^2}{2\sigma_{\mathrm{n}}^2}\right) dz \\ &= \frac{2D}{\sqrt{2\pi}\,\sigma_{\mathrm{n}}^3} \int_{0}^{\infty} z \exp\left(-\frac{z^2}{2\sigma_{\mathrm{n}}^2}\right) dz \quad \text{(by symmetry)} \\ &= \frac{\sqrt{2}D}{\sqrt{\pi}\sigma_{\mathrm{n}}} \end{aligned} \tag{21}$$

(2) *Cubic Nonlinearity.* $f(y) = K y^3$ (t). Again, assuming no bias, the random component DF is

$$\begin{aligned} N_{\mathrm{n}}(\sigma_{\mathrm{n}}) &= \frac{1}{\sqrt{2\pi}\,\sigma_{\mathrm{n}}^3} \int_{-\infty}^{\infty} z f(b+z) \exp\left(-\frac{z^2}{2\sigma_{\mathrm{n}}^2}\right) dz \\ &= \frac{2K}{\sqrt{2\pi}\sigma_{\mathrm{n}}^3} \int_{0}^{\infty} z^4 \exp\left(-\frac{z^2}{2\sigma_{\mathrm{n}}^2}\right) dz \qquad \text{(by symmetry)} \\ &= 3K\sigma_{\mathrm{n}}^2 \end{aligned} \tag{22}$$

Comparison of Describing Functions for Different Input Classes. Given the unified framework in the subsection "Describing: Functions for General Classes of Inputs" above, it is natural to ask: how much influence does the assumed amplitude density function have on the corresponding DF? To provide some insight, we may investigate the specific DF versus amplitude plots for the four density functions presented above and for the unity limiter or saturation element,

$$f(y) = \begin{cases} y, & |y| < 1 \\ \mathrm{sgn}(y), & |y| \geq 1 \end{cases} \tag{23}$$

and the cubic nonlinearity. These results are obtained by numerical integration in MATLAB and shown in Fig. 4. From an engineering point of view, the effect of varying the assumed input amplitude density function is not dramatic. For the limiter, the spread, (SIDF − RIDF)/2, is less than 10% of the average, (SIDF + RIDF)/2, which provides good agreement. For the cubic case, the ratio of the RIDF to the SIDF is more substantial, namely a factor of two [taking into account that $a^2 = 2\,\sigma^2{}_{\mathrm{s}}$ in Eq. (8)].

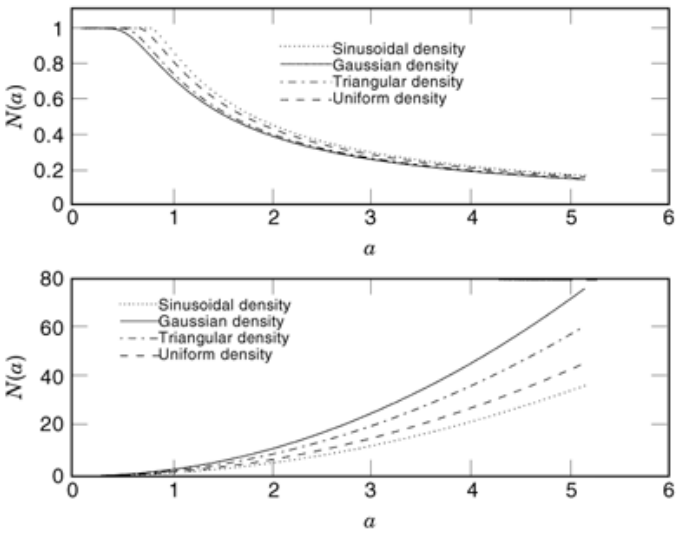


Fig. 4. Influence of amplitude density function on DF evaluation: Limiter (top), cubic nonlinearity (bottom).

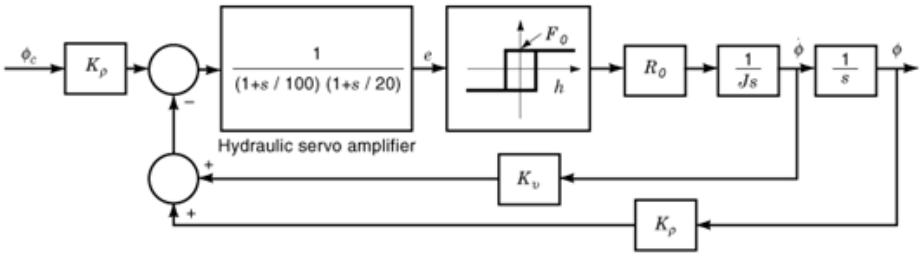


Fig. 5. Block diagram, missile roll-control problem (10).

Traditional Limit Cycle Analysis Methods: One Nonlinearity

Much of the process, terminology, and derivation for the traditional approach to limit cycle analysis has been presented in the preceding section. We proceed to investigate a more realistic (physically motivated) and complex example.

Example 2. A more meaningful example—and one that illustrates the use of complex-valued SIDFs to characterize multivalued nonlinearities—is provided by a missile roll control problem from Gibson (10): Assume a pair of reaction jets is mounted on the missile, one to produce torque about the roll axis in the clockwise sense and one in the counterclockwise sense. The force exerted by each jet is $F_0 = 445$ N, and the moment arms are $R_0 = 0.61$ m. The moment of inertia about the roll axis is $J = 4.68$ N·m/s^2. Let the control jets and associated servo actuator have a hysteresis $h = 22.24$ N and two lags corresponding to time constants of 0.01 s and 0.05 s. To control the roll motion, there is roll and roll-rate feedback, with gains of $K_\mathrm{p} = 1868$ N/rad and $K_\mathrm{v} = 186.8$ N/(rad/s) respectively. The block diagram for this system is shown in Fig. 5.

Before we can proceed with solving for the LC conditions for this problem, it is necessary to turn our attention to the derivation of complex-valued SIDFs for multivalued characteristics (relay with hysteresis). As in the introductory section, we can evaluate this SIDF quite directly.

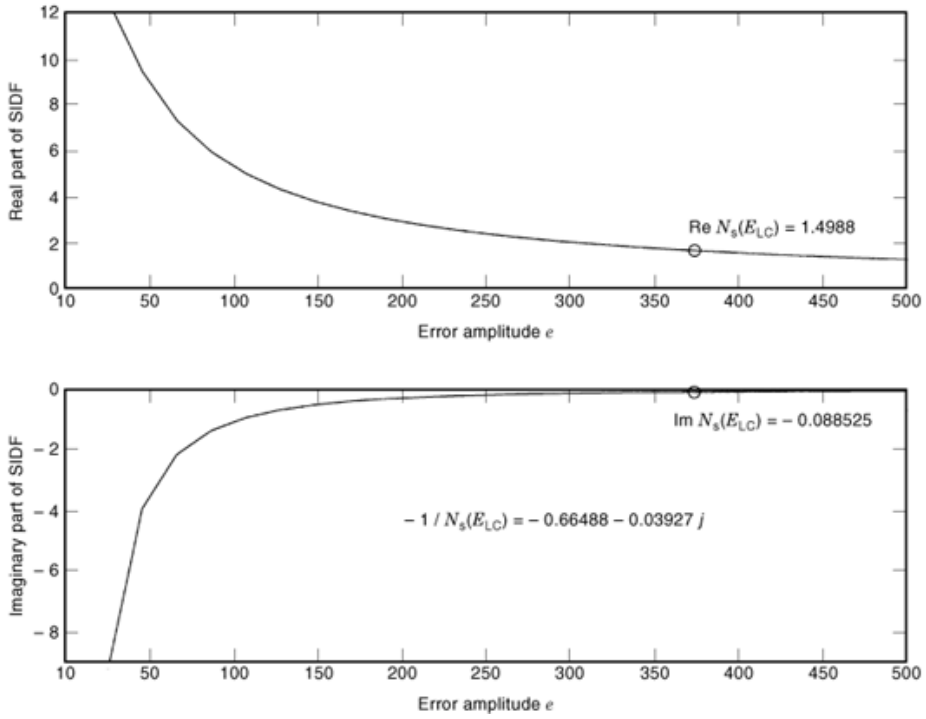


Fig. 6. Complex-valued SIDF for relay with hysteresis.

Defining the output levels to be $\pm D$ and the hysteresis to be h ($D = F_0$ in Fig. 5), then if we assume no dc level [$y(t) = a \cos \omega t$], we can set up and evaluate the integral for the first Fourier coefficient divided by a as follows:

$$\begin{aligned} N_s(a) &= \frac{1}{\pi a}\int_0^{2\pi} f(a\cos x)\exp(jx)dx \\ &= \frac{2D}{\pi a}\left(\int_0^{x_1}\exp(jx)dx - \int_{x_1}^{\pi}\exp(jx)dx\right) \quad \text{(by symmetry)} \\ &= \frac{4D}{\pi a}\left[\sqrt{1-\left(\frac{h}{a}\right)^2} - j\frac{h}{a}\right] \end{aligned} \tag{24}$$

where the switching point x_1 is $x_1 = \cos^{-1}(-h/a)$. Note that strictly speaking $N_s(a) \equiv 0$ if $a \leq h$, because the relay will not switch under that condition; the output will remain at D or $-D$ for all time, so the assumption that the nonlinearity output is periodic is invalid. The real and imaginary parts of this SIDF are shown in Fig. 6.

Given the SIDF for a relay with hysteresis, the solution to the problem of determining LC conditions for the system protrayed in Fig. 5 is depicted in Fig. 7. For a single-valued nonlinearity, and hence a real-valued SIDF, we would be interested in the real-axis crossing of $G(j\omega)$, at $\omega_{CO} = 28.3$ rad/s, $G_{CO} = -0.5073$. In this case, however, the intersection of $-1/N_s$ with $G(j\omega)$ no longer lies on the negative real axis, and so $\omega_{LC} = 24.36$ rad/s $\neq \omega_{CO}$. The amplitude of the variable e is read directly from the plot of $-1/N_s(a)$ as $E_{LC} = 377.2$; to obtain the LC

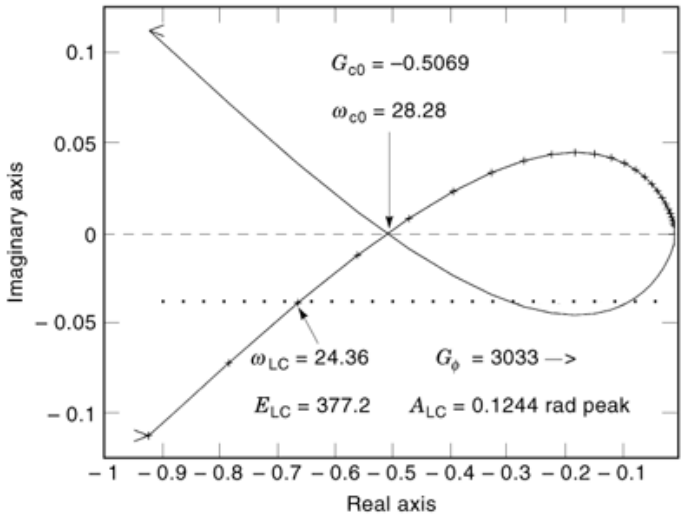


Fig. 7. Solution for missile roll-control problem: Nyquist diagrams.

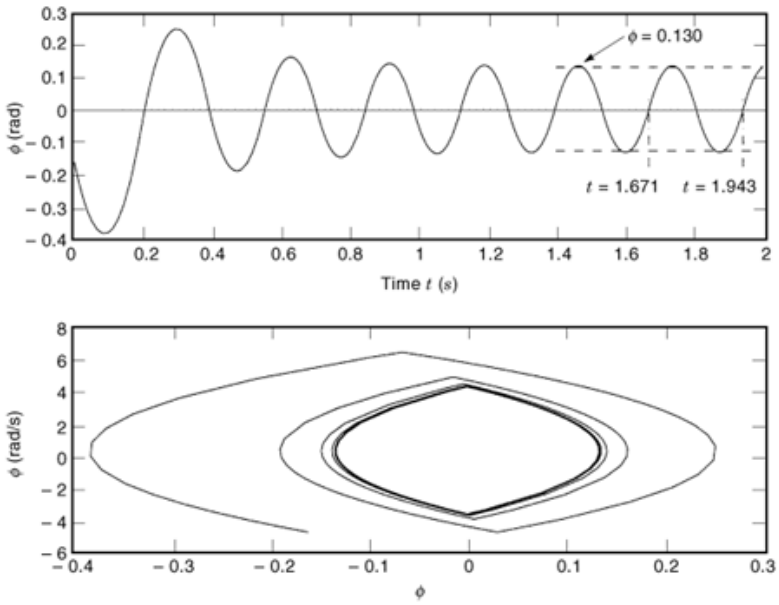


Fig. 8. Missile roll-control simulation result.

amplitude in roll, one must obtain the loop gain from e to ϕ, which is $G_\phi = R_0\, N_s(E_{LC})/(J\omega^2_{LC}) = 1/3033$, giving the LC amplitude in roll as $A_{LC} = 0.124$ rad (peak). In 10 it is said that an analog computer solution yielded $\omega_{LC} = 22.9$ rad/s and $A_{LC} = 0.135$ rad, which agrees quite well. A highly rigorous digital simulation approach (for which MATLAB-based software is available from the author (11)), using *modes* to capture the switching characteristics of the hysteretic relay, yielded $A_{LC} = 0.130$, $\omega_{LC} = 23.1$ rad/s, as shown in Fig. 8 which is in even

better agreement with the SIDF analysis. As is generally true, the approximation for ω_{LC} is better than that for a – the solution for ω_{LC} is a second-order approximation, while that for a is of first order (1,2).

Finally, it should be observed that for the particular case of nonlinear systems with relays a complementary approach for LC analysis exists, due to Tsypkin (12); see also Ref. 2 and Nonlinear Control Systems, Analytical Methods. Rather than assuming that the nonlinearity input is a sinusoid, one assumes that the nonlinearity output is a switching signal, in this case a signal that switches between F_0 and $-F_0$ at unknown times; one may solve exactly for the switching times and signal waveform by expressing the relay output in terms of a Fourier series expansion and solving for the switching conditions and coefficients. Alternatively, one could extend the SIDF approach by setting up and solving the harmonic balance relations for higher terms; that approach would converge to the exact solution as the number of harmonics considered increases (13).

Frequency Response Modeling

As mentioned in the introductory comments, SIDF approaches have been used for two primary purposes: limit-cycle analysis and characterizing the input/output (I/O) behavior of a nonlinear plant in the frequency domain. In this section we outline and illustrate two methods for determining the *amplitude-dependent frequency response* of a nonlinear system, hereafter more succinctly called an SIDF I/O model. After that, we discuss some broader but more complicated issues, to establish a context for this process.

Methods for Determining the Frequency Response. As mentioned, SIDF I/O modeling may be accomplished using either of two techniques:

(1) *Analytic Method Using Harmonic Balance.* Given the general nonlinear dynamics as

$$\dot{x} = f(x, u) \tag{25}$$

with a scalar input signal $u(t) = u_0 + b_a \cos \omega t$ and the n -dimensional state vector x assumed to be nearly sinusoidal,

$$x \approx x_c + \mathrm{Re}[a_x \exp(j\omega t)] \tag{26}$$

The variable a_x is a complex amplitude vector (in phasor notation), and x_c is the state vector center value. We proceed to develop a quasilinear state-space model of the system in which every nonlinear element is replaced analytically by the corresponding scalar SIDF, and formulate the quasilinear equations:

$$\begin{aligned} f(x, u) \approx\ & F_{DF}(u_0, b_u, x_c, a_x) \\ & + \mathrm{Re}[A_{DF}(u_0, b_u, x_c, a_x) \cdot a_x \exp(j\omega t)] \\ & + \mathrm{Re}[B_{DF}(u_0, b_u, x_c, a_x) \cdot b_u \exp(j\omega t)] \end{aligned} \tag{27}$$

Then, we formulate the equations of harmonic balance, which for the dc and sinusoidal components are

$$F_{DF}(u_0, b_u, x_c, a_x) = 0 \tag{28}$$

$$j\omega I a_x = A_{DF} a_x + B_{DF} b_u \tag{29}$$

One can, in principle, solve these equations for the unknown amplitudes x_c, a_x given values for u_0, b_a and then evaluate A_{DF} and B_{DF}; however, this is difficult in general and requires special nonlinear equation-solving software. Then, assuming finally that there is a linear output relation $y = C\,x$ for simplicity, the I/O model may be evaluated as

$$G(j\omega; u_0, b_u) = C[j\omega I - A_{DF}]^{-1} B_{DF} \tag{30}$$

Note that all arrays in the quasilinear model may depend on the input amplitude, u_0, b_a. This approach was used in Ref. 14 in developing an automated modeling approach called the *model-order deduction algorithm for nonlinear systems* (*MODANS*); refer to Ref. 15 for further details in the solution of the harmonic balance problem. This approach is subject to argument about the validity of assuming that every nonlinearity input is nearly sinusoidal. It is also more difficult than the following, and not particularly recommended. It is, however, the "pure" SIDF method for solving the problem.

(2) *Simulation Method.* Apply a sinusoidal signal to the nonlinear system model, perform direct Fourier integration of the system output in parallel with simulating the model's response to the sinusoidal input, and simulate until steady state is achieved to obtain the dynamic or frequency-domain SIDF $G(j\omega; u_0, b_a)$ (16).

To elaborate on the second method and illustrate its use, we assume for simplicity that $u_0 = 0$ and focus on determining $G(j\omega, b)$ for a range of input amplitudes $[b_{min}, b_{max}]$ to cover the expected operating range of the system and frequencies $[\omega_{min}, \omega_{max}]$ to span a frequency range of interest. Then specific sets of values $\{b_j \in [b_{min}, b_{max}]\}$ and $\{\omega_j \in [\omega_{min}, \omega_{max}]\}$ are selected for generating $G(j\omega_j, b_i)$. The nonlinear system model is augmented by adding new states corresponding to the Fourier integrals

$$\text{Re } G_K(j\omega_j, bi) = \frac{\omega_j}{\pi b_i} \int_{KT}^{(K+1)T} y(t) \cos \omega t \, dt$$

$$\text{Im } G_K(j\omega_j, b_i) = \frac{-\omega_j}{\pi b_i} \int_{KT}^{(K+1)T} y(t) \sin \omega t \, dt$$

where Re·and Im·are the real and imaginary parts of the SIDF I/O model $G(j\omega_j, b_i)$, $T = 2\pi/\omega_j$, and $y(t)$ is the output of the nonlinear system. In other words, the derivatives of the argumented states are proportional to $y(t)$ cos ωt and $y(t)$ sin ωt. Achieving steady state for a given b_j and w_j is guaranteed by setting tolerances and convergence criteria on the magnitude and phase of G_K, where K is the number of cycles simulated; the integration is interrupted at the end of each cycle, and the convergence criteria checked to see if the results are within tolerance (G_K is acceptably close to G_{K-1}), so that the simulation can be stopped and $G(j\omega_j, b_i)$ reported. For further detail, refer to Refs. 16 and 17 It should be mentioned that convergence can be slow if the simulation initial conditions are chosen without thought, especially if lightly damped modes are present. We have found that a converged solution point from the simulation for ω_j can serve as a good initial condition for the simulation for ω_{j+1}, especially if the frequencies are closely spaced. The MATLAB-based software for performing this task is available from the author.

Example 3. First, a brief demonstration of setting up and solving harmonic balance relations. Given a simple closed-loop system composed of an ideal relay and a linear dynamic block $W(j\omega)$, as shown in Fig. 9. If the input is $u(t) = b$ cos ωt, then $y(t) \approx \text{Re}[c(b)e^{j\omega t}]$ and similarly $e(t) \approx \text{Re}[e(b)e^{j\omega t}]$, where, in general, c and e are complex phasors. These three phasors are related by

$$\begin{aligned} c &= N_s(|e|)W(j\omega)e \\ &= N_s(|e|)W(j\omega)(b - c) \end{aligned} \tag{31}$$

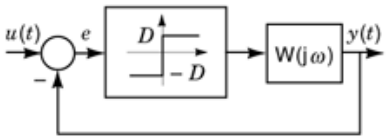


Fig. 9. System with relay and linear dynamics.

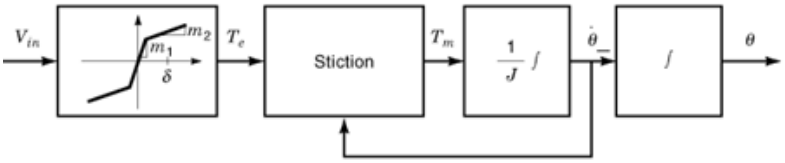


Fig. 10. Motor plus load: model schematic.

The SIDF for the ideal relay is $N_s = 4\,D/(\pi\,|e|)$, so the overall I/O relation is

$$G(j\omega, b) = \frac{c}{b} = \frac{4D}{\pi|b-c|} \cdot \frac{b-c}{b} \tag{32}$$

Taking the magnitude of this relation factor by factor yields

$$|G(j\omega, b)| = \frac{4D}{\pi b}|W(j\omega)| \tag{33}$$

It is interesting to note that the feedback does not change the frequency dependence of $|W(j\omega)|$, just the phase—this is not surprising, since the output of the relay always has the same amplitude, which is then modified only by $|W\,(j\omega)|$. The relationship between the phases of G and W is not so easy to determine, even in this special case (ideal relay).

Example 4. To illustrate the simulation approach to generating SIDF I/O models, consider the simple nonlinear model of a motor and load depicted in Fig. 10, where the primary nonlinear effects are torque saturation and stiction (nonlinear friction characterized by sticking whenever the load velocity passes through zero). This model has been used as a challenging exemplary problem in a series of projects studying various SIDF-based approaches for designing nonlinear controllers, as discussed in a later section. The mathematical model for stiction is given by the torque relation

$$T_m = \begin{cases} T_e - f_v\dot{\theta} - f_c\,\mathrm{sgn}(\dot{\theta}), & \dot{\theta} \neq 0 \text{ or } |T_e| > f_c \\ 0, & \text{otherwise} \end{cases} \tag{34}$$

where T_e and T_m denote electrical and mechanical torque, respectively, and, of necessity, we include a viscous friction term $f_v\theta$ along with a Coulomb component of value f_c.

To generate the amplitude-dependent SIDF models, we selected twelve frequencies, from 5 rad/s to 150 rad/s, and eight amplitudes, from quite small ($b_1 = 0.25$ V) to quite large ($b_8 = 12.8$ V). The results are shown in Fig. 11. The magnitude of $G(j\omega, b)$ varies by nearly a factor of 8, and the phase varies by up to 45 deg, showing that this system is substantially nonlinear over this operating range.

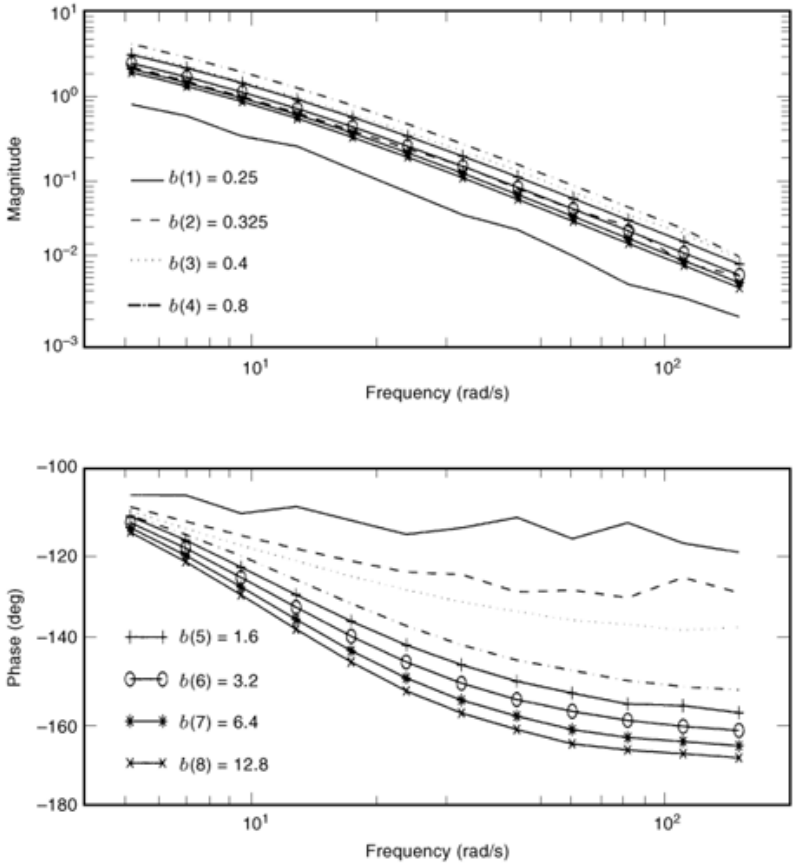


Fig. 11. Motor plus load: SIDF I/O models.

Methods for Accommodating Nonlinearity. Various ways exist to allow for amplitude sensitivity in nonlinear dynamic systems. This is a very important consideration, both generally and particularly in the context of models for control system design. Approaches for dealing with static nonlinear characteristics in such systems include replacing nonlinearities with linear elements having gains that lie in ranges based on:

- Nonlinearity sector bounds
- Nonlinearity slope bounds
- Random-input describing functions (*RIDFs*), or
- Sinusoidal-input describing functions (*SIDFs*)

In brief, frequency-domain plant I/O models based on SIDFs provide an excellent tradeoff between conservatism and robustness in this context. In particular, it can be shown by example that sector and slope bounds may be excessively conservative, while RIDFs are generally not robust, in the sense that a nonlinear control system design predicted to be stable based on RIDF plant models may limit cycle or be unstable. Another important attribute of SIDF-based frequency-domain models is that they allow for the fact that the effect of most nonlinear elements depends on frequency as well as amplitude; none of the other techniques capture both of these traits. These points are discussed in more detail below; note that it is assumed that no biases exist in

the nonlinear dynamic system, for the sake of simplicity; extending the arguments to systems with biases is straightforward.

Linear model families ($\dot{x} = Ax + Bu$) can be obtained by replacing each plant nonlinearity with a linear element having a gain that lies in a range based on its sector bound or slope bound. We will hereafter call these model families *sector I/O models* and *slope I/O models*, respectively. (Robustness cannot be achieved using *one* linear model based on the slope of each nonlinearity at the operating point for design, so that alternative is not considered.)

From the standpoint of robustness in the sense of maintaining stability in the presence of plant I/O variation due to amplitude sensitivity, it has been established that none of the model families defined above provide an adequate basis for a guarantee. The idea that sector I/O models would suffice is called the *Aizerman conjecture*, and the premise that slope *I/O* models are useful in this context is the *conjecture of Kalman*; both have been disproven even in the case of a single nonlinearity (for discussion, see Ref. 18). Both SIDF and RIDF models similarly can be shown to be inadequate for a robustness guarantee in this sense (see also Ref. 18).

Despite the fact that these model families cannot be used to guarantee stability robustness, it is also true that in many circumstances they are conservative. For example, a particular nonlinearity may pass well outside the sector for which the Aizerman conjecture would suggest stability, and yet the system may still be stable. On the other hand, only very conservative conditions such as those imposed by the Popov criterion (19) [MATLAB-based software for applying the Popov criterion is available from the author (20)] and the off-axis circle criterion (21) serve this purpose rigorously—however, the very stringent conditions these criteria impose and the difficulty of extension to systems with multiple nonlinearities generally inhibit their use. Thus many control system designs are based on one of the model families under consideration as a (hopeful) basis for robustness. It can be argued that designs based on SIDF I/O models that predict that limit cycles will not exist by a substantial margin is the best one can achieve in terms of robustness (see also Ref. 22). In SIDF-based synthesis the frequency-domain design objective (see "Design of Conventional Nonlinear Controllers" below) must ensure this.

Returning to conservatism, considering a static nonlinearity, and assuming that it is single-valued and its derivative exists everywhere, it can be stated that slope I/O models are always more conservative than sector I/O models, which in turn are always more conservative than SIDF models. This is because the range of an SIDF cannot exceed the sector range, and the sector range cannot exceed the slope range. An additional argument that sector and slope model families may be substantially more conservative than SIDF I/O models is based on the fact that only SIDF models allow for the frequency dependence of each nonlinear effect. This is especially important in the case of multiple nonlinearities, as illustrated by the simple example depicted in Fig. 12: Denoting the minimum and maximum slopes of the gain-changing nonlinearities f_k by $m_{_k}$ and $\bar{m}_k$ respectively, we see that the sector and slope I/O models correspond to all linear systems with gains lying in the indicated rectangle, while SIDF I/O models only correspond to a *gain trajectory* as shown (the exact details of which depend on the linear dynamics that precede each nonlinearity). In many cases, the SIDF model will clearly prove to be a less restrictive basis for control synthesis.

Returning to DF models, there are two basic differences between SIDF and RIDF models for a static nonlinearity, as mentioned above, namely, the assumed input amplitude distribution is different, and the fact that SIDFs can characterize the effective phase shift caused by multivalued nonlinearities such as those commonly used to represent hysteresis and backlash, while RIDFs cannot. In the section "Comparison of Describing Functions for Different Input Classes" above, we see that the input amplitude distribution issue is generally not a major consideration.

However, there is a third difference (15) that affects the I/O model of a nonlinear plant in a fundamental way. This difference is related to how the DF is used in determining the I/O model; the result is that RIDF plant models (as usually defined) also fail to capture the frequency dependence of the system nonlinear effects.

This difference arises from the fact that the standard RIDF model is the result of one quasilinearization procedure carried out over a wide band of frequencies, while the SIDF model is obtained by multiple quasilin-

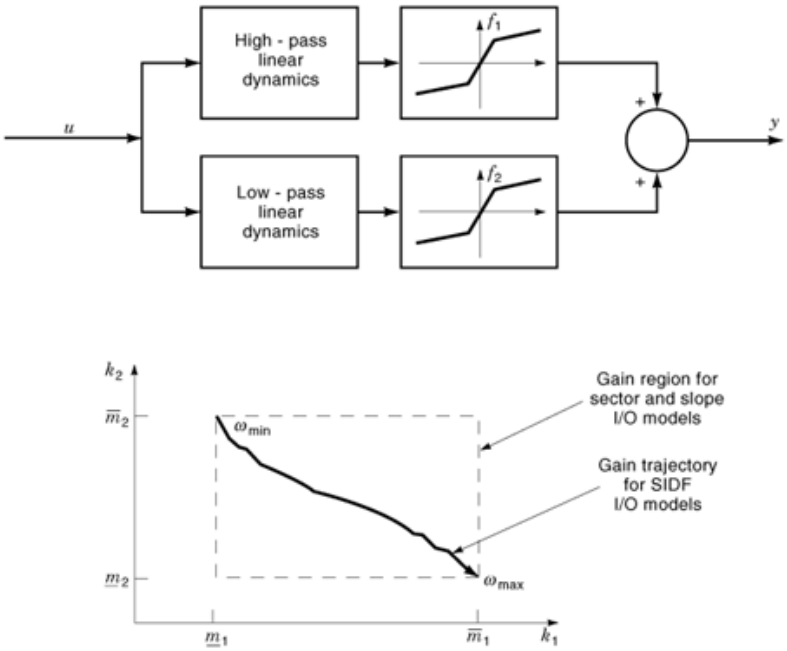


Fig. 12. Frequency dependence of multiple limiters.

earizations at a number of frequencies, as we have seen. This behavior is best understood via a simple example (15) involving a low-pass linear system followed by a saturation (unity limiter), defined in Eq. (23):

- Considering sinusoidal inputs of amplitude substantially greater than unity, the following behavior is exhibited: Low-frequency inputs are only slightly attenuated by the linear dynamics, resulting in heavy saturation and reduced SIDF gain; however, as frequency and thus attenuation increases, saturation decreases correspondingly and eventually disappears, giving a frequency response $G(j\omega, b)$ that approaches the response of the low-pass linear dynamics alone.
- A random input with rms value greater than one, on the other hand, results in saturation at all frequencies, so $G(j\omega, b)$ is identical to the linear dynamics followed by a gain less than unity.

In other words, the SIDF approach captures both a gain change and an effective increase in the transfer function magnitude corner frequency for larger input amplitudes, while the RIDF model shows only a gain reduction.

Methods for Analyzing the Performance of Nonlinear Stochastic Systems

This application of RIDFs represents the most powerful use of statistical linearization. It also represents a major departure from the class of problems considered so far. To set the stage, we must outline a class of nonlinear stochastic problems that will be tackled and establish some relations and formalism (23).

The dynamics of a nonlinear continuous-time stochastic system can be represented by a first-order vector differential equation in which $x(t)$ is the system state vector and $w(t)$ is a forcing function vector,

$$\dot{x}(t) = f(x, t) + G(t)w(t) \tag{35}$$

The state vector is composed of any set of variables sufficient to describe the behavior of the system completely. The forcing function vector $w(t)$ represents disturbances as well as control inputs that may act upon the system. In what follows, $w(t)$ is assumed to be composed of a mean or deterministic part $b(t)$ and a random part $u(t)$, the latter being composed of elements that are uncorrelated in time; that is, $u(t)$ is a white noise process having the spectral density matrix $Q(t)$. Similarly, the state vector has a deterministic part $m(t) = E[x(t)]$ and a random part $r(t) = x(t) - m(t)$; for simplicity, $m(t)$ is usually called the mean vector. Thus the state vector $x(t)$ is described statistically by its mean vector $m(t)$ and covariance matrix $S(t) = E[r(t)r^{\mathrm{T}}(t)]$. Henceforth, the time dependence of these variables will usually not be denoted explicitly by (t).

Note that the input to Eq. (35) enters in a linear manner. This is for technical reasons, related to the existence of solutions. (In Itô calculus the stochastic differential equation $dx_i = f(x, t)\, dt + g(x, t)\, d\beta_i$ has well-defined solutions, where β_i is a Brownian motion process. Equation (35) is an informal representation of such systems.) This is not a serious limitation; a stochastic system may have random inputs that enter nonlinearly if they are, for example, band-limited first-order Markov processes, characterized by $\dot{z} = A(t)z + B(t)w$, where again w contains white noise components—in this case, one may append the Markov process states z to the physical system states x so $f(x, z)$ models the nonlinear dependence of $\dot{x}$ on the random input z.

The differential equations that govern the propagation of the mean vector and covariance matrix for the system described by Eq. (35) can be derived directly, as demonstrated in Ref. 24, to be

$$\begin{aligned} \dot{m} &= E[f(x, t)] + G(t)b \\ &\triangleq \hat{f} + G(t)b \\ \dot{S} &= E[fr^{\mathrm{T}}] + E[rf^{\mathrm{T}}] + G(t)QG^{\mathrm{T}}(t) \end{aligned} \tag{36}$$

The equation for S can be put into a form analogous to the covariance equations corresponding to $f(x, t)$ being linear, by defining the auxiliary matrix N_{R} through the relationship

$$N_{\mathrm{R}}S \triangleq E[f(x, t)r^{\mathrm{T}}] \tag{37}$$

Note that the RIDF matrix N_{R} is the direct vector–matrix extension of the scalar describing function definition, Eq. (18). Then Eq. (36) may be written as

$$\begin{aligned} \dot{m} &= \hat{f} + G(t)b \\ \dot{S} &= N_{\mathrm{R}}S + SN_{\mathrm{R}}^{\mathrm{T}} + G(t)QG^{\mathrm{T}}(t) \end{aligned} \tag{38}$$

The quantities f° and N_{R} defined in Eq. (36) and (37) must be determined before one can proceed to solve Eq. (38). Evaluating the indicated expected values requires knowledge of the joint probability density function (pdf) of the state variables. While it is possible, in principle, to evolve the n -dimensional joint pdf $p(x, t)$ for a nonlinear system with random inputs by solving a set of partial differential equations known as the Fokker–Planck equation or the forward equation of Kolmogorov (24), this has only been done for simple, low-order systems, so this procedure is generally not feasible from a practical point of view. In cases where the pdf is not available, exact solution of Eq. (38) is precluded.

One procedure for obtaining an approximate solution to Eq. (38) is to assume the form of the joint pdf of the state variables in order to evaluate f° and N_R according to Eqs. (36) and (37). Although it is possible to use any joint pdf, most development has been based on the assumption that the state variables are jointly normal; the choice was made because it is both reasonable and convenient.

While the above assumption is strictly true only for linear systems driven by Gaussian inputs, it is often approximately valid in nonlinear systems with non-Gaussian inputs. Although the output of a nonlinearity with a Gaussian input is generally non-Gaussian, it is known from the central limit theorem (25) that random processes tend to become Gaussian when passed through low-pass linear dynamics (*filtered*). Thus, in many instances, one may rely on the linear part of the system to ensure that non-Gaussian nonlinearity outputs result in nearly Gaussian system variables as signals propagate through the system. By the same token, if there are non-Gaussian system inputs that are passed through low-pass linear dynamics, the central limit theorem can again be invoked to justify the assumption that the state variables are approximately jointly normal. The validity of the Gaussian assumption for nonlinear systems with Gaussian inputs has been studied and verified for a wide variety of systems.

From a pragmatic viewpoint, the Gaussian hypothesis serves to simplify the mechanization of Eq. (38) significantly, by permitting each scalar nonlinear relation in $f(x, t)$ to be treated in isolation (4), with f° and N_R formed from the individual RIDFs for each nonlinearity. Since RIDFs have been catalogued for numerous single-input nonlinearities (1,2), the implementation of this technique is a straightforward procedure for the analysis of many nonlinear systems.

As a consequence of the Gaussian assumption, the RIDFs for a given nonlinearity are dependent only upon the mean and the covariance of the system state vector. Thus, f° and N_R may be written explicitly as

$$\begin{aligned} \hat{f} &= \hat{f}(m, S, t) \\ N_R &= N_R(m, S, t) \end{aligned} \tag{39}$$

Relations of the form indicated in Eq. (39) permit the direct evaluation of f° and N_R at each integration step in the propagation of m and S. Note that the dependence of f° and N_R on the statistics of the state vector is due to the existence of nonlinearities in the system.

A comparison of quasilinearization with the classical Taylor series or small-signal linearization technique provides a great deal of insight into the success of the RIDF in capturing the essence of nonlinear effects. Figure 4 illustrates this comparison with concrete examples. If a saturation or limiter is present in a system and its input v is zero-mean, the Taylor series approach leads to replacing $f(v)$ with a unity gain regardless of the input amplitude, while quasilinearization gives rise to a gain that decreases as the rms value of v, σ_v, increases. The latter approximate representation of $f(v)$ much more accurately reflects the nonlinear effect; in fact, the saturation is completely neglected in the small-signal linear model, so it would not be possible to determine its effect. The fact that RIDFs retain this essential characteristic of system nonlinearities—input-amplitude dependence—provides the basis for the proven accuracy of this technique.

Example 5. An antenna pointing and tracking study is treated in some detail, to illustrate this methodology. This problem formulation is taken directly from Ref. 26; a discussion of the approach and results in Ref. 26 vis-à-vis the current treatment is given in Ref. 27.

The function of the antenna pointing and tracking system modeled in Fig. 13 is to follow a target line-of-sight (LOS) angle θ_t. Assume that θ_t is a deterministic ramp,

$$\dot{\theta}_t = \Omega u_{-1}(t) \tag{40}$$

where Ω is the slope of the ramp and u_{-1} denotes the unit step function. The pointing error, $e = \theta_t - \theta_a$ where θ_a is the antenna centerline angle, is the input to a nonlinearity $f(e)$, which represents the limited beamwidth

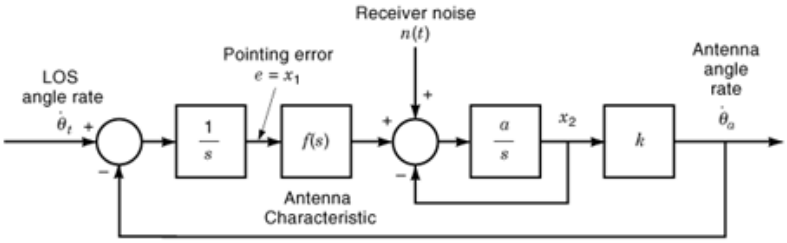


Fig. 13. Antenna pointing and tracking model.

of the antenna; for the present discussion,

$$f(e) = e(1 - k_a e^2) \tag{41}$$

where k_a is suitably chosen to represent the antenna characteristic. The noise $n(t)$ injected by the receiver is a white noise process having zero mean and spectral density q.

In a state-space formulation, Fig. 13 is equivalent to

$$\dot{x} = f(x) + w \tag{42}$$

where x_1 is the pointing error e, x_2 models the slewing of the antenna via a first-order lag, as defined in Fig. 13, and

$$f(x) = \begin{bmatrix} -kx_2 \\ a[f(x_1) - x_2] \end{bmatrix}, \qquad w = \begin{bmatrix} \dot{\theta}_t \\ a\,n(t) \end{bmatrix} \tag{43}$$

The statistics of the input vector w are given by

$$E[w] \triangleq b = \begin{bmatrix} \Omega \\ 0 \end{bmatrix} \tag{44}$$

$$E\left[[w(t) - b][w(\tau) - b]^T\right] \triangleq Q\delta(t - \tau) = \begin{bmatrix} 0 & 0 \\ 0 & a^2 q \end{bmatrix} \delta(t - \tau) \tag{45}$$

The initial state variable statistics, assuming $x_2(0) = 0$, are

$$E[x(0)] \triangleq m_0 = \begin{bmatrix} m_{e0} \\ 0 \end{bmatrix},$$
$$E\left[[x(0) - m_0][x(0) - m_0]^T\right] \triangleq \begin{bmatrix} \sigma_{e0}^2 & 0 \\ 0 & 0 \end{bmatrix} \tag{46}$$

where m_{e0} and σ_{e0} are the initial mean and standard deviation of the pointing error, respectively.

The above problem statement is in a form suitable for the application of the RIDF-based covariance analysis technique. The quasilinear RIDF representation for $f(e)$ in Eq. (41) is of the form

$$\begin{aligned} f(x_1) &\approx \hat{f} + N_n(x_1 - m_1) \\ &= [m_1 - k_a(m_1^2 + 3\sigma_1^2)m_1] + [1 - 3k_a(m_1^2 + \sigma_1^2)](x_1 - m_1) \end{aligned} \tag{47}$$

where m_1 and $\sigma^2{}_1$ are elements of m and S, respectively. The solution is then obtained by solving Eq. (38), which specializes to

$$\dot{m} = \begin{bmatrix} -km_2 \\ a(\hat{f} - m_2) \end{bmatrix} + b \tag{48}$$

$$\dot{S} = N_R S + S N_R^T + Q, \qquad N_R = \begin{bmatrix} 0 & -k \\ aN_n & -a \end{bmatrix} \tag{49}$$

subject to the initial conditions in Eq. (46). As noted previously, Eq. (49) is exact if x is a vector of jointly Gaussian random variables; if the initial conditions and noise are Gaussian and the effect of the nonlinearity is not too severe, the RIDF solution will provide a good approximation.

The goal of this study is to determine the system's tracking capability for various values of Ω; for brevity, only the results for $\Omega = 5$ deg/s are shown. The system parameters are: $a = 50\ \text{s}^{-1}$, $k = 10\ \text{s}^{-1}$, $k_a = 0.4\ \text{deg}^{-2}$; the pointing error initial condition statistics are $m_{e0} = 0.4$ deg, $\sigma_{e0} = 0.1$ deg; and the noise spectral density is $q = 0.004\ \text{deg}^2$. The RIDF solution depicted in Fig. 14 is based on the Gaussian assumption.

Three solutions are presented in Fig. 14, to provide a basis for assessing the accuracy of RIDF-based covariance analysis. In addition to the RIDF results, ensemble statistics from a 500-trial Monte Carlo simulation are plotted, along with the corresponding 95% confidence error bars, calculated on the basis of estimated higher-order statistics (28). Also, the results of a linearized covariance analysis are shown, based on assuming that the antenna characteristic is linear ($k_a = 0$). The linearized analysis indicates that the pointing error statistics reach steady-state values at about $t = 0.2$ s. However, it is evident from the two nonlinear analyses that this not the case: in fact, the tracking error can become so large that the antenna characteristic effectively becomes a negative gain, producing unstable solutions (the antenna loses track). The same is true for higher slewing rates; for example, $\Omega = 6$ deg/s was investigated in Refs. 26 and 27; in fact, the second-order Volterra analysis in Ref. 26 also missed the instability (loss of track). Returning to the RIDF-based covariance analysis solution, observe that the time histories of $m_1(t)$ and $\sigma_1(t)$ are well within the Monte Carlo error bars, thus providing a excellent fit to the Monte Carlo data. The fourth central moment was also captured, to permit an assessment of deviation from the Gaussian assumption; the parameter λ (*kurtosis*, the ratio of the fourth central moment to the variance squared) grew to $\lambda = 8.74$ at $t = 0.3$s, which indicates a substantial departure from the Gaussian case ($\lambda = 3$); this is the reason the error bars are so much wider at the end of the study ($t = 0.3$s) than near the beginning when in fact $\lambda \approx 3$—higher kurtosis leads directly to larger 95% confidence bands (28).

Many other applications of RIDF-based covariance analysis have been performed (for several examples, see Ref. 23). In every case, its ability to capture the significant impact of nonlinearities on system performance has been excellent, until system variables become highly non-Gaussian (roughly, until the kurtosis exceeds about 10 to 15). It is recommended that some cases be spot-checked by Monte Carlo simulation; however, one should recognize that one will have to perform many trials if the kurtosis is high, and that knowing how many trials to perform is itself problematical (for details, see Ref. 28).

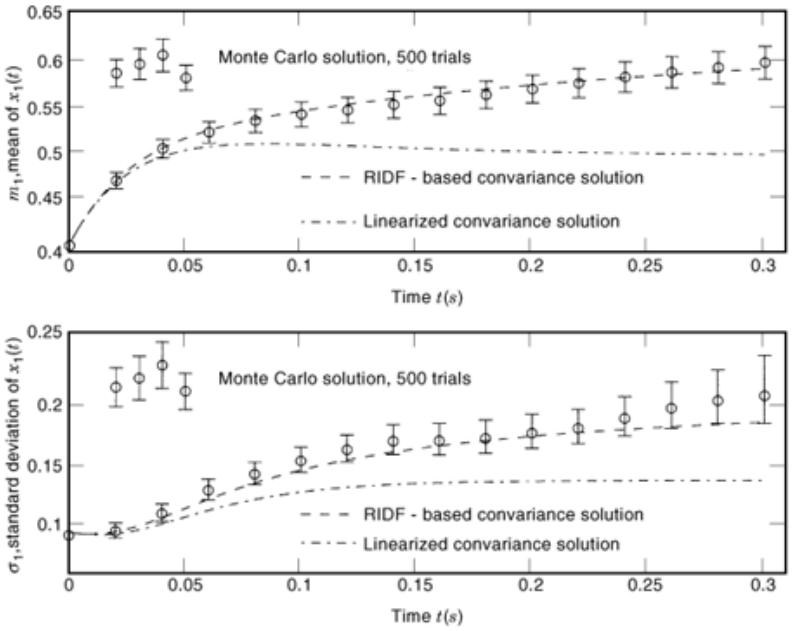


Fig. 14. Antenna pointing-error statistics.

Limit Cycle Analysis: Systems with Multiple Nonlinearities

Using SIDFs, as developed in the first two sections, is a well-known approach for studying LCs in nonlinear systems with one dominant nonlinearity. Once that problem was successfully solved, many attempts were made to extend this method to permit the analysis of systems containing more than one nonlinearity. At first, the nonlinear system models that could be treated by such extensions were quite restrictive (limited to a few nonlinearities, or to certain specific configurations; cf. Ref. 2). Furthermore, some results involved only conservative conditions for LC avoidance, rather than actual LC conditions. The technique described in this section (29) removes all constraints: Systems described by a general state vector differential equation, with any number of nonlinearities, may be analyzed. In addition, the nonlinearities may be multi-input, and bias effects can be treated. This general SIDF approach was first fully developed and applied to a study of wing-rock phenomena in aircraft at high angle of attack (5). It was also applied to determine limit cycle conditions for rail vehicles (7). Its power and use are illustrated here by application to a second-order differential equation derived from a two-mode panel flutter model (30). While the mathematical analysis is more protracted than in the single nonlinearity case, it is very informative and reveals the rich complexity of the problem.

The most general system model considered here is again as given in Eq. (25). Assuming that u is a vector of constants, denoted u_0, it is desired to determine if Eq. (25) may exhibit LC behavior. As before, we assume that the state variables are nearly sinusoidal [Eq. (26)], where a_x is a complex amplitude vector and x_c is the state-vector center value [which is not an equilibrium, or solution to $f(x_0, u_0) = 0$, unless the nonlinearities satisfy certain stringent symmetry conditions with respect to x_0]. Then we again neglect higher harmonics, to make the approximation

$$f(x, u_0) \approx F_{\mathrm{DF}}(u_0, x_c, a_x) + \mathrm{Re}[A_{\mathrm{DF}}(u_0, x_c, a_x) \cdot a_x \exp(j\omega t)] \tag{50}$$

The real vector F_{DF} and the matrix A_{DF} are obtained by taking the Fourier expansion of $f\ (x_c + \mathrm{Re}[a_x \exp(j\omega t)],\ u_0)$ as illustrated below, and provide the quasilinear or describing function representation of the nonlinear dynamic relation. The assumed LC exists for $u = u_0$ if x_c and a_x can be found so that

$$\begin{aligned} F_{\mathrm{DF}}(u_0, x_c, a_x) &= 0 \\ [j\omega I - A_{\mathrm{DF}}(u_0, x_c, a_x)]a_x &= 0, \quad a_x \neq 0 \end{aligned} \tag{51}$$

The nonlinear algebraic equations in Eq. (51) are often difficult to solve. A second-order system with two nonlinearities (from a two-mode panel flutter model) can be treated by direct analysis, as shown below (6). Even for this case the analysis is by no means trivial; it is included here for completeness and as guidance for the serious pursuit of LC conditions for multivariable nonlinear systems. It may be mentioned that iterative solution methods (e.g., based on successive approximation) have been used successfully to solve the dc and first-harmonic balance equations above for substantially more complicated problems such as the aircraft wing-rock problem [eight state variables, five multivariable nonlinear relations (5,29)]. More recently, a computer-aided design package for LC analysis of free-structured multivariable nonlinear systems was developed (13) using both SIDF methods and extended harmonic balance (including the solution of higher-harmonic balance relations); it is noteworthy that the extended harmonic balance approach can, in principle, provide solutions with excellent accuracy, as long as enough higher harmonics are considered—one interesting example included balancing up to the 19th harmonic.

Example 6. The following second-order differential equation has been derived to describe the local behavior of solutions to a two-mode panel flutter model (30):

$$\ddot{\chi} + (\alpha + \chi^2)\dot{\chi} + (\beta + \chi^2)\chi = 0 \tag{52}$$

Heuristically, it is reasonable to predict that LCs may occur for negative α (so the second term provides damping that is negative for small values of χ but positive for large values). Observe also that there are three singularities if β is negative: $\chi_0 = 0, \pm -\beta$ Making the usual choice of state vector, $x = [\chi\chi]^{\mathrm{T}}$, the corresponding state-vector differential equation is

$$\dot{x} = \begin{bmatrix} \dot{\chi} \\ \ddot{\chi} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\beta & -\alpha \end{bmatrix} x - \begin{bmatrix} 0 \\ x_1^2(x_1 + x_2) \end{bmatrix} \tag{53}$$

The SIDF assumption corresponding to Eq. (26) for this system of equations is that

$$\begin{aligned} x_1 = \chi &\approx \chi_c + a_1 \cos \omega t \\ x_2 = \dot{\chi} &\approx -a_1 \omega \sin \omega t \end{aligned}$$

(From the relation $x_2 = \chi$ it is clear that x_2 has a center value of 0 and that $a_2 = -j\omega a_1$—recognizing this at the outset greatly simplifies the analysis.) Therefore, the combined nonlinearity in Eq. (53) may be

quasilinearized to obtain

$$\begin{aligned} x_1^2(x_1+x_2) &= (\chi_c + a_1\cos\omega t)^2(\chi_c + a_1\cos\omega t - a_1\omega\sin\omega t) \\ &\approx \left(\chi_c^3 + \frac{3}{2}\chi_c a_1^2\right) + \left(3\chi_c^2 + \frac{3}{4}a_1^2\right)a_1\cos\omega t \\ &\quad + \left(\chi_c^2 + \frac{1}{4}a_1^2\right)(-a_1\omega\sin\omega t) \\ &\triangleq \Phi_0 + N_{s1}\cdot a_1\cos\omega t + N_{s2}\cdot(-a_1\omega\sin\omega t) \end{aligned} \tag{54}$$

This result is obtained by expanding the first expression using trigonometric identities to reduce $\cos^2 v$, $\cos^3 v$, and $\sin v \cos^2 v$ into terms involving $\cos kv$, $\sin kv$, $k = 0, 1, 2, 3$, and discarding all terms except the fundamental ones ($k = 0, 1$).

Therefore, the conditions of Eq. (51) require that

$$F_{DF} = \begin{bmatrix} 0 \\ -\chi_c\left(\beta + \chi_c^2 + \frac{3}{2}a_1^2\right) \end{bmatrix} = 0 \tag{55}$$

$$A_{DF} = \begin{bmatrix} 0 & 1 \\ -(\beta + 3\chi_c^2 + \frac{3}{4}a_1^2) & -(\alpha + \chi_c^2 + \frac{1}{4}a_1^2) \end{bmatrix} \triangleq \begin{bmatrix} 0 & 1 \\ -\omega_{LC}^2 & 0 \end{bmatrix} \tag{56}$$

where we have taken advantage of the canonical second-order form of an A matrix with imaginary eigenvalues $\pm j\omega_{LC}$; again, the canonical form of A_{DF} ensures harmonic balance, not "pure imaginary eigenvalues." The relation in Eq. (55) shows two possibilities:

- *Case 1.* $\chi_c = 0$, in which case Eq. (56) yields

$$a_1 = 2\sqrt{-\alpha}, \qquad \omega_{LC} = \sqrt{\beta - 3\alpha} \tag{57}$$

  The amplitude a_1 and frequency ω_{LC} must be real for LCs to exist. Thus, as conjectured, $\alpha < 0$ is required for a LC to exist centered about the origin. The second parameter must satisfy $\beta > 3\alpha$, so β can take on any positive value but cannot be more negative than 3α.
- *Case 2.* $\chi_c = \pm(\beta - 6\alpha)/5$, yielding

$$a_1 = 2\sqrt{\frac{\alpha - \beta}{5}}, \qquad \omega_{LC} = \sqrt{\beta - 3\alpha} \tag{58}$$

For the two LCs to exist, centered at $\chi_c = \pm$r1br($\beta - 6\alpha$)/5er, it is necessary that $3\alpha < \beta < \alpha$, so again limit cycles cannot exist unless $\alpha < 0$. One additional constraint must be imposed: $|\chi_c| > a_1$ must hold, or the two LCs will overlap; this condition reduces the permitted range of β to $2\alpha < \beta < \alpha$.

The stability of the Case 1 LC can be determined as follows: Take any $\epsilon > 0$, and perturb the LC amplitude to a slightly larger value, say, $a^2{}_1 = -4\alpha + \epsilon$. Substituting into Eq. (56) yields

$$A_{\mathrm{DF}} = \begin{bmatrix} 0 & 1 \\ -(\beta - 3\alpha + \frac{3}{4}\epsilon) - \frac{1}{4}\epsilon & \end{bmatrix} \qquad (59)$$

which for $\epsilon > 0$ has "*slightly stable* eigenvalues" (leads to loop gain less than unity). Thus a trajectory perturbed just outside the LC will decay, indicating that the Case 1 LC is stable. A similar analysis of the Case 2 LC is more complicated (since a perturbation in a_1 produces a shift in χ_c that must be considered), and thus is omitted.

Based on the SIDF-based LC analysis outlined above, the behavior of the original system Eq. (53) is portrayed for $\alpha = -1$ and seven values of β in Fig. 15. The analysis has revealed the rather rich set of possibilities that may occur, depending on the values of α and β. One may use traditional singularity analysis to verify the detail of the solutions near each center (29), as shown in Fig. 15, but that is beyond the scope of this article. Also, one may use center manifold techniques to analyze the LC behavior shown here (30), but that effort would require additional higher-level mathematics and substantially more analysis to obtain the same qualitative results. We provide one simulation example in Fig. 16, for $\alpha = -1$, $\beta = -1.1$, which should produce the behavior portrayed in case E in Fig. 15. The results for two initial conditions, $x_1 = 0.4, 0.5$, $x_2 = 0$ (marked $\circ$), bracket the unstable LC with predicted center at (0.99, 0), while for two larger starting values, $x_1 = 1.6, 2.0$, $x_2 = 0$ (marked $\times$), we observe clear convergence to the stable LC with predicted center at (0, 0). While the resultant stable oscillation is highly nonsinusoidal (and thus the Case 1 LC amplitude prediction is quite inaccurate), the SIDF prediction of panel flutter behavior is remarkably close.

Finally, it is worth mentioning that the "*eigenvectors*" of the matrix A_{DF} [state-vector amplitude vectors in phasor form, Eq. (51)] may be very useful in some cases. For the wing-rock problem mentioned previously we encountered *obscuring modes*, slow unstable modes that made it essentially impossible to use simulation to verify the SIDF-based LC predictions. We were able to circumvent this difficulty by picking simulation initial conditions based on a_x corresponding to the predicted LC and thereby minimizing the excitation of the unstable mode and giving the LC time to develop before it was obscured by the instability.

Methods for Designing Nonlinear Controllers

Describing function methods—especially, using SIDF frequency-domain models as illustrated in Fig. 11—for the design of linear and nonlinear controllers for nonlinear plants has a long history, and many approaches may be found in the literature. In general, the approach for *linear* controllers involves a direct use of frequency-domain design techniques applied to the family or a single (generally worst-case) SIDF model. The more interesting and powerful SIDF controller design approaches are those directed towards *nonlinear* compensation; that is the primary emphasis of the following discussions and examples.

A major issue in designing robust controllers for nonlinear systems is the amplitude sensitivity of the nonlinear plant and final control system. Failure to recognize and accommodate this factor may give rise to nonlinear control systems that behave differently for small and for large input excitation, or perhaps exhibit LCs or instability. Sinusoidal-input describing functions (*SIDFs*) have been shown to be effective in dealing with amplitude sensitivity in two areas: *modeling* (providing plant models that achieve an excellent tradeoff between conservatism and robustness, as in the section "Frequency Response Modeling" above) and *nonlinear control synthesis*. In addition, SIDF-based modeling and synthesis approaches are broadly applicable, in that there are very few and mild restrictions on the class of systems that can be handled. Several practical SIDF-

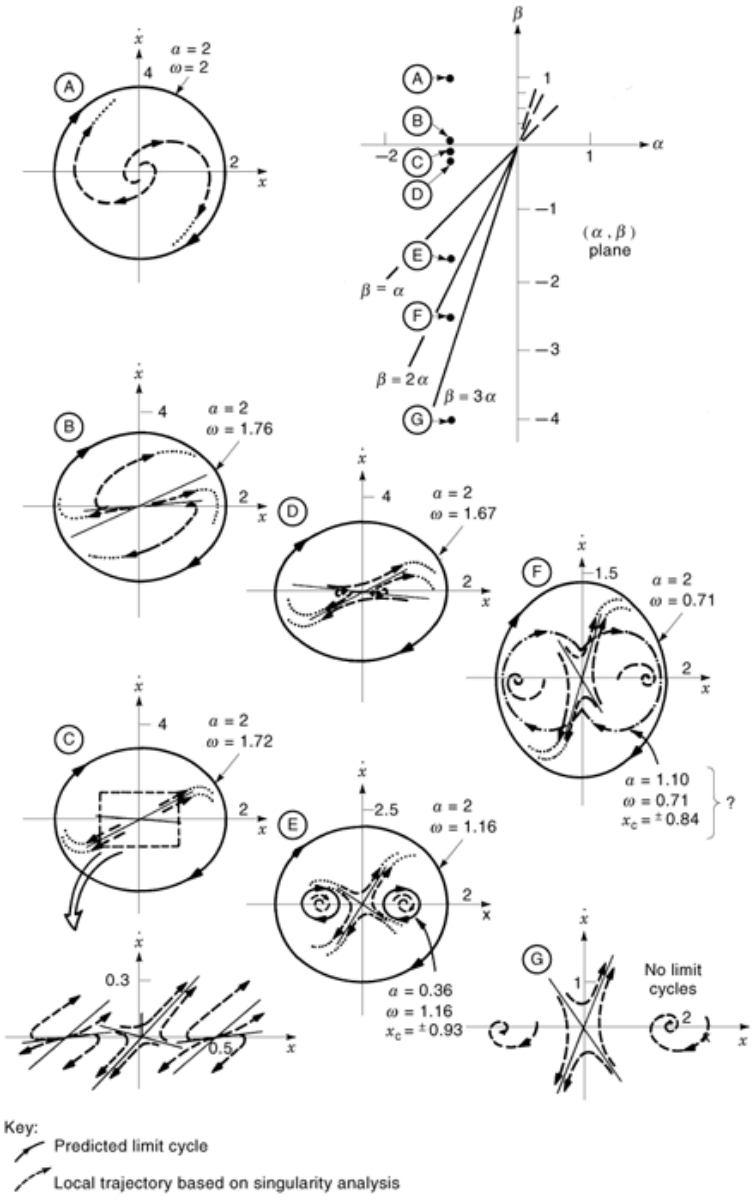


Fig. 15. Limit cycle conditions for the panel flutter problem (from Ref. 29).

based nonlinear compensator synthesis approaches are presented here and illustrated via application to a position control problem.

Before delving into specific approaches and results, the question of control system stability must be addressed. As mentioned in the subsection "Introduction to Describing Functions for Sinusoids" above, the use of SIDFs to determine LC conditions is not exact. Specifically, if one were to try to determine the critical value of a parameter that would just cause a control system to begin to exhibit LCs, the conventional SIDF result

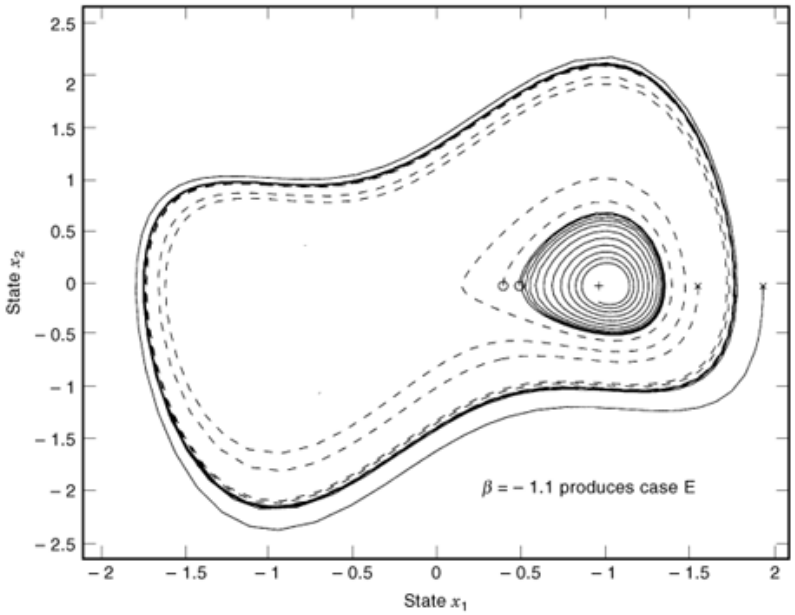


Fig. 16. Panel flutter problem: simulation result, $\alpha = -1$.

might not be very accurate [this is not to say that more detailed analysis such as inclusion of higher harmonics in the harmonic balance method could not eventually give accurate results (13)]. Therefore, if one desired to design a system to operate just at the margin of stability (onset of LCs), the SIDF method would not provide any guarantee.

To shun the use of SIDFs for nonlinear controller design for this reason would seem unwarranted, however. Generally one designs to *safe* specifications (good gain and phase margin, for example) that are far from the margin of stability or LC conditions. The use of SIDF models in these circumstances is clearly so superior to the use of conventional linearized models (or even a set of linearized models generated to try to cover for uncertainty and nonlinearity, as discussed in the section "Methods for Accommodating Nonlinearity" above) that we have no compunctions about recommending this practice. The added information (amplitude dependence) and intuitive support they provide is extremely valuable, as we hope the following examples demonstrate.

Design of Conventional Nonlinear Controllers. Design approaches based on SIDF models are all frequency-domain in orientation. The basic idea of a family of techniques developed by the author and students (15,16,31,32) is to define a *frequency-domain objective* for the open-loop compensated system and synthesize a nonlinear controller to meet that objective as closely as possible for a variety of error signal amplitudes (e.g., for small, medium, and large input signals, where the numerical values associated with the terms "small," "medium," and "large" are based on the desired operating regimes of the final system). The designs are then at least validated in the time domain [e.g., step-response studies (16,31)]; recent approaches have added time-domain optimization to further reduce the amplitude sensitivity (32,33). The methods presented below all follow this outline.

Modeling and synthesis approaches based on these principles are broadly applicable. Plants may have any number of nonlinearities, of arbitrary type (even discontinuous or hysteretic), in any configuration. These methods are robust in several senses: In addition to dealing effectively with amplitude sensitivity, the exact form of each plant nonlinearity does not have to be known as long as the SIDF plant model captures the amplitude sensitivity with decent accuracy, and the final controller design is not specifically based on precise knowledge

of the plant nonlinearities. The resulting controllers are simple in structure and thus readily implemented, with either piecewise linear characteristics (16,31) or fuzzy logic (32,33).

Before proceeding, it is important to consider the premises of the SIDF design approaches that we have been developing:

(1) The nonlinear system design problem being addressed is the synthesis of controllers that are effective for plants having frequency-domain I/O models that are *sensitive* to input amplitude (e.g., for plants that behave very differently for small, medium, and large input signals).
(2) Our primary objective in nonlinear compensator design is to arrive at a closed-loop system that is as *insensitive* to input amplitude as possible.

This encompasses a limited but important set of problems, for which gain-scheduled compensators cannot be used (gain-scheduled compensators can handle plants whose behavior differs at different operating points, but not amplitude-dependent plants; while a gain-scheduled controller is often implemented with piecewise linear relations or other curve fits to produce a controller that smoothly changes its behavior as the operating point changes, these curve fits are usually completely unrelated to the differing behavior of the plant for various input amplitudes *at a given operating point*) and for which other approaches (e.g., variable structure systems, model-reference adaptive control, global linearization) do not apply because their objectives are different (e.g., their objectives deal with *asymptotic* solution properties rather than *transient* behavior, or they deal with the behavior of *transformed* variables rather than physical variables).

A number of controller configurations have been investigated as these approaches were developed, ranging from one nonlinearity followed by a linear compensator (which has quite limited capability to compensate for amplitude dependence) to a two-loop configuration with nonlinear rate feedback and a nonlinear proportional–integral (*PI*) controller in the forward path. Since the latter is most effective, we will focus on that option (16).

An outline of the synthesis algorithm for the nonlinear PI plus rate feedback (PI + RF) controller is as follows:

(1) Select sets of input amplitudes and frequencies that characterize the operating regimes of interest.
(2) Generate SIDF I/O models of the plant corresponding to the input amplitudes and frequencies of interest (see the section "Frequency Response Modeling" above).
(3) Design amplitude-dependent RF gains, using an extension of the D'Azzo–Houpis algorithm (34) devised by Taylor and O'Donnell (16).
(4) Convert these linear designs into a piecewise linear characteristic (RF nonlinearity) by sinusoidal-input describing function inversion.
(5) Find SIDF I/O models for the nonlinear plant plus nonlinear RF compensation.
(6) Design PI compensator gains using the frequency-domain sensitivity minimization technique described in Taylor and O'Donnell (16).
(7) Convert these linear designs into a piecewise linear PI controller, also by sinusoidal-input describing function inversion.
(8) Develop a simulation model of the plant with nonlinear PI + RF control.
(9) Validate the design through step-response simulation.

Steps 1, 2, and 5 are already described in detail in the section "Frequency Response Modeling." In fact, the example and SIDF I/O model presented there (Fig. 11) were used in demonstrating the PI + RF controller design method. Steps 3 and 4 proceed as follows:

The general objective when designing the inner-loop RF controller is to give the same benefits expected in the linear case, namely, stabilizing and damping the system, if necessary, and reducing the sensitivity of the system to disturbances and plant nonlinearities. At the same time, we wish to design a nonlinearity to be used with the controller that will desensitize the inner loop as much as possible to different input amplitudes.

As shown in D'Azzo and Houpis (34), it is convenient to work with inverse Nyquist plots of the plant I/O model, that is, invert the SIDF frequency-response information in complex-gain form and plot the result in the complex plane. In the linear case, this allows us to study the closed-inner-loop (*CIL*) frequency response $G_{\mathrm{CIL}}(j\omega)$ in the inverse form

$$\frac{1}{G_{\mathrm{CIL}}(j\omega)} = \frac{1 + G(j\omega)H(j\omega)}{G(j\omega)} = \frac{1}{G(j\omega)} + H(j\omega)$$

where the effect of $H(j\omega)$ on $1/G_{\mathrm{CIL}}(j\omega)$ is easily determined.

The inner-loop tachometer feedback design algorithm given by D'Azzo and Houpis and referred to as Case 2 uses a construction amenable to extension to nonlinear systems. For linear systems, this algorithm is based on evaluating a tachometer gain and external gain in order to adjust the inverse Nyquist plot to be tangent to a given M circle at a selected frequency. The algorithm is extended to the nonlinear case by applying it to each SIDF model $G(j\omega, b_{\mathrm{i}})$. Then for each input amplitude b_{i} a tachometer gain $K_{\mathrm{T},i}$ and an external (to the inner loop) gain $A_{2,\mathrm{i}}$ are found. The gains $A_{2,\mathrm{i}}$ are discarded, since the external gain will be subsumed in the cascade portion of the controller that is synthesized in step 6.

The set of desired tachometer gains $K_{\mathrm{T},i}(b_{\mathrm{i}})$ is then used to synthesize the tachometer nonlinearity f_{T}. As first described in Ref. 15, these gain–amplitude data are interpreted as SIDF information for an unknown static nonlinearity. A least-squares routine is used to adjust the parameters of a general piecewise linear nonlinearity so that the SIDF of that nonlinearity fits these gain–amplitude data with minimum mean squared error; this generates the desired RF controller nonlinearity, completing this step. This process of SIDF inversion is illustrated below (Fig. 17).

The final step in the complete controller design procedure is generating the nonlinear cascade PI compensator. The general idea is to first generate SIDF I/O models for the nonlinear plant (which, in this approach, is actually the nonlinear plant with nonlinear RF) over the range of input amplitudes and frequencies of interest. This information forms a frequency-response map as a function of both input amplitude and frequency. A single nominal input amplitude is selected, b^*, and a linear compensator is found that best compensates the plant *at that amplitude*. This compensator, in series with the nonlinear plant, is used to calculate the corresponding desired open-loop I/O model $\mathrm{CG}^*(j\omega; b^*)$, the *frequency-domain objective function*. Then, at each input amplitude b_{i} a least-squares algorithm is used to adjust the parameters of a linear PI compensator, $K_{\mathrm{P},i}(b_{\mathrm{i}})$ and $K_{\mathrm{I},i}(b_{\mathrm{i}})$, to minimize the difference between the resulting frequency responses, found using the linear compensator and interpolating on the SIDF frequency- response map, and $\mathrm{CG}^*(j\omega; b^*)$, as described in Ref. 31. The nonlinear PI compensator is then obtained by synthesizing the nonlinearities f_{P} and f_{I} by SIDF inversion.

An important part of this procedure is the process called SIDF inversion, or adjusting the parameters of a general piecewise-linear nonlinearity so that the SIDF of that nonlinearity fits the gain–amplitude data with minimum mean squared error. This step is illustrated in Fig. 17, where the piecewise characteristic had two breakpoints (δ_1, δ_2) and three slopes (m_1, m_2, m_3) that were adjusted to fit the gain–amplitude data (small circles) with good accuracy.

The final validation of the design is to simulate a family of step responses for the nonlinear control system. The results for the controller from Ref. 16 [which are identical to the results obtained with a later fuzzy-logic implementation that is also based on the direct application of this SIDF approach (33)] are depicted

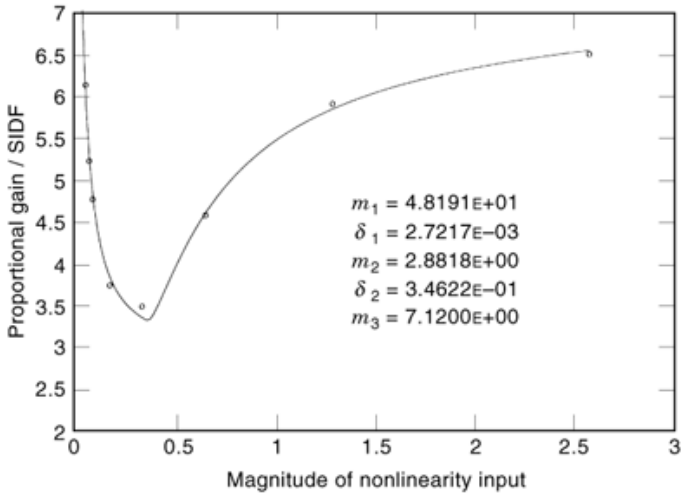


Fig. 17. Proportional nonlinearity synthesis via DF inversion.

in Fig. 18, along with similar step responses generated using the linear PI + RF controller corresponding to the frequency-domain objective function. In both response sets input amplitudes ranging from $b_1 = 0.20$ to $b_8 = 10.2$ were used, and the output normalized by dividing by b_i. Comparing the responses of the two controllers, it is evident that the SIDF design achieves significantly better performance, both in the sense of lower overshoot and less settling time and in the sense of very low sensitivity of the response over the range of input amplitudes considered. The high overshoot in the linear case is caused by integral windup for large step commands, and the long settling for small step inputs is due to stiction. The nonlinear controller greatly alleviates these problems.

Design of Autotuning Linear and Nonlinear Controllers.

Linear Autotuning Controllers. A clever procedure for the automatic tuning of proportional–integral–derivative (PID) regulators for linear or nearly linear plants has been introduced and used commercially (35). It incorporates a very simple SIDF-based method to identify key parameters in the frequency response of a plant, to serve as the basis for automatically determining the parameters of a PID controller (a process called *autotuning*). It is based on performing system identification via relay-induced oscillations. The system is connected in a feedback loop with a known relay to produce an LC; frequency-domain information about the system dynamics is derived from the LC's amplitude and frequency. With an ideal relay, the oscillation will give the critical point where the Nyquist curve intersects the negative real axis. Other points on the Nyquist curve can be explored by adding hysteresis to the relay characteristic. Linear design methods based on knowledge of part of the Nyquist curve are called *tuning rules*; the Ziegler–Nichols rules (36) are the most familiar.

To appreciate how these key parameters are identified, refer to Fig. 7: clearly, if the process is seen to be in an LC, one can readily observe the amplitude and frequency of the oscillation. Since the relay height and hysteresis are known, along with the amplitude, $N_s(a)$ can be calculated [Eq. (24)], and the point $-1/N_s(a)$ establishes a point on the Nyquist plot of the process. On changing the hysteresis the locus of $-1/N_s(a)$ will shift vertically, the LC amplitude and frequency will also change, and one can calculate another point on the Nyquist curve.

Basing identification on relay-induced oscillation has several advantages. An input signal that is near-optimal for identification is generated automatically, and the experiment is safe in the sense that it is easy to control the amplitude of the oscillation by choosing the relay height accordingly.

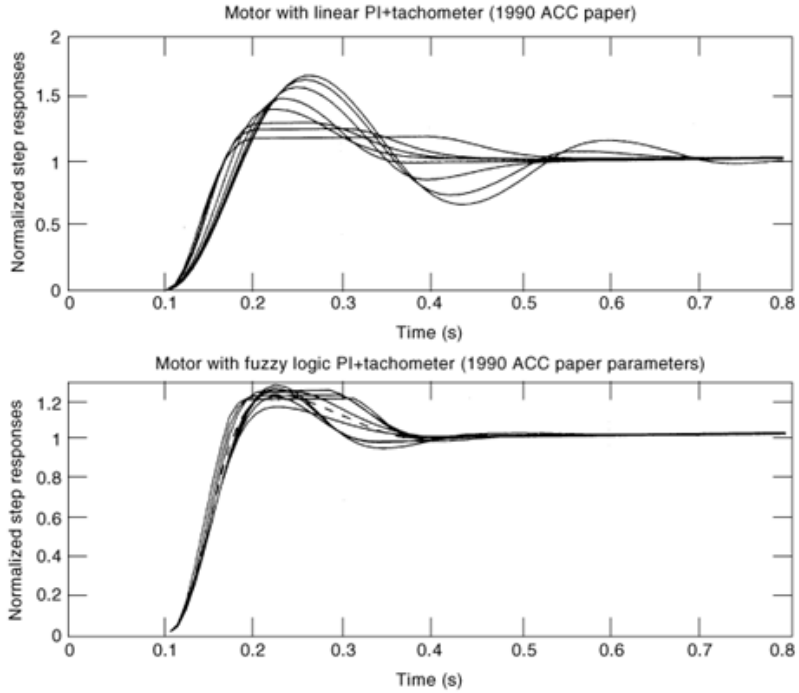


Fig. 18. Linear and nonlinear controller: step responses.

The most basic tuning rules require only one point on $G(j\omega)$, so this procedure is fast and easy to implement. More elaborate tuning rules may require more points on $G(j\omega)$, but that presents no difficulty.

Nonlinear Autotuning Controllers. This same procedure can be extended to the case of nonlinear plants quite directly. The amplitude of the periodic signal forcing the plant is determined by the relay height D. Therefore, by selecting a number of values, D_i, one may identify points on the family of frequency response curves, $G(j\omega, b_i)$. Nonlinear controllers can be synthesized from this information, using the methods outlined and illustrated in the preceding section. This technique, and a sample application, are presented in Ref. 37.

Describing Function Methods: Concluding Remarks

Describing function methods all follow the formula "assume a signal form, choose an approximation criterion, evaluate the DF $N(a)$, use this as a quasilinear gain to replace the nonlinearity with a linear term, and solve the problem using linear systems theoretic machinery." We should keep in mind that in so doing we are deciding what type of phenomena we may investigate, and thereby avoid the temptation to reach erroneous conclusions; for example, if for some values of (m, S) the RIDF matrix N_R [Eq. (39)] has "eigenvalues" that are pure imaginary, we must not jump to the conclusion that LCs may exist. We should also take care not to slip into "linear-system thinking" and read too much into a DF result—for example, $X(j\omega)$ in Eq. (6) is not exactly the Fourier transform of an eigenvector.

The DF approach has proven to be immensely powerful and successful over the 50 years since its conception, especially in engineering applications. The primary reasons for this are:

(1) Engineering applications usually are too large and/or too complicated to be amenable to exact solution methods.
(2) The ability to apply linear systems-theoretic machinery (e.g., the use of Nyquist plots to solve for LC conditions) alleviates much of the analytic burden associated with the analysis and design of nonlinear systems.
(3) The behavior of DFs (the form of amplitude sensitivity, Fig. 2) is simple to grasp intuitively, so one can even use DFs in a qualitative manner without analysis.

The techniques and examples presented in this article are intended to demonstrate these points. This material represents a very limited exposure to a vast body of work. The reference books by Gelb and Vander Velde (1) and Atherton (2) detail the first half of this corpus (in chronological terms); subsequent work by colleagues and students of these pioneers plus that of others inspired by those contributions, has produced a body of literature that is massive and of great value to the engineering profession.

BIBLIOGRAPHY

1. A. Gelb W. E. Vander Velde, *Multiple-Input Describing Functions and Nonlinear System Design*, New York: McGraw-Hill, 1968.
2. D. P. Atherton, *Nonlinear Control Engineering*, London and New York: Van Nostrand-Reinhold, full ed. 1975; student ed. 1982.
3. I. E. Kazakov, Generalization of the method of statistical linearization to multi-dimensional systems, *Avlom. Telemekh.*, **26**: 1210–1215, 1965.
4. A. Gelb R. S. Warren, Direct statistical analysis of nonlinear systems: CADET, *AIAA J.*, **11** (5): 689–694, 1973.
5. J. H. Taylor, *An Algorithmic State-Space / Describing Function Technique for Limit Cycle Analysis*, TIM-612-1, Reading, MA: Office of Naval Research, The Analytic Sciences Corporation (*TASC*), 1975; presented as: A new algorithmic limit cycle analysis method for multivariable systems, *IFAC Symposium on Multivariable Technological Systems*, Fredericton, NB, Canada, 1977.
6. J. H. Taylor, A general limit cycle analysis method for multivariable systems, *Engineering Foundation Conference in New Approaches to Nonlinear Problems in Dynamics*, Asilomar, CA, 1979; published as a chapter in P. J. Holmes (ed.), *New Approaches to Nonlinear Problems in Dynamics*, Philadelphia: *SIAM* (Society of Industrial and Applied Mathematics), pp. 521–529, 1980.
7. D. N. Hannebrink et al., Influence of axle load, track gauge, and wheel profile on rail vehicle hunting, *Trans. ASME J. Eng. Ind.*, pp. 186–195, 1977.
8. R. V. Ramnath, J. K. Hedrick, H. M. Paynter, ed, *Nonlinear System Analysis and Synthesis*, New York: ASME Book, 1980, Vol. 2, Chs. 7, 9, 13, 16.
9. T. W. Körner, *Fourier Analysis*, Cambridge, UK: Cambridge University Press, 1988.
10. J. E. Gibson, *Nonlinear Automatic Control*, New York: McGraw-Hill, 1963.
11. J. H. Taylor D. Kebede, Modeling and simulation of hybrid systems, *Proc. IEEE Conf. Decis. Control*, New Orleans, LA, pp. 2685–2687, 1995. MATLAB-based software is available on the author's Web site, http://www.ee.unb.ca/jtaylor (including documentation).
12. Ya. Z. Tsypkin, On the determination of steady-state oscillations of on–off feedback systems, *IRE Trans. Circuit Theory*, **CT-9** (3), 1962; original citation: Ob ustoichivosti periodicheskikh rezhimov v relejnykh systemakh avtomaticheskogo regulirovanija, *Avtom. Telemekh.* **14** (5), 1953.
13. O. P. McNamara D. P. Atherton, Limit cycle prediction in free structured nonlinear systems, *Proc. IFAC World Congress*, Munich, Germany, 1987.
14. J. H. Taylor B. H. Wilson, A frequency domain model-order-deduction algorithm for nonlinear systems, *Proc. IEEE Conf. Control Appl.*, Albany, NY, pp. 1053–1058, 1995.
15. J. H. Taylor, A systematic nonlinear controller design approach based on quasilinear system models, *Proc. Am. Control Conf.*, San Francisco, pp. 141–145, 1983.

16. J. H. Taylor J. R. O'Donnell, Synthesis of nonlinear controllers with rate feedback via SIDF methods, *Proc. Am. Control Conf.*, San Diego, CA, pp. 2217–2222, 1990.
17. J. H. Taylor J. Lu, Computer-aided control engineering environment for the synthesis of nonlinear control systems, *Proc. Am. Control Conf.*, San Francisco, pp. 2557–2561, 1993.
18. K. S. Narendra J. H. Taylor, *Frequency Domain Criteria for Absolute Stability*, New York: Academic Press, 1973.
19. V. M. Popov, Nouveaux critériums de stabilité pour les systèmes automatiques nonlinéaries, *Rev. Electrotech. Energ.*, (Romania), **5** (1), 1960.
20. J. H. Taylor C. Chan, MATLAB tools for linear and nonlinear system stability theorem implementation, *Proc. 6th IEEE Conf. Control Appl.*, Hartford CT, pp. 42–47, 1997. MATLAB-based software is available on the author's Web site, http://www.ee.unb.ca/jtaylor (including documentation).
21. Y. S. Cho K. S. Narendra, An off-axis circle criterion for the stability of feedback systems with a monotonic nonlinearity, *IEEE Trans. Autom. Control*, **AC-13**, 413–416, 1968.
22. D. P. Atherton, *Stability of Nonlinear Systems*, Chichester: Research Studies Press (Wiley), 1981.
23. J. H. Taylor et al., Covariance analysis of nonlinear stochastic systems via statistical linearization, in R. V. Ramnath, J. K. Hedrick, and H. M. Paynter (eds.), *Nonlinear System Analysis and Synthesis*, New York: ASME Book, Vol. 2, pp. 211–226, 1980.
24. A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, New York: Academic Press, 1970.
25. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York: McGraw-Hill, 1965.
26. M. Landau C. T. Leondes, Volterra series synthesis of nonlinear stochastic tracking systems, *IEEE Trans. Aerosp. Electron. Syst.*, **AES-11**: 245–265, 1975.
27. J. H. Taylor, Comments on "Volterra series synthesis of nonlinear stochastic tracking systems," *IEEE Trans. Aerosp. Electron. Syst.*, **AES-14**: 390–393, 1978.
28. J. H. Taylor, Statistical performance analysis of nonlinear stochastic systems by the Monte Carlo method (invited paper), *Trans. Math. Comput. Simul.*, **23**: 21–33, 1981.
29. J. H. Taylor, Applications of a general limit cycle analysis method for multi-variable systems, in R. V. Ramnath, J. K. Hedrick, and H. M. Paynter (eds.), *Nonlinear System Analysis and Synthesis*, New York: ASME Book, Vol. 2, pp. 143–159, 1980.
30. P. J. Holmes J. E. Marsden, Bifurcations to divergence and flutter in flow induced oscillations—an infinite dimensional analysis, *Automatica*, **14**: 367–384, 1978.
31. J. H. Taylor K. L. Strobel, Nonlinear compensator synthesis via sinusoidal-input describing functions, *Proc. Am. Control Conf.*, Boston, pp. 1242–1247, 1985.
32. J. H. Taylor L. Sheng, Recursive optimization procedure for fuzzy-logic controller synthesis, *Proc. Am. Control Conf.*, Philadelphia, pp. 2286–2291, 1998.
33. J. H. Taylor L. Sheng, Fuzzy-logic controller synthesis for electro-mechanical systems with nonlinear friction, *Proc. IEEE Conf. Control App.*, Dearborn, MI, pp. 820–826, 1996.
34. J. J. D'Azzo C. H. Houpis, *Feedback Control System Analysis and Synthesis*, New York: McGraw-Hill, 1960.
35. K. J. Åström T. Hägglund, Automatic tuning of simple regulators for phase and amplitude margin specifications, *Proc. IFAC Workshop on Adaptive Systems for Control and Signal Processing*, San Francisco, 1983.
36. K. J. Åström B. Wittenmark, *Adaptive Control*, 2nd ed., Reading, MA: Addison-Wesley, 1995.
37. J. H. Taylor K. J. Åström, A nonlinear PID autotuning algorithm, *Proc. Am. Control Conf.*, Seattle, WA, pp. 2118–2123, 1986.

JAMES H. TAYLOR
University of New Brunswick

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Duality-Mathematics
Standard Article
David Yang Gao[1]
[1]Department of Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, Virginia
Copyright © 2007 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2412.pub2
Article Online Posting Date: November 16, 2007
Abstract | Full Text: HTML PDF (400K)

Abstract

The sections in this article are

Introduction

Framework in Quadratic Minimization

Canonical Lagrangian Duality Theory

Primal-Dual Solutions and Central Path

Canonical Duality Theory in Nonconvex Systems

Application in Nonconvex Variational Problem

Applications in Global Optimization

Conclusions

Keywords: duality; complementary; configuration variables; geometric operator; source variable; intermediate variable

| | |

Copyright © 1999-2008 All Rights Reserved.

DUALITY, MATHEMATICS

The term *duality* used in our daily life means the sort of harmony of two opposite or *complementary* parts by which they integrate into a whole. Inner beauty in natural phenomena is bound up with duality, which has always been a rich source of inspiration in human knowledge through the centuries. The theory of duality is a vast subject, significant in art and natural science. Mathematics lies at its root. By using abstract languages, a common mathematical structure can be found in many physical theories. This structure is independent of the physical contents of the system and exists in wider classes of problems in engineering and sciences (see Ref. 1).

According to Tonti (2), for every physical theory we can identify (a) some *configuration variables* that describe the state of the system and (b) some *source variables* that describe the source of the phenomenon. The displacement vector in continuum mechanics and the scalar potential in electrostatics are examples of configuration variables. Forces and charges are examples of source variables. Besides these two types of quantities, there are also some paired (i.e., one-to-one) *intermediate variables* that describe the internal (or constitutive) properties of the system, such as velocity and momentum in dynamics, electrical field intensity and the flux density in electrostatics, and the two electromagnetic tensors in electromagnetism.

Let $\mathcal{U}$ and $\mathcal{U}^*$ be, respectively, the real vector spaces of configuration variables and source variables, and let $\mathcal{V}$ and $\mathcal{V}^*$ denote the paired intermediate variable spaces. By introducing a so-called *geometric operator* $\Lambda : \mathcal{U} \to \mathcal{V}$ and an equilibrium operator $B : \mathcal{V}^* \to \mathcal{U}^*$, such that the duality relation between $\mathcal{V}$ and $\mathcal{V}^*$ is a one-to-one mapping, the *primal system* $\mathcal{S}_p := \{\mathcal{U}, \mathcal{V}; \Lambda\}$ and the *dual system* $\mathcal{S}_d := \{\mathcal{U}^*, \mathcal{V}^*; B\}$ are linked into a whole system $\mathcal{S} = \mathcal{S}_p \cup \mathcal{S}_d$. The system is called *geometrically linear* if Λ is linear. In this case, B is the adjoint operator of Λ. If Λ is an $m \times n$ matrix, then $\mathcal{S}$ is a finite-dimensional *algebraic system.* Optimization in such systems is known as *mathematical programming.* If Λ is a continuous (partial) differential operator, then $\mathcal{S}$ is an infinite-dimensional (*partial*) *differential system,* and optimization problems fall into the *calculus of variations.* It is shown in Ref. 3 that under certain conditions, if there is a theorem in the primal system $\mathcal{S}_p$, then in the dual system $\mathcal{S}_d$ there exists a complementary theorem and vice versa. If there is a theorem defined on the whole system $\mathcal{S}$, then exchanging the dual elements in this theorem leads to another parallel theorem. Generally speaking, the *theory of duality* is the study of the intrinsic relations between the primal system $\mathcal{S}_p$ and the dual system $\mathcal{S}_d$.

In the theory of optimization, let $P : \mathcal{U} \to \mathbb{R}$ and $P^* : \mathcal{V}^* \to \mathbb{R}$ be real-valued functions. If $P(\boldsymbol{u}) \geq P^*(\boldsymbol{v}^*)$ for all vectors $(\boldsymbol{u}, \boldsymbol{v}^*)$ in the *Cartesian product space* $\mathcal{U} \times \mathcal{V}^*$, then an infimum of P and a supremum of P^* exist and $\inf P(\boldsymbol{u}) \geq \sup P^*(\boldsymbol{v}^*)$. Under certain conditions we have $\inf P(\boldsymbol{u}) = \sup P^*(\boldsymbol{v}^*)$. A statement of this type is called a *duality theorem.*

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

Let $L(\boldsymbol{u}, \boldsymbol{v}^*): \mathscr{U} \times \mathscr{V}^* \to \mathbb{R}$ be a so-called *Lagrangian form.* Under certain conditions we have $\inf_{\boldsymbol{u}} \sup_{\boldsymbol{v}^*} L(\boldsymbol{u}, \boldsymbol{v}^*) = \sup_{\boldsymbol{v}^*} \inf_{\boldsymbol{u}} L(\boldsymbol{u}, \boldsymbol{v}^*)$. Such a statement is called a *minimax theorem.* In convex optimization, the minimax theorem is connected to a *saddle-point theorem.* The main purpose of the theory of duality in mathematical optimization is to make inquiries about a corresponding pair of optimization problems, namely, (a) the primal problem to find $\overline{\boldsymbol{u}}$ such that $P(\overline{\boldsymbol{u}}) = \inf_{\boldsymbol{u}} P(\boldsymbol{u})$ and (b) the dual problem to find $\overline{\boldsymbol{v}}^*$ such that $P^*(\overline{\boldsymbol{v}}^*) = \sup_{\boldsymbol{v}^*} P^*(\boldsymbol{v}^*)$ and to discover relations between corresponding duality, minimax, and theorems of critical points. In numerical analysis, the primal problem provides only upper-bound approaches to the solution. However, the dual problem will give a lower bound of solutions. The numerical methods to find the *primal–dual solution* $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ in each iteration are known as *primal–dual methods.* In finite element analysis of boundary value problems, such methods as *mixed / hybrid methods* have been studied extensively by engineers for more than 30 years. In the past decade, *primal–dual algorithms* have emerged as the most important and useful algorithms for mathematical programming (4).

Duality in natural science is amazingly beautiful. It has excellent theoretical properties, powerful practical applications, and pleasing relationships with the existing fundamental theories. In geometrically linear systems, the common mathematical structure and theorems take particularly symmetric forms. The duality theory has been well studied for both convex problems (see Refs. 5–8) and nonconvex problems (see Refs. 9 and 10). However, in *geometrically nonlinear systems,* where Λ is a nonlinear operator, such symmetry is broken. The duality theory in these systems was studied in Ref. 11. An interesting *triality theorem* in nonconvex systems has been discovered recently in Refs. 12 and 13, which can be used either to solve some nonlinear variational problems or to develop algorithms for numerical solutions in nonconvex, *nonsmooth optimization* (see Ref. 3).

FRAMEWORK AND CANONIC EQUATIONS

Let $\mathscr{U}$, $\mathscr{U}^*$ and $\mathscr{V}$, $\mathscr{V}^*$ be two pairs of *real vector spaces,* finite- or infinite-dimensional, and let $(*, *): \mathscr{U} \times \mathscr{U}^* \to \mathbb{R}$ and $\langle *, * \rangle: \mathscr{V} \times \mathscr{V}^* \to \mathbb{R}$ be certain *bilinear forms.* We say that these two bilinear forms put the paired spaces $\mathscr{U}$, $\mathscr{U}^*$ and $\mathscr{V}$, $\mathscr{V}^*$ *in duality,* respectively. Let the *geometric operator* Λ be a continuous linear transformation from $\mathscr{U}$ to $\mathscr{V}$. The *equilibrium operator* B in geometrically linear system is simply the adjoint operator $\Lambda^*: \mathscr{V}^* \to \mathscr{U}^*$ defined by

$$\langle \Lambda \boldsymbol{u}, \boldsymbol{v}^* \rangle = (\boldsymbol{u}, \Lambda^* \boldsymbol{v}^*) \qquad \forall \boldsymbol{u} \in \mathscr{U}, \boldsymbol{v}^* \in \mathscr{V}^* \tag{1}$$

Thus the two paired dual spaces $\mathscr{U}$, $\mathscr{U}^*$ and $\mathscr{V}$, $\mathscr{V}^*$ are linked, respectively, by a so-called *geometrical* (*or definition*) *equation:*

$$\boldsymbol{v} = \Lambda \boldsymbol{u} \tag{2}$$

and an *equilibrium equation:*

$$\boldsymbol{u}^* = \Lambda^* \boldsymbol{v}^* \tag{3}$$

In calculus of variations, if Λ is a gradient-like operator, its adjoint Λ^* should be a divergence-like operator, Eq. (1) is then the well-known Gauss–Green formula.

The readers who are interested primarily in the finite-dimensional case will not need knowledge of *convex analysis* in what follows. Instead, they can simply interpret $\mathscr{U} = \mathscr{U}^* = \mathbb{R}^n$, $\mathscr{V} = \mathscr{V}^* = \mathbb{R}^m$, with $(\boldsymbol{u}, \boldsymbol{u}^*)$ and $\langle \boldsymbol{v}, \boldsymbol{v}^* \rangle$ as the ordinary inner products on the Euclidean spaces $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. In this case, we can of course identify Λ with a certain $m \times n$ matrix $A = \{a_{ij}\}$, and

$$\langle A\boldsymbol{u}, \boldsymbol{v}^* \rangle = \sum_{j=1}^{m} \sum_{i=1}^{n} a_{ji} u_i v_j^* = \sum_{i=1}^{n} u_i \left(\sum_{j=1}^{m} a_{ji} v_j^* \right) = (\boldsymbol{u}, A^* \boldsymbol{v}^*)$$

So the adjoint of $\Lambda = A$ is a transpose matrix $A^* = A^T$.

A subset $\mathscr{C} \subset \mathscr{U}$ is said to be a *convex set* if for any given $\theta \in [0, 1]$, we have

$$\theta \boldsymbol{u}_1 + (1 - \theta)\boldsymbol{u}_2 \in \mathscr{C} \qquad \forall \boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathscr{C}$$

By a *convex function* $F: \mathscr{U} \to \overleftrightarrow{\mathbb{R}} := [-\infty, +\infty]$ we shall mean that for any given $\theta \in [0, 1]$, we obtain

$$F(\theta \boldsymbol{u}_1 + (1 - \theta)\boldsymbol{u}_2) \le \theta F(\boldsymbol{u}_1) + (1 - \theta) F(\boldsymbol{u}_2) \qquad \forall \boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathscr{U} \tag{4}$$

F is *strictly convex* if the inequality is strict. The *indicator function* of a subset $\mathscr{C} \subset \mathscr{U}$ is defined by

$$\Psi_C(\boldsymbol{u}) = \begin{cases} 0 & \text{if } \boldsymbol{u} \in \mathscr{C} \\ +\infty & \text{otherwise} \end{cases} \tag{5}$$

which plays an important role in constrained optimization. This is a convex function if and only if $\mathscr{C}$ is convex.

A function F on $\mathscr{U}$ is said to be *proper* if $F(\boldsymbol{u}) > -\infty \ \forall \boldsymbol{u} \in \mathscr{U}$ and $F(\boldsymbol{u}) < +\infty$ for at least one $\boldsymbol{u}$. Conversely, given a convex function F defined on a nonempty convex set $\mathscr{C}$, one can set $\overline{F}(\boldsymbol{u}) = F(\boldsymbol{u}) + \Psi_{\mathscr{C}}(\boldsymbol{u})$. In this way one can relax the constraint $\boldsymbol{u} \in \mathscr{C}$ on F to get a proper function $F(\boldsymbol{u}) + \Psi_{\mathscr{C}}(\boldsymbol{u})$ on the whole space $\mathscr{U}$.

A function F on $\mathscr{U}$ is *lower semicontinuous* (l.s.c.) if

$$\lim_{\boldsymbol{u}_n \to \boldsymbol{u}} \inf F(\boldsymbol{u}_n) \ge F(\boldsymbol{u}) \qquad \forall \boldsymbol{u} \in \mathscr{U} \tag{6}$$

So $\Psi_{\mathscr{C}}(\boldsymbol{u})$ is l.s.c. if and only if $\mathscr{C}$ is closed. A function F is said to be *concave, upper semicontinuous* (u.s.c.) if $-F$ is convex, l.s.c. The theory of concave functions thus parallels the theory of convex functions, with only the obvious and dual changes.

If F is finite on $\mathscr{C}$, the *Gâteaux variation* of F at $\boldsymbol{u} \in \mathscr{C}$ in the direction $\boldsymbol{v}$ is defined as

$$\delta F(\boldsymbol{u}; \boldsymbol{v}) = \lim_{\theta \to 0^+} \frac{F(\boldsymbol{u} + \theta \boldsymbol{v}) - F(\boldsymbol{u})}{\theta} \tag{7}$$

$F(\boldsymbol{u})$ is said to be *Gâteaux* (*or G-*) *differentiable* at $\boldsymbol{u}$ if $\delta F(\boldsymbol{u}; \boldsymbol{v}) = (DF(\boldsymbol{u}), \boldsymbol{v})$, where $DF: \mathscr{C} \subset \mathscr{U} \to \mathscr{U}^*$ is called the *Gâteaux derivative* of F. In finite-dimensional space, the Gâteaux variation is simply the directional derivative, and $DF = \nabla F$.

Let F and V be two real-valued functions. Throughout this article we assume that F and V are (a) convex or concave and (b) G-differentiable on the convex sets $\mathscr{C} \subset \mathscr{U}$ and $\mathscr{D} \subset \mathscr{V}$, respectively. Then the two *duality equations* between the paired spaces $\mathscr{U}$, $\mathscr{U}^*$ and $\mathscr{V}$, $\mathscr{V}^*$ can be given by

$$\boldsymbol{u}^* = DF(\boldsymbol{u}), \qquad \boldsymbol{v}^* = DV(\boldsymbol{v}) \tag{8}$$

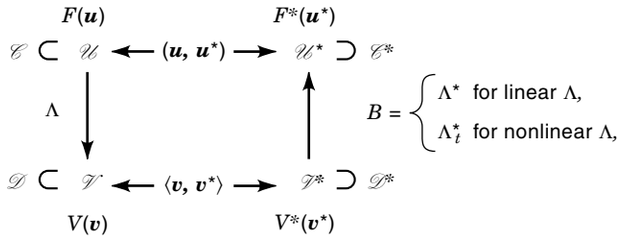


Figure 1. Framework in fully nonlinear systems.

In mathematical physics, the duality equation $\boldsymbol{v}^* = DV(\boldsymbol{v})$ is known as the *constitutive equation.* However, the duality equation $\boldsymbol{u}^* = DF(\boldsymbol{u})$ usually gives *natural boundary conditions* in variational boundary value problems.

Let $\mathscr{U}_a = \{\boldsymbol{u} \in \mathscr{U} |\ \boldsymbol{u} \in \mathscr{C},\ \Lambda\boldsymbol{u} \in \mathscr{D}\}$ be a so-called *feasible set.* On $\mathscr{U}_a$, the three types of canonical equations, Eqs. (2), (3) and (8), can be written in a so-called *fundamental equation:*

$$\Lambda^* DV(\Lambda\boldsymbol{u}) = DF(\boldsymbol{u}) \tag{9}$$

The system is called *physically linear* if both duality equations are linear. The system is called *geometrically linear* if the geometric operator $\Lambda : \mathscr{U} \to \mathscr{V}$ is linear. By the term *linear system* we mean that it is both geometrically and physically linear. In this case, if, for a given $\overline{\boldsymbol{u}}^* \in \mathscr{U}^*$, $F(\boldsymbol{u}) = (\boldsymbol{u}, \overline{\boldsymbol{u}}^*)$ is linear and $V(\boldsymbol{v}) = \frac{1}{2}\langle C\boldsymbol{v}, \boldsymbol{v}\rangle$ is quadratic, where $C : \mathscr{V} \to \mathscr{V}^*$ is a linear operator, then the fundamental equation can be written as

$$\Lambda^* C\Lambda\boldsymbol{u} = \overline{\boldsymbol{u}}^* \tag{10}$$

If C is symmetric, then the operator $K = \Lambda^* C\Lambda : \mathscr{U} \to \mathscr{U}^*$ is *self-adjoint* $K = K^*$. In *partial differential systems, K* is an *elliptic operator* if C is either *positive or negative definite,* whereas K is hyperbolic if C is *nonsingular and indefinite.*

The common mathematical structure in geometrically linear systems is shown in Fig. 1. In the textbook by Strang (1), this nice symmetrical structure can be seen from continuous theories to discrete systems. However, the symmetry in this structure is broken in geometrically nonlinear systems where Λ is a nonlinear operator. If we assume that $\boldsymbol{v} = \Lambda(\boldsymbol{u})$ is Gâteaux differentiable, then it can be split as

$$\Lambda = \Lambda_t + \Lambda_n \tag{11}$$

where Λ_t is the G-derivative of Λ and $\Lambda_n = \Lambda - \Lambda_t$, both of them depending on $\boldsymbol{u}$ (see Ref. 11). For a given $\overline{\boldsymbol{u}}^* \in \mathscr{U}^*$, the *virtual work principle* gives

$$\langle \delta\boldsymbol{v}(\overline{\boldsymbol{u}}; \boldsymbol{u}), \boldsymbol{v}^*\rangle = (\boldsymbol{u}, \Lambda_t^*(\overline{\boldsymbol{u}})\boldsymbol{v}^*) = (\boldsymbol{u}, \overline{\boldsymbol{u}}^*) \qquad \forall \boldsymbol{u} \in \mathscr{U} \tag{12}$$

In this case, the equilibrium operator $B = \Lambda_t^* : \mathscr{V}^* \to \mathscr{U}^*$ is the adjoint of Λ_t, which depends on the configuration variable. Then the equilibrium equation in geometrically nonlinear systems should be

$$\Lambda_t^*(\overline{\boldsymbol{u}})\boldsymbol{v}^* = \overline{\boldsymbol{u}}^* \tag{13}$$

The relation between the two bilinear forms is then

$$\langle \Lambda(\boldsymbol{u}), \boldsymbol{v}^*\rangle = (\boldsymbol{u}, \Lambda_t^*(\boldsymbol{u})\boldsymbol{v}^*) - G(\boldsymbol{u}, \boldsymbol{v}^*) \tag{14}$$

where $G(\boldsymbol{u}, \boldsymbol{v}^*)$ is the so-called *complementary gap function* introduced in Ref. 11:

$$G(\boldsymbol{u}, \boldsymbol{v}^*) = \langle -\Lambda_n(\boldsymbol{u})\boldsymbol{u}, \boldsymbol{v}^*\rangle \tag{15}$$

In geometrically nonlinear systems, this gap function recovers the duality theorems in convex optimization and plays an important role in nonconvex problems.

Example 1. Let us consider a mixed boundary value problem in electrostatics:

$$\operatorname{div}[\epsilon \operatorname{grad}\phi(x)] + \rho(x) = 0 \qquad \forall x \in \Omega \subset \mathbb{R}^n \tag{16}$$

$$\phi(x) = 0 \qquad \forall x \in \Gamma_1$$

$$\boldsymbol{n} \cdot \operatorname{grad}\phi(x) = D_n \qquad \forall x \in \Gamma_2 \tag{17}$$

$$\Gamma_1 \cup \Gamma_2 = \partial\Omega$$

The configuration $\boldsymbol{u}$ is the *electrostatic potential* $\phi(x)$. The source variable is the *charge density* $\overline{\boldsymbol{u}}^* = \rho(x)$ in Ω and electric flux $\overline{\boldsymbol{u}}^* = D_n$ on Γ_2. ϵ is the *dielectric constant.* $\boldsymbol{n} \in \mathbb{R}^n$ is a unit vector normal to the boundary. Let $\Lambda = -\text{grad}$, and thus $\boldsymbol{v} = -\text{grad}\ \phi$ is the *electric field intensity,* denoted by $\boldsymbol{E}$. Let $\mathscr{D} = \mathscr{H}(\Omega; \mathbb{R}^n)$ be a *Hilbert space* with domain Ω and range $\mathbb{R}^n$, $\mathscr{C} = \{\phi \in \mathscr{H}(\Omega; \mathbb{R}) |\ \phi(x) = 0\ \forall x \in \Gamma_1\}$, and

$$F(\phi) = \int_\Omega \rho\phi\, d\Omega + \int_{\Gamma_2} \phi D_n\, d\Gamma - \Psi_{\mathscr{C}}(\phi) \tag{18}$$

$$V(\boldsymbol{E}) = \int_\Omega \frac{1}{2}\epsilon \boldsymbol{E}^T\boldsymbol{E}\, d\Omega + \Psi_{\mathscr{D}}(\boldsymbol{E}) \tag{19}$$

So on $\mathscr{C}$ and $\mathscr{D}$, F and V are finite, G-differentiable. Thus $\boldsymbol{D} = DV(\boldsymbol{E}) = \epsilon\boldsymbol{E}$ is the *electric flux density,* $\boldsymbol{u}^*(x) = DF(\phi) = \{\rho(x)\ \forall x \in \Omega, D_n\ \forall x \in \Gamma_2\}$. By the Gauss–Green theorem,

$$\begin{aligned}\langle \Lambda\phi, \boldsymbol{D}\rangle &= \int_\Omega (-\operatorname{grad}\phi) \cdot \boldsymbol{D}\, d\Omega \\ &= \int_\Omega \phi(\nabla \cdot \boldsymbol{D})\, d\Omega - \oint_{\partial\Omega} \phi\boldsymbol{n} \cdot \boldsymbol{D}\, d\Gamma = (\phi, \Lambda^*\boldsymbol{D})\end{aligned}$$

Hence, the adjoint operator Λ^* and the abstract equilibrium equation (3) are

$$\boldsymbol{u}^* = \Lambda^*\boldsymbol{D} = \begin{cases} \operatorname{div}\boldsymbol{D} = \rho & \text{in } \Omega \\ -\boldsymbol{n} \cdot \boldsymbol{D} = D_n & \text{on } \Gamma_2 \end{cases} \tag{20}$$

The fundamental equation, Eq. (10), in this problem is a Poisson equation and $K = \Lambda^* C\Lambda = -\epsilon\Delta$ is a Laplace operator for constant $\epsilon \in \mathbb{R}$.

FENCHEL–ROCKAFELLAR DUALITY

For a given function $V : \mathscr{V} \to \overleftrightarrow{\mathbb{R}}$, its *conjugate function* is defined by the following *Fenchel transformation:*

$$V^*(\boldsymbol{v}^*) = \sup_{\boldsymbol{v} \in \mathscr{V}} \{\langle \boldsymbol{v}, \boldsymbol{v}^*\rangle - V(\boldsymbol{v})\} \tag{21}$$

which is always l.s.c. and convex on $\mathscr{V}^*$. The following *Fenchel–Young inequality* holds:

$$V(\boldsymbol{v}) \geq \langle \boldsymbol{v}, \boldsymbol{v}^*\rangle - V^*(\boldsymbol{v}^*) \qquad \forall \boldsymbol{v} \in \mathscr{V}, \boldsymbol{v}^* \in \mathscr{V}^* \tag{22}$$

If V is strictly convex, G-differentiable on a convex set $\mathscr{D} \subset \mathscr{V}$, then Eq. (21) is the classical *Legendre transformation,* and the following relations are equivalent to each other:

$$\boldsymbol{v}^* = DV(\boldsymbol{v}) \Leftrightarrow \boldsymbol{v} = DV^*(\boldsymbol{v}^*) \Leftrightarrow V(\boldsymbol{v}) + V^*(\boldsymbol{v}^*) = (\boldsymbol{v}, \boldsymbol{v}^*) \tag{23}$$

In this section we assume that

$$(A1) \qquad \begin{cases} V : \mathscr{V} \to \vec{\mathbb{R}} := (-\infty, +\infty] \text{ is proper, convex and l.s.c.} \\ F : \mathscr{U} \to \overleftarrow{\mathbb{R}} := [-\infty, +\infty) \text{ is proper, concave and u.s.c.} \end{cases} \tag{24}$$

The conjugate function of a concave function F is defined by

$$F^*(\boldsymbol{u}^*) = \inf_{\boldsymbol{u} \in \mathscr{U}} \{(\boldsymbol{u}, \boldsymbol{u}^*) - F(\boldsymbol{u})\} \tag{25}$$

Let $\mathscr{C} \subset \mathscr{U}$ be a nonempty convex set on which F is finite, G-differentiable, and define $\mathscr{C}^*$, $\mathscr{D}$, and $\mathscr{D}^*$ similarly for F^*, V, and V^*. Then on $\mathscr{C}^*$ and $\mathscr{D}^*$, the duality equations are invertible and

$$\boldsymbol{u} = DF^*(\boldsymbol{u}^*), \qquad \boldsymbol{v} = DV^*(\boldsymbol{v}^*) \tag{26}$$

Two extremum problems associated with the fundamental equation, Eq. (9), are

$$(\mathscr{P}_{\inf}) \quad \text{minimize} \qquad P(\boldsymbol{u}) = V(\Lambda \boldsymbol{u}) - F(\boldsymbol{u}) \qquad \forall \boldsymbol{u} \in \mathscr{U} \tag{27}$$

$$(\mathscr{P}^*_{\sup}) \quad \text{maximize} \qquad P^*(\boldsymbol{v}^*) = F^*(\Lambda^* \boldsymbol{v}^*) - V^*(\boldsymbol{v}^*) \qquad \forall \boldsymbol{v}^* \in \mathscr{V}^* \tag{28}$$

Note that $P : \mathscr{U} \to \vec{\mathbb{R}}$ is l.s.c., convex. It is finite at $\boldsymbol{u}$ if and only if the following *implicit constraint* of $(\mathscr{P}_{\inf})$ is satisfied:

$$\boldsymbol{u} \in \mathscr{U}_a := \{\boldsymbol{u} \in \mathscr{U} | \boldsymbol{u} \in \mathscr{C}, \quad \Lambda \boldsymbol{u} \in \mathscr{D}\} \tag{29}$$

A vector $\overline{\boldsymbol{u}} \in \mathscr{U}_a$ is called an *optimal solution* (*or minimizer*) to $(\mathscr{P}_{\inf})$ if the infimum is achieved at $\overline{\boldsymbol{u}}$ and is not $+\infty$. We write $P(\overline{\boldsymbol{u}}) = \min_{\boldsymbol{u}} P(\overline{\boldsymbol{u}})$. Similarly, the condition

$$\boldsymbol{v}^* \in \mathscr{V}^*_a := \{\boldsymbol{v}^* \in \mathscr{V}^* | \boldsymbol{v}^* \in \mathscr{D}, \quad \Lambda^* \boldsymbol{v}^* \in \mathscr{C}^*\} \tag{30}$$

is called the implicit constraint of $(\mathscr{P}^*_{\sup})$. A vector $\overline{\boldsymbol{v}}^* \in \mathscr{V}^*_a$ is a *dual optimal solution* (*or maximizer*) to $(\mathscr{P}^*_{\sup})$ if the supremum in $(\mathscr{P}^*_{\sup})$ is achieved at $\overline{\boldsymbol{v}}^*$ and is not $-\infty$. We write $P^*(\overline{\boldsymbol{v}}^*) = \max_{\boldsymbol{v}^*} P^*(\boldsymbol{v}^*)$.

For any given F and V, we always have

$$\inf P(\boldsymbol{u}) \geq \sup P^*(\boldsymbol{v}^*) \qquad \forall \boldsymbol{u} \in \mathscr{U}, \boldsymbol{v}^* \in \mathscr{V}^* \tag{31}$$

The difference $\inf P - \sup P^*$ is the so-called *duality gap.* The duality gap is zero if P is convex.

A vector $\overline{\boldsymbol{u}} \in \mathscr{U}_a$ is called a *critical point* of P if P is G-differentiable at $\overline{\boldsymbol{u}}$ and $DP(\overline{\boldsymbol{u}}) = 0$, which gives the *Euler–Lagrange equation* of $(\mathscr{P}_{\inf})$:

$$\Lambda^* DV(\Lambda \overline{\boldsymbol{u}}) - DF(\overline{\boldsymbol{u}}) = 0 \tag{32}$$

If $\mathscr{U}_a$ is an *open set,* the critical point $\overline{\boldsymbol{u}}$ should be a *global minimizer* of the convex function P on $\mathscr{U}_a$. Similarly, the critical condition $DP^*(\overline{\boldsymbol{v}}^*) = 0$ gives the *dual Euler–Lagrange equation* of $(\mathscr{P}^*_{\sup})$:

$$\Lambda DF^*(\Lambda^* \overline{\boldsymbol{v}}^*) - DV^*(\overline{\boldsymbol{v}}^*) = 0 \tag{33}$$

If $\mathscr{V}^*_a$ is an open set, $\overline{\boldsymbol{v}}^*$ should be a *global maximizer* of the concave function P^* on $\mathscr{V}^*_a$.

We say that $(\mathscr{P}_{\inf})$ is *stable* if there exists at least one vector $\boldsymbol{u}_0 \in \mathscr{U}$ such that F is finite at $\boldsymbol{u}_0$ and $V(\boldsymbol{v})$ is finite and continuous at $\boldsymbol{v} = \Lambda \boldsymbol{u}_0$.

Strong Duality Theorem 1. $(\mathscr{P}_{\inf})$ is stable if and only if $(\mathscr{P}^*_{\sup})$ has at least one solution and

$$\inf P = \max P^* \tag{34}$$

Dually, $(\mathscr{P}^*_{\sup})$ is stable if and only if $(\mathscr{P}_{\inf})$ has at least one solution and

$$\min P = \sup P^* \tag{35}$$

If $(\mathscr{P}_{\inf})$ and $(\mathscr{P}^*_{\sup})$ are both stable, then both have solutions and

$$+\infty > \min P = \max P^* > -\infty \tag{36}$$

This theorem shows that as long as the primal problem is stable, the dual problem is sure to have at least one solution. However, the existence conditions for the primal solution are stronger.

Existence and Uniqueness Theorem. Let $\mathscr{U}$ be a reflexive (i.e., $\mathscr{U} = \mathscr{U}^{**}$) Banach space with norm $\| \|$. We assume that the feasible set $\mathscr{U}_a \subset \mathscr{U}$ is a nonempty closed convex subset and conditions in (A1) hold. If $\mathscr{C}$ is bounded, or if P is coercive over $\mathscr{C}$, i.e. if

$$\lim P(\boldsymbol{u}) = +\infty \qquad \forall \boldsymbol{u} \in \mathscr{C}, \|\boldsymbol{u}\| \to \infty$$

Then the problem $(\mathscr{P}_{\inf})$ has at least one minimizer. The minimizer is unique if P is strictly convex over $\mathscr{C}$.

All finite-dimensional spaces are reflexive. But some infinite-dimensional vector spaces are not reflexive. So the primal solution in infinite-dimensional systems may or may not exist. If the primal solution does not exist, the dual problem can provide a generalized solution of the problem.

Dual Equivalence Theorem. The following statements are equivalent to each others:

1. $(\mathscr{P}_{\inf})$ is stable and has a solution $\overline{\boldsymbol{u}}$.
2. $(\mathscr{P}^*_{\sup})$ is stable and has a solution $\overline{\boldsymbol{v}}^*$.
3. The extremality relation $P(\overline{\boldsymbol{u}}) = P^*(\overline{\boldsymbol{v}}^*)$ is satisfied.

On $\mathscr{U}_a$ and $\mathscr{V}^*_a$, the extremality condition $P(\overline{\boldsymbol{u}}) = P^*(\overline{\boldsymbol{v}}^*)$ and the Euler–Lagrange equations, Eqs. (32) and (33), are equivalent to each other. On the convex sets $\mathscr{U}_a$ and $\mathscr{V}^*_a$, the extremum problems $(\mathscr{P}_{\inf})$ and $(\mathscr{P}_{\sup})$ and the following variational inequalities are equivalent to each other in the sense

that they have the same solution set.

$$\text{(PVI)} \qquad (DP(\overline{\boldsymbol{u}}), \boldsymbol{u} - \overline{\boldsymbol{u}}) \geq 0 \qquad \forall \boldsymbol{u} \in \mathscr{U}_a \tag{35}$$

$$\text{(DVI)} \qquad \langle DP^*(\overline{\boldsymbol{v}}^*), \boldsymbol{v}^* - \overline{\boldsymbol{v}}^* \rangle \leq 0 \qquad \forall \boldsymbol{v}^* \in \mathscr{V}_a^* \tag{36}$$

Furthermore, if $\mathscr{C} = \{\boldsymbol{u} \in \mathscr{U} | \ \boldsymbol{u} \geq 0\}$ is a *convex cone,* $\mathscr{C}^* = \{\boldsymbol{u}^* \in \mathscr{U}^* | \ \boldsymbol{u}^* \geq 0\}$ is its *polar cone,* then these problems are equivalent to the following *nonlinear complementarity problem* (NCP):

$$\text{(NCP)} \qquad \boldsymbol{s} = DP(\boldsymbol{u}), \qquad \boldsymbol{u} \in \mathscr{C}, \boldsymbol{s} \in \mathscr{C}^*, \boldsymbol{u} \perp \boldsymbol{s} \tag{37}$$

where $\boldsymbol{s} \in \mathscr{C}^*$ is the so-called *vector of dual slacks.* The *complementarity condition* $\boldsymbol{u} \perp \boldsymbol{s}$ means that $\boldsymbol{u}$ and $\boldsymbol{s}$ are perpendicular to each other. Conditions in Eq. (39) are called the Karush–Kuhn–Tucker (KKT) *constraint qualification* in convex programming. To construct the dual complementarity problem, we need the inverse operator Λ^{-1} (see Ref. 12). In infinite-dimensional systems, to find Λ^{-1} is usually very difficult.

LAGRANGE DUALITY AND HAMILTONIAN

In order to study duality theory in nonconvex problems, we need the so-called Lagrangian form. Let $L: \mathscr{U} \times \mathscr{V}^* \to \bar{\mathbb{R}}$ be an arbitrarily given real-valued function. The following inequality is always true:

$$\sup_{\boldsymbol{v}^* \in \mathscr{V}^*} \inf_{\boldsymbol{u} \in \mathscr{U}} L(\boldsymbol{u}, \boldsymbol{v}^*) \leq \inf_{\boldsymbol{u} \in \mathscr{U}} \sup_{\boldsymbol{v}^* \in \mathscr{V}^*} L(\boldsymbol{u}, \boldsymbol{v}^*) \tag{40}$$

A point $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ is said to be a *minimax point* of L if

$$\sup_{\boldsymbol{v}^* \in \mathscr{V}^*} \inf_{\boldsymbol{u} \in \mathscr{U}} L(\boldsymbol{u}, \boldsymbol{v}^*) = L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) = \inf_{\boldsymbol{u} \in \mathscr{U}} \sup_{\boldsymbol{v}^* \in \mathscr{V}^*} L(\boldsymbol{u}, \boldsymbol{v}^*) \tag{41}$$

A point $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ is said to be a *saddle point* of L if

$$L(\overline{\boldsymbol{u}}, \boldsymbol{v}^*) \leq L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) \leq L(\boldsymbol{u}, \overline{\boldsymbol{v}}^*) \qquad \forall (\boldsymbol{u}, \boldsymbol{v}^*) \in \mathscr{U} \times \mathscr{V}^* \tag{42}$$

A point $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ is said to be a *subcritical* (*or* ∂^-*-critical*) *point* of L if

$$L(\overline{\boldsymbol{u}}, \boldsymbol{v}^*) \geq L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) \leq L(\boldsymbol{u}, \overline{\boldsymbol{v}}^*) \qquad \forall (u, \boldsymbol{v}^*) \in \mathscr{U} \times \mathscr{V}^* \tag{43}$$

A point $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ is said to be a *supercritical* (*or* ∂^+*-critical*) *point* of L if

$$L(\overline{\boldsymbol{u}}, \boldsymbol{v}^*) \leq L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) \geq L(\boldsymbol{u}, \overline{\boldsymbol{v}}^*) \qquad \forall (\boldsymbol{u}, \boldsymbol{v}^*) \in \mathscr{U} \times \mathscr{V}^* \tag{44}$$

Obviously, the function L possesses a saddle point $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ on $\mathscr{U} \times \mathscr{V}^*$ if and only if

$$\max_{\boldsymbol{v}^* \in \mathscr{V}^*} \inf_{\boldsymbol{u} \in \mathscr{U}} L(\boldsymbol{u}, \boldsymbol{v}^*) = \min_{\boldsymbol{u} \in \mathscr{U}} \sup_{\boldsymbol{v}^* \in \mathscr{V}^*} L(\boldsymbol{u}, \boldsymbol{v}^*) = L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) \tag{45}$$

In general, we have the following connection between the minimax theorem and the saddle point theorem:

Minimax Theorem. If there exists a minimax point $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) \in \mathscr{U} \times \mathscr{V}^*$ such that

$$L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) = \min_{\boldsymbol{u} \in \mathscr{U}} \max_{\boldsymbol{v}^* \in \mathscr{V}^*} L(\boldsymbol{u}, \boldsymbol{v}^*) = \max_{\boldsymbol{v}^* \in \mathscr{V}^*} \min_{\boldsymbol{u} \in \mathscr{U}} L(\boldsymbol{u}, \boldsymbol{v}^*) \tag{46}$$

then $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ is a saddle point. Conversely, if $L(\boldsymbol{u}, \boldsymbol{v}^*)$ possesses a saddle point $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) \in \mathscr{U} \times \mathscr{V}^*$, then the following minimax theorem holds:

$$L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) = \min_{\boldsymbol{u} \in \mathscr{C}} \max_{\boldsymbol{v}^* \in \mathscr{V}^*} L(\boldsymbol{u}, \boldsymbol{v}^*) = \max_{\boldsymbol{v}^* \in \mathscr{D}^*} \min_{\boldsymbol{u} \in \mathscr{U}} L(\boldsymbol{u}, \boldsymbol{v}^*) \tag{47}$$

This theorem shows that the existence of a saddle point implies the existence of a minimax point. However, the inverse result holds only on $\mathscr{C} \times \mathscr{D}^*$. This is because $\max_{\mathscr{V}^*} L(\boldsymbol{u}, \boldsymbol{v}^*)$ may not necessarily exist for all $\boldsymbol{u} \in \mathscr{U}$ and also $\min_{\mathscr{U}} L(\boldsymbol{u}, \boldsymbol{v}^*)$ may not necessarily exist for all $\boldsymbol{v}^* \in \mathscr{V}^*$.

The function $L(\boldsymbol{u}, \boldsymbol{v}^*)$ is said to be a *Lagrangian form* of problem $(\mathscr{P}^*_{\text{sup}})$ if

$$L(\boldsymbol{u}, \boldsymbol{v}^*) = \langle \Lambda \boldsymbol{u}, \boldsymbol{v}^* \rangle - V^*(\boldsymbol{v}^*) - F(\boldsymbol{u}) \tag{48}$$

A vector $\overline{\boldsymbol{u}} \in \mathscr{U}$ is said to be a *Lagrange multiplier* for $(\mathscr{P}^*_{\text{sup}})$ if $\overline{\boldsymbol{u}}$ is an optimal solution to $(\mathscr{P}_{\text{inf}})$.

Dually, the Lagrangian form of problem $(\mathscr{P}_{\text{inf}})$ is defined by

$$L^*(\boldsymbol{u}, \boldsymbol{v}^*) = -\langle \Lambda \boldsymbol{u}, \boldsymbol{v}^* \rangle + V(\Lambda \boldsymbol{u}) + F^*(\Lambda^* \boldsymbol{v}^*) \tag{49}$$

which is also called the *conjugate Lagrangian form.* A vector $\overline{\boldsymbol{v}}^* \in \mathscr{V}^*$ is said to be a *Lagrange multiplier* for $(\mathscr{P}_{\text{inf}})$ if $\overline{\boldsymbol{v}}^*$ is an optimal solution to $(\mathscr{P}^*_{\text{sup}})$. Obviously, we have $L + L^* = P + P^*$. If $\Lambda: \mathscr{U} \to \mathscr{V}$ and $\Lambda^*: \mathscr{V}^* \to \mathscr{U}^*$ are one-to-one and surjective, then the duals of the following results about L also hold for L^*.

A point $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) \in \mathscr{C} \times \mathscr{D}^*$ is said to be a critical point of L if L is G-differentiable at $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ with respect to both $\boldsymbol{u}$ and $\boldsymbol{v}^*$ separately and

$$D_{\boldsymbol{u}} L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) = 0, \qquad \Rightarrow \Lambda^* \overline{\boldsymbol{v}}^* = DF(\overline{\boldsymbol{u}}) \tag{50}$$

$$D_{\boldsymbol{v}^*} L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) = 0, \qquad \Rightarrow \Lambda \overline{\boldsymbol{u}} = DV^*(\overline{\boldsymbol{v}}^*) \tag{51}$$

It is easy to establish the following result:

Critical Points Theorem. If $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) \in \mathscr{C} \times \mathscr{D}^*$ is either a saddle point or a super (or sub) critical point of L, then $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ is a critical point of L, $DP(\overline{\boldsymbol{u}}) = 0$, $DP^*(\overline{\boldsymbol{v}}^*) = 0$ and

$$P(\overline{\boldsymbol{u}}) = L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) = P^*(\overline{\boldsymbol{v}}^*) \tag{52}$$

If $F: \mathscr{U} \to \bar{\mathbb{R}}$ is u.s.c., concave, and $V: \mathscr{V} \to \bar{\mathbb{R}}$ is l.s.c., convex, then L is a saddle function, and

$$P(\boldsymbol{u}) = \sup_{\boldsymbol{v}^* \in \mathscr{V}^*} L(\boldsymbol{u}, \boldsymbol{v}^*), \qquad P^*(\boldsymbol{v}^*) = \inf_{\boldsymbol{u} \in \mathscr{U}} L(\boldsymbol{u}, \boldsymbol{v}^*) \tag{53}$$

In this case, $P(\boldsymbol{u}) \geq L(\boldsymbol{u}, \boldsymbol{v}^*) \geq P^*(\boldsymbol{v}^*)$ $\forall (\boldsymbol{u}, \boldsymbol{v}^*) \in \mathscr{U} \times \mathscr{V}^*$, and we have

Saddle Point Theorem. $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ is a saddle point of L if and only if $\overline{\boldsymbol{u}}$ is a primal solution of $(\mathscr{P}_{\text{inf}})$, $\overline{\boldsymbol{v}}^*$ is a dual solution of $(\mathscr{P}^*_{\text{sup}})$, and $\inf P = \sup P^*$.

If both $F: \mathscr{U} \to \bar{\mathbb{R}}$ and $V: \mathscr{V} \to \bar{\mathbb{R}}$ are convex, l.s.c., then $L: \mathscr{U} \times \mathscr{V}^* \to \bar{\mathbb{R}}$ is a supercritical function and

$$P(\boldsymbol{u}) = \sup_{\mathscr{V}^*} L(\boldsymbol{u}, \boldsymbol{v}^*), \qquad P^*(\boldsymbol{v}^*) = \sup_{\boldsymbol{u} \in \mathscr{U}} L(\boldsymbol{u}, \boldsymbol{v}^*) \tag{54}$$

In this case, both P and P^* are nonconvex and $P(\boldsymbol{u}) \geq L(\boldsymbol{u}, \boldsymbol{v}^*) \leq P^*(\boldsymbol{v}^*)$ $\forall (\boldsymbol{u}, \boldsymbol{v}^*) \in \mathscr{U} \times \mathscr{V}^*$.

Dual Max–Min Theorem. If $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) \in \mathscr{C} \times \mathscr{D}^*$ is a supercritical point of L, then either

$$P(\overline{\boldsymbol{u}}) = \sup_{\boldsymbol{u}\in\mathscr{U}} P(\boldsymbol{u}) = \sup_{\boldsymbol{v}^*\in\mathscr{V}^*} P^*(\boldsymbol{v}^*) = P^*(\overline{\boldsymbol{v}}^*) \tag{55}$$

or

$$P(\overline{\boldsymbol{u}}) = \inf_{\boldsymbol{u}\in\mathscr{U}} P(\boldsymbol{u}) = \inf_{\boldsymbol{v}^*\in\mathscr{V}^*} P^*(\boldsymbol{v}^*) = P^*(\overline{\boldsymbol{v}}^*) \tag{56}$$

Proof. Since $P(\overline{\boldsymbol{u}}) = L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) = P^*(\overline{\boldsymbol{v}}^*)$, if $\overline{\boldsymbol{u}}$ maximizes P, then

$$\begin{aligned} P(\overline{\boldsymbol{u}}) &= \sup_{\boldsymbol{u}} P(\boldsymbol{u}) = \sup_{\boldsymbol{u}} \sup_{\boldsymbol{v}^*} L(\boldsymbol{u}, \boldsymbol{v}^*) = \sup_{\boldsymbol{v}^*} \sup_{\boldsymbol{u}} L(\boldsymbol{u}, \boldsymbol{v}^*) \\ &= \sup_{\boldsymbol{v}^*} P^*(\boldsymbol{v}^*) = P^*(\overline{\boldsymbol{v}}^*) \end{aligned} \tag{57}$$

as we can take the suprema in either order. If $\overline{\boldsymbol{u}}$ minimizes P, then

$$P(\overline{\boldsymbol{u}}) = \inf_{\boldsymbol{u}} P(\boldsymbol{u}) = \inf_{\boldsymbol{u}} \sup_{\boldsymbol{v}^*} L(\boldsymbol{u}, \boldsymbol{v}^*) = L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$$

Since $\overline{\boldsymbol{v}}^*$ is a critical point of P^*, it could be either a local extremum point or a saddle point of P^*. If $\overline{\boldsymbol{v}}^*$ is a saddle point of P^* and it maximizes P^* in the direction $\boldsymbol{v}_o^*$, then we have

$$P^*(\overline{\boldsymbol{v}}^*) = \sup_{\theta\geq 0} P^*(\overline{\boldsymbol{v}}^* + \theta\boldsymbol{v}_o^*) = \sup_{\boldsymbol{u}} \sup_{\theta\geq 0} L(\boldsymbol{u}, \overline{\boldsymbol{v}}^* + \theta\boldsymbol{v}_o^*) = \sup_{\boldsymbol{u}} P(\boldsymbol{u})$$

But $\overline{\boldsymbol{u}}$ is a minimizer of P. This contradiction shows that $\overline{\boldsymbol{v}}^*$ must be a minimizer of P^*.

In geometrically linear systems, the Lagrangian L is usually a saddle function for static problems. But in dynamic problems, L is usually a supercritical function. If V is a *kinetic energy* and F is a *potential energy,* then P is called the *total action* and P^* is called the *dual action.*

By using the Legendre transformation, the *Hamiltonian* $H: \mathscr{U} \times \mathscr{V}^* \to \overleftrightarrow{\mathbb{R}}$ can then be obtained from the Lagrangian as

$$H(\boldsymbol{u}, \boldsymbol{v}^*) = \langle \Lambda\boldsymbol{u}, \boldsymbol{v}^*\rangle - L(\boldsymbol{u}, \boldsymbol{v}^*) \tag{58}$$

If H is G-differentiable on $\mathscr{C} \times \mathscr{D}^*$, we have the following *Hamiltonian canonical equations:*

$$\Lambda\boldsymbol{u} = D_{\boldsymbol{v}^*} H(\boldsymbol{u}, \boldsymbol{v}^*), \qquad \Lambda^*\boldsymbol{v}^* = D_{\boldsymbol{u}} H(\boldsymbol{u}, \boldsymbol{v}^*) \tag{59}$$

If $\Lambda = d/dt$, its adjoint should be $\Lambda^* = -d/dt$. If $V(\Lambda\boldsymbol{u}) = \frac{1}{2}\langle \Lambda\boldsymbol{u}, C\Lambda\boldsymbol{u}\rangle$ is quadratic and the operator $K = \Lambda^* C\Lambda = K^*$ is self-adjoint, then the total action can be written as

$$I(\boldsymbol{u}) = \tfrac{1}{2}\langle \boldsymbol{u}, K\boldsymbol{u}\rangle - F(\boldsymbol{u}) \tag{60}$$

Let $I^c(\boldsymbol{u}) = -P^*(C\Lambda\boldsymbol{u})$, and thus the function $I^c: \mathscr{U} \to \overleftrightarrow{\mathbb{R}}$

$$I^c(\boldsymbol{u}) = \tfrac{1}{2}\langle \boldsymbol{u}, K\boldsymbol{u}\rangle - F^*(K\boldsymbol{u}) \tag{61}$$

is the so-called Clarke dual action (see Ref. 14). Let $K: \mathscr{C} \subset \mathscr{U} \to \mathscr{U}^*$ be a *closed, self-adjoint operator,* and let $\operatorname{Ker} K = \{\boldsymbol{u} \in \mathscr{U} \mid K\boldsymbol{u} = 0 \in \mathscr{U}^*\}$ be the *null space* of K, then we have

Clarke Duality Theorem. If $\overline{\boldsymbol{u}} \in \mathscr{C}$ is a critical point of I, then any vector $\boldsymbol{u} \in \operatorname{Ker} K + \overline{\boldsymbol{u}}$ is a critical point of I^c. Conversely, if there exists a $\boldsymbol{u}_o \in \mathscr{C}$ such that $K\boldsymbol{u}_o \in \mathscr{C}^*$, then for a given critical point $\overline{\boldsymbol{u}}$ of I^c, any vector $\boldsymbol{u} \in \operatorname{Ker} K + \overline{\boldsymbol{u}}$ is a critical point of I.

A comprehensive study of duality theory in linear dynamics is given in Ref. 15.

PRIMAL–DUAL SOLUTIONS AND CENTRAL PATH

Let us now demonstrate how the above scheme fits in with finite-dimensional linear programming. Let $\mathscr{U} = \mathscr{U}^* = \mathbb{R}^n$, $\mathscr{V} = \mathscr{V}^* = \mathbb{R}^m$, with the standard inner products $(\boldsymbol{u}, \boldsymbol{u}^*) = \boldsymbol{u}^T\boldsymbol{u}^*$ in $\mathbb{R}^n$, and $\langle \boldsymbol{v}, \boldsymbol{v}^*\rangle = \boldsymbol{v}^T\boldsymbol{v}^*$ in $\mathbb{R}^m$. For fixed $\overline{\boldsymbol{u}}^* = \boldsymbol{c} \in \mathbb{R}^n$ and $\overline{\boldsymbol{v}} = \boldsymbol{b} \in \mathbb{R}^m$, the primal problem is a constrained linear optimization problem:

$$(\mathscr{P}_{\text{lin}}) \qquad \min_{\boldsymbol{u}\in\mathbb{R}^n} (\boldsymbol{c}, \boldsymbol{u}) \quad s.t. \quad A\boldsymbol{u} = \boldsymbol{b}, \boldsymbol{u} \geq 0 \tag{62}$$

where $A \in \mathbb{R}^{m\times n}$ is a matrix. To reformulate this linear constrained optimization problem in the model form $(\mathscr{P}_{\text{inf}})$, we need to set $\Lambda = A$, $\mathscr{C} = \{\boldsymbol{u} \in \mathbb{R}^n \mid \boldsymbol{u} \geq 0\}$, and $\mathscr{D} = \{\boldsymbol{v} \in \mathbb{R}^m \mid \boldsymbol{v} = \boldsymbol{b}\}$, and let

$$F(\boldsymbol{u}) = -(\boldsymbol{c}, \boldsymbol{u}) - \Psi_{\mathscr{C}}(\boldsymbol{u}), \qquad V(\boldsymbol{v}) = \Psi_{\mathscr{D}}(\boldsymbol{v})$$

The conjugate functions in this elementary case may be calculated at once as

$$\begin{aligned} V^*(\boldsymbol{v}^*) &= \sup_{\boldsymbol{v}\in\mathscr{D}} \langle \boldsymbol{v}, \boldsymbol{v}^*\rangle = \langle \boldsymbol{b}, \boldsymbol{v}^*\rangle \qquad \forall \boldsymbol{v}^* \in \mathscr{D}^* = \mathbb{R}^m \\ F^*(\boldsymbol{u}^*) &= \inf_{\boldsymbol{u}\in\mathscr{C}} (\boldsymbol{u}, \boldsymbol{u}^* + \boldsymbol{c}) = -\Psi_{\mathscr{C}^*}(\boldsymbol{u}^* + \boldsymbol{c}) \end{aligned}$$

where $\mathscr{C}^*$ is a polar cone of $\mathscr{C}$. Let $\boldsymbol{p} = -\boldsymbol{v}^* \in \mathbb{R}^m$, the dual problem $(\mathscr{P}^*_{\text{sup}})$ can be written as

$$(\mathscr{P}^*_{\text{lin}}) \qquad \max_{\boldsymbol{p}\in\mathbb{R}^m} \{P^*(\boldsymbol{p}) = \langle \boldsymbol{b}, \boldsymbol{p}\rangle - \Psi_{\mathscr{C}^*}(\boldsymbol{c} - A^*\boldsymbol{p})\} \tag{63}$$

The implicit constraint in this problem is

$$\boldsymbol{p} \in \mathscr{V}_a^* = \{\boldsymbol{p} \in \mathbb{R}^m \mid \boldsymbol{c} - A^*\boldsymbol{p} \geq 0\}$$

For a given $\alpha \in \mathbb{R}^+ := \{\alpha \in \mathbb{R} \mid \alpha \geq 0\}$, let

$$\Psi_\alpha(\boldsymbol{p}) = \tfrac{1}{2}\alpha \|(A^*\boldsymbol{p} - \boldsymbol{c})_+\|^2$$

where $(x)_+ = \max\{0, x\}$. We have

$$\lim_{\alpha\to\infty} \Psi_\alpha(\boldsymbol{p}) = \Psi_{\mathscr{C}^*}(\boldsymbol{c} - A^*\boldsymbol{p})$$

So the inequality constraint in $(\mathscr{P}^*_{\text{lin}})$ can be relaxed by the following so-called *external penalty method:*

$$(\mathscr{P}^*_p) \qquad \lim_{\alpha\to\infty} \max_{\boldsymbol{p}\in\mathbb{R}^m} \{P^*_p(\boldsymbol{p}; \alpha) = \langle \boldsymbol{b}, \boldsymbol{p}\rangle - \Psi_\alpha(\boldsymbol{p})\} \tag{64}$$

For any given sequence $\{\alpha_k\} \to +\infty$, $P^*_p: \mathbb{R}^m \to \mathbb{R}$ is always concave, and the solution of $(\mathscr{P}^*_p)$ should be also a solution of $(\mathscr{P}^*_{\text{lin}})$. The main disadvantage of the penalty method is that the problem $(\mathscr{P}^*_p)$ will become unstable when the penalty parameter α_k increases.

The Lagrangian $L(\boldsymbol{u}, \boldsymbol{v}^*)$ of $(\mathscr{P}^*_{\text{lin}})$ is

$$L(\boldsymbol{u}, \boldsymbol{p}) = \langle A\boldsymbol{u}, -\boldsymbol{p}\rangle - \langle \boldsymbol{b}, -\boldsymbol{p}\rangle + (\boldsymbol{c}, \boldsymbol{u}) = (\boldsymbol{b}, \boldsymbol{p}) - (\boldsymbol{u}, A^*\boldsymbol{p} - \boldsymbol{c}) \tag{65}$$

But for the inequality constraint in $\mathscr{V}^*_a$, the Lagrange multiplier $\boldsymbol{u} \in \mathbb{R}^n$ has to satisfy the following KKT optimality conditions:

$$\begin{aligned} A\boldsymbol{u} &= \boldsymbol{b}, \qquad \boldsymbol{s} = \boldsymbol{c} - A^*\boldsymbol{p} \\ \boldsymbol{u} &\geq 0, \qquad \boldsymbol{s} \geq 0, \qquad \boldsymbol{s}^T\boldsymbol{u} = 0 \end{aligned} \tag{66}$$

The problem to find $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{p}}, \overline{\boldsymbol{s}})$ satisfying Eq. (66) is also known as the *mixed linear complementarity problem* (see Ref. 16)

Combining both the penalty method and the Lagrange method, we have

$$L_{pd}(\boldsymbol{u}, \boldsymbol{p}; \alpha) = L(\boldsymbol{u}, \boldsymbol{p}) - \Psi_\alpha(\boldsymbol{p}) \tag{67}$$

The so-called *augmented Lagrangian method* for solving constrained problem $(\mathscr{P}^*_{\text{lin}})$ is then

$$(\mathscr{P}^*_{pd}) \qquad \min_{(\alpha,\boldsymbol{u})\in\mathbb{R}^+\times\mathbb{R}^n} \max_{\boldsymbol{p}\in\mathbb{R}^m} L_{pd}(\boldsymbol{u}, \boldsymbol{p}; \alpha) \tag{68}$$

Penalty–Duality Theorem. There exists a finite $\alpha_o > 0$ such that for any given $\alpha \in [\alpha_o, +\infty)$, the solution of the following saddle point problem:

$$\min_{\boldsymbol{u}\in\mathbb{R}^n} \max_{\boldsymbol{p}\in\mathbb{R}^m} L_{pd}(\boldsymbol{u}, \boldsymbol{p}; \alpha) \tag{69}$$

is also a solution of $(\mathscr{P}^*_{\text{lin}})$. Moreover, for a given penalty-duality sequence $(\alpha_k, \boldsymbol{u}_k) \in \mathbb{R}^+ \times \mathbb{R}^n$, the optimal solution $\overline{\boldsymbol{p}}_k$ of the following unconstrained problem

$$\max_{\boldsymbol{p}\in\mathbb{R}^m} L_{pd}(\boldsymbol{u}_k, \boldsymbol{p}; \alpha_k) \tag{70}$$

is an optimal solution of $(\mathscr{P}^*_{\text{lin}})$ if and only if $\boldsymbol{p}_k \in \mathscr{V}^*_a$.

This theorem shows that by constructing a penalty-duality sequence $(\alpha_k, \boldsymbol{u}_k) \in [\alpha_o, +\infty) \times \mathbb{R}^n$, the constrained problem $(\mathscr{P}^*_{\text{lin}})$ can be relaxed by an unconstrained problem [Eq. (70)]. This method is much better than the pure penalty method. Detailed study of the augmented Lagrange methods and applications are given in Ref. 17.

By using the vector of dual slacks $\boldsymbol{s} \in \mathbb{R}^n$, the dual problem $(\mathscr{P}^*_{\text{lin}})$ can be rewritten as

$$(\mathscr{P}^*_{\text{lin}}) \qquad \max_{\boldsymbol{p}\in\mathbb{R}^m} \langle \boldsymbol{b}, \boldsymbol{p}\rangle \quad s.t. \quad A^*\boldsymbol{p} + \boldsymbol{s} = \boldsymbol{c}, \qquad \boldsymbol{s} \geq 0 \tag{71}$$

We can see that the primal variable $\boldsymbol{u}$ is the Lagrange multiplier for the constraint $A^*\boldsymbol{p} - \boldsymbol{c} \leq 0$ in the dual problem. However, the dual variables $\boldsymbol{p}$ and $\boldsymbol{s}$ are, respectively, Lagrange multipliers for the constraints $A\boldsymbol{u} = \boldsymbol{b}$ and $\boldsymbol{u} \geq 0$ in the primal problem. These choices are not accidents.

Strong Duality Theorem 2. The vector $\overline{\boldsymbol{u}} \in \mathbb{R}^n$ is a solution of $(\mathscr{P}_{\text{lin}})$ if and only if there exists Lagrange multiplier $(\overline{\boldsymbol{p}}, \overline{\boldsymbol{s}}) \in \mathbb{R}^m \times \mathbb{R}^n$ for which the KKT optimality conditions [Eq. (66)] hold for $(\boldsymbol{u}, \boldsymbol{p}, \boldsymbol{s}) = (\overline{\boldsymbol{u}}, \overline{\boldsymbol{p}}, \overline{\boldsymbol{s}})$. Dually, the vector $(\overline{\boldsymbol{p}}, \overline{\boldsymbol{s}}) \in \mathbb{R}^m \times \mathbb{R}^n$ is a solution of $(\mathscr{P}^*_{\text{lin}})$ if and only if there exists a Lagrange multiplier $\overline{\boldsymbol{u}} \in \mathbb{R}^n$ such that the KKT conditions [Eq. (66)] hold for $(\boldsymbol{u}, \boldsymbol{p}, \boldsymbol{s}) = (\overline{\boldsymbol{u}}, \overline{\boldsymbol{p}}, \overline{\boldsymbol{s}})$.

The vector $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{p}}, \overline{\boldsymbol{s}})$ is called a *primal–dual solution* of $(\mathscr{P}_{\text{lin}})$. The so-called *primal–dual methods* in mathematical programming are those methods to find primal–dual solutions $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{p}}, \overline{\boldsymbol{s}})$ by applying variants of Newton's method to the three equations in Eq. (66) and modifying the search directions and step lengths so that the inequalities in Eq. (66) are satisfied at every iteration. If the inequalities are strictly satisfied, the methods are called *primal–dual interior-point methods.* In these methods, the so-called *central path* $\mathscr{C}_{\text{path}}$ plays a vital role in the theory of primal–dual algorithms. It is a parametrical curve of strictly feasible points defined by

$$\mathscr{C}_{\text{path}} = \{(\boldsymbol{u}_\tau, \boldsymbol{p}_\tau, \boldsymbol{s}_\tau)^T \in \mathbb{R}^{2n+m} | \tau > 0\} \tag{72}$$

where each point $(\boldsymbol{u}_\tau, \boldsymbol{p}_\tau, \boldsymbol{s}_\tau)$ solves the following system:

$$\begin{aligned} A\boldsymbol{u} &= \boldsymbol{b}, \qquad A^*\boldsymbol{p} + \boldsymbol{s} = \boldsymbol{c} \\ \boldsymbol{u} > 0, \qquad \boldsymbol{s} &> 0, \qquad u_i s_i = \tau, \quad i = 1, 2, \ldots, n \end{aligned} \tag{73}$$

This problem has a unique solution $(\boldsymbol{u}_\tau, \boldsymbol{p}_\tau, \boldsymbol{s}_\tau)$ for each $\tau > 0$ if and only if the *strictly feasible set*

$$\mathscr{T}_o = \{(\boldsymbol{u}, \boldsymbol{p}, \boldsymbol{s}) | \, A\boldsymbol{u} = \boldsymbol{b}, A^T\boldsymbol{p} + \boldsymbol{s} = \boldsymbol{c}, \boldsymbol{u} > 0, \boldsymbol{s} > 0\} \tag{74}$$

is nonempty. A comprehensive study of the primal–dual interior-point methods in mathematical programming has been given in Ref. 4

DUALITY IN FULLY NONLINEAR OPTIMIZATION

In fully nonlinear systems, $\Lambda(\boldsymbol{u})$ is a nonlinear operator. The nonlinear Lagrangian form is (see Ref. 11)

$$L(\boldsymbol{u}, \boldsymbol{v}^*) = \langle \Lambda(\boldsymbol{u}), \boldsymbol{v}^*\rangle - V^*(\boldsymbol{v}^*) - F(\boldsymbol{u}) \tag{75}$$

The critical condition $\delta L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*; \boldsymbol{u}, \boldsymbol{v}^*) = 0 \; \forall (\boldsymbol{u}, \boldsymbol{v}^*) \in \mathscr{C} \times \mathscr{D}^*$ gives the canonic equations

$$D_{\boldsymbol{u}} L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) = 0 \Rightarrow \Lambda_t^*(\overline{\boldsymbol{u}})\overline{\boldsymbol{v}}^* = DF(\overline{\boldsymbol{u}}) \tag{76}$$

$$D_{\boldsymbol{v}} L(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) = 0 \Rightarrow \Lambda(\overline{\boldsymbol{u}}) = DV^*(\overline{\boldsymbol{v}}^*) \tag{77}$$

Since V is either convex or concave on $\mathscr{D}$, the inverse constitutive equation is equivalent to $\overline{\boldsymbol{v}}^* = DV(\Lambda(\overline{\boldsymbol{u}}))$. Then the fundamental equation in fully nonlinear systems should be

$$\Lambda_t^*(\overline{\boldsymbol{u}}) DV(\Lambda(\overline{\boldsymbol{u}})) = DF(\overline{\boldsymbol{u}}) \tag{78}$$

We can see that the symmetry is broken in geometrically nonlinear systems.

If Λ is a quadratic operator, the Taylor expansion of Λ at $\overline{\boldsymbol{u}}$ should be $\Lambda(\overline{\boldsymbol{u}} + \delta\boldsymbol{u}) = \Lambda(\overline{\boldsymbol{u}}) + \Lambda_t(\overline{\boldsymbol{u}})\delta\boldsymbol{u} + \frac{1}{2}\delta^2\Lambda(\overline{\boldsymbol{u}}; \delta\boldsymbol{u})$. We now assume that

(*A*2) $F : \mathscr{C} \to \mathbb{R}$ is linear and Λ is a quadratic operator such that $\delta^2\Lambda(\overline{\boldsymbol{u}}; \delta\boldsymbol{u}) = -2\Lambda_n(\delta\boldsymbol{u})$

Under this assumption, if $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ is a critical point of L, we have $L(\boldsymbol{u}, \boldsymbol{v}^*) - L(\overline{\boldsymbol{u}}, \boldsymbol{v}^*) = G(\boldsymbol{u} - \overline{\boldsymbol{u}}, \boldsymbol{v}^*)$. If $V : \mathscr{D} \to \vec{\mathbb{R}}$ is

convex, then (see Ref. 12)

$(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$is a saddle point of L if and only if $G(\boldsymbol{u}, \overline{\boldsymbol{v}}^*) \geq 0$

$$\forall \boldsymbol{u} \in \mathscr{C} \quad (79)$$

$(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$is a supercritical point of L if and only if

$$G(\boldsymbol{u}, \overline{\boldsymbol{v}}^*) < 0 \qquad \forall \boldsymbol{u} \in \mathscr{C} \quad (80)$$

In this case, $P(\boldsymbol{u}) = \sup_{v^* \in \mathscr{V}^*} L(\boldsymbol{u}, \boldsymbol{v}^*) = V(\Lambda(\boldsymbol{u})) - F(\boldsymbol{u})$. But its conjugate function will depend on the sign of G (see Ref. 12):

$$P^*(\boldsymbol{v}^*) = \begin{cases} \inf_{\boldsymbol{u} \in \mathscr{U}} L(\boldsymbol{u}, \boldsymbol{v}^*) & \text{if } G(\boldsymbol{u}, \overline{\boldsymbol{v}}^*) \geq 0 \quad \forall \boldsymbol{u} \in \mathscr{U} \\ \sup_{\boldsymbol{u} \in \mathscr{U}} L(\boldsymbol{u}, \boldsymbol{v}^*) & \text{if } G(\boldsymbol{u}, \overline{\boldsymbol{v}}^*) < 0 \quad \forall \boldsymbol{u} \in \mathscr{U} \end{cases} \quad (81)$$

We have the following interesting result:

Triality Theorem. Suppose that the assumption (A2) holds and $V: \mathscr{V} \to \vec{\mathbb{R}}$ is convex, proper and l.s.c. Let $\mathscr{C}_b \times \mathscr{D}_b^*$ be a neighborhood of a critical point $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ of L such that on $\mathscr{C}_b \times \mathscr{D}_b^*$, $(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$ is the only critical point. Then if $G(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) \geq 0$, we obtain

$$P(\overline{\boldsymbol{u}}) = \inf_{\boldsymbol{u} \in \mathscr{C}_b} \sup_{\boldsymbol{v}^* \in \mathscr{D}_b^*} L(\boldsymbol{u}, \boldsymbol{v}^*) = \sup_{\boldsymbol{v}^* \in \mathscr{D}_b^*} \inf_{\boldsymbol{u} \in \mathscr{C}_b} L(\boldsymbol{u}, \boldsymbol{v}^*) = P^*(\overline{\boldsymbol{v}}^*) \quad (82)$$

If $G(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*) < 0$, we have either

$$P(\overline{\boldsymbol{u}}) = \inf_{\boldsymbol{u} \in \mathscr{C}_b} \sup_{\boldsymbol{v}^* \in \mathscr{D}_b^*} L(\boldsymbol{u}, \boldsymbol{v}^*) = \inf_{\boldsymbol{v}^* \in \mathscr{D}_b^*} \sup_{\boldsymbol{u} \in \mathscr{C}_b} L(\boldsymbol{u}, \boldsymbol{v}^*) = P^*(\overline{\boldsymbol{v}}^*) \quad (83)$$

or

$$P(\overline{\boldsymbol{u}}) = \sup_{\boldsymbol{u} \in \mathscr{C}_b} \sup_{\boldsymbol{v}^* \in \mathscr{D}_b^*} L(\boldsymbol{u}, \boldsymbol{v}^*) = \sup_{\boldsymbol{v}^* \in \mathscr{D}_b^*} \sup_{\boldsymbol{u} \in \mathscr{C}_b} L(\boldsymbol{u}, \boldsymbol{v}^*) = P^*(\overline{\boldsymbol{v}}^*) \quad (84)$$

The proof of this theorem was given in Ref. 18. This theorem can be used to solve some nonconvex variational problems (see Refs. 3, 18, and 19).

If $V: \mathscr{D} \to \vec{\mathbb{R}}$ is concave, then

$(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$is a saddle point of $-L$ if and only if $G(\boldsymbol{u}, \overline{\boldsymbol{v}}^*) \leq 0$

$$\forall \boldsymbol{u} \in \mathscr{C} \quad (85)$$

$(\overline{\boldsymbol{u}}, \overline{\boldsymbol{v}}^*)$is a subcritical point of L if and only if $G(\boldsymbol{u}, \overline{\boldsymbol{v}}^*) > 0$

$$\forall \boldsymbol{u} \in \mathscr{C} \quad (86)$$

In this case, $P(\boldsymbol{u}) = \sup_{v^*} L(\boldsymbol{u}, \boldsymbol{v}^*)$. The dual problem depends also on the sign of the gap function and we have a similar triality theorem (see Ref. 18).

Example 2. Let us consider the minimization of the following nonconvex variational problem:

$$(\mathscr{P}_u) \qquad P(u) = \int_0^1 w(\Lambda(u))\, dt - \int_0^1 fu\, dt \to \min \qquad \forall u \in \mathscr{U}_a \quad (87)$$

where the source variable f is a given function and we let $f(1) = 0$; $w(v)$ could be either a convex or concave function of $v = \Lambda(u)$. As an example, we simply let $w(v) = \frac{1}{2}C(v - \lambda)^2$, with a given parameter $\lambda > 0$ and a material constant $C > 0$ and let Λ be a quadratic operator $\Lambda u = \frac{1}{2}[(d/dt)u]^2 = \frac{1}{2}(u_{,t})^2$. Then $w(\Lambda(u))$ is a double-well function of $\epsilon = u_{,t}$, and

$$\Lambda_t(\overline{u})u = \overline{u}_{,t}u_{,t}, \qquad \Lambda_n(\overline{u})u = -\tfrac{1}{2}\overline{u}_{,t}u_{,t} \quad (88)$$

For a mixed boundary value problem, the convex set $\mathscr{C}$ is a hyperplane

$$\mathscr{C} = \{u \in \mathscr{H}[0, 1] |\ u(0) = 0\}$$

and $\mathscr{D} = \{v \in \mathscr{H}[0, 1] |\ v(t) \geq 0\ \forall t \in [0, 1]\}$ is a convex cone. Then on the feasible set $\mathscr{U}_a$, $P(u)$ is nonconvex, Gâteaux differentiable.

The *direct methods* for solving nonconvex variational problems are difficult. However, by the triality theorem, a closed-form solution of this problem can be easily obtained (see Ref. 12). To do so, we need first to find the conjugate functions. We let $F(u) = \int_0^1 uf\, dt\ \forall u \in \mathscr{C}$. On $\mathscr{D}$, $V(v) = \frac{1}{2}\int_0^1 C(v - \lambda)^2\, dt$ is quadratic. Then the constitutive equation $v^* = \sigma = DV(v) = C(v - \lambda)$ is linear.

$$F^*(u^*) = \inf_{u \in \mathscr{C}} \left\{ \int_0^1 uu^*\, dt + u(1)u^*(1) - \int_0^1 uf\, dt \right\} = -\Psi_{\mathscr{C}^*}(u^*) \quad (89)$$

$$\begin{aligned} V^*(\sigma) &= \sup_{\boldsymbol{v} \in \mathscr{D}} \left\{ \int_0^1 \sigma v\, dt - V(v) \right\} \\ &= \int_0^1 \left(\frac{1}{2C}\sigma^2 + \lambda\sigma \right) dt + \Psi_{\mathscr{D}^*}(\sigma) \end{aligned} \quad (90)$$

where $\mathscr{C}^* = \{u^* \in \mathscr{H}[0, 1] |\ u^*(t) = f(t)\ \forall t \in (0, 1), u^*(1) = 0\}$ is a hyperplane, and $\mathscr{D}^* = \{\sigma \in \mathscr{H}[0, 1] |\ \sigma \neq 0\}$ ($\sigma = 0$ implies that $v = \lambda$.) Since $v = \frac{1}{2}u_{,t}^2 \geq 0\ \forall u \in \mathscr{C}$ the range of $\sigma = DV(v)$ should be $\mathscr{D}_r^* = \{\sigma \in \mathscr{H}[0, 1] - \lambda C \leq \sigma < +\infty\}$.

The Lagrangian $L: \mathscr{C} \times \mathscr{D}^* \to \mathbb{R}$ for this problem should be

$$L(u, \sigma) = \int_0^1 \left[\frac{1}{2}(u_{,t})^2\sigma - \left(\frac{1}{2C}\sigma^2 + \lambda\sigma \right) - fu \right] dt \quad (91)$$

The optimality conditions for this problem are

$$\frac{1}{2}u_{,t}^2 = \frac{1}{E}\sigma + \lambda \qquad \forall t \in (0, 1), u(0) = 0 \quad (92)$$

$$-[u_t\sigma]_{,t} = f(t) \qquad \forall t \in (0, 1), \sigma(1) = 0 \quad (93)$$

Let $\tau(t) = u_{,t}\sigma$. It is easy to find that

$$\tau(t) = \int_0^t -f(s)\, ds + \int_0^1 f(s)\, ds \quad (94)$$

The gap function in this problem is a quadratic function of u:

$$G(u, \sigma) = \langle \sigma, -\Lambda_n(u)u \rangle = \frac{1}{2}\int_0^1 \sigma u_{,t}^2\, dt$$

If $\lambda < 0$, then the gap function is positive on $\mathscr{D}_r^*$. In this case, $P(u)$ is convex, and the problem has a unique solution. If $\lambda >$

0, the gap function could be negative on $\mathscr{D}_r^*$. In this case, $P(u)$ is nonconvex and the primal problem may have more than one solution.

On $\mathscr{D}^*$, the conjugate function P^* obtained by Eq. (81) is well-defined:

$$P^*(\sigma) = -\int_0^1 \left[\frac{1}{2C}\sigma^2 + \lambda\sigma + \frac{1}{2}\tau^2/\sigma\right] dt \tag{95}$$

The dual Euler–Lagrange equation in this example is a cubic algebraic equation:

$$2\sigma^2\left(\frac{1}{C}\sigma + \lambda\right) = \tau^2 \qquad \forall t \in (0,1) \tag{96}$$

For a given $f(t)$ such that $\tau(t)$ is obtained by Eq. (94), this equation has at most three solutions σ_i $(i = 1, 2, 3)$. Since $\tau = u_t\sigma$, $u(0) = 0$, the analytic solution for this nonconvex variational problem is

$$u_i(t) = \int_0^t \frac{\tau(s)}{\sigma_i(s)}\, ds, \qquad i = 1, 2, 3 \tag{97}$$

By the Triality Theorem we know that $P(u_i) = P^*(\sigma_i)$. The properties of u_i are given by the triality theorem. For certain given f and λ such that $\sigma_1 > 0 > \sigma_2 > \sigma_3$, u_1 is a global minimizer of P, u_2 is a local minimizer, and u_3 is a local maximizer of P. To see this, let $C = 1$; the conjugate function of

$$W^*(\sigma) = -\tfrac{1}{2}(\sigma^2 + 2\lambda\sigma + \tau/\sigma) \tag{98}$$

is the well-known van der Waals *double-well function:*

$$W(u) = \tfrac{1}{2}(\tfrac{1}{2}u^2 - \lambda)^2 - \tau u \tag{99}$$

Figure 2 shows the graphs of W (solid line) and W^* (dashed line). The Lagrangian associated with the problem min $W(u)$ is simply given as

$$L(u,\sigma) = \tfrac{1}{2}u^2\sigma - (\tfrac{1}{2}\sigma^2 + \lambda\sigma) - \tau u \tag{100}$$

Figure 3 shows that L is a saddle function when $\sigma \geq 0$. L is concave if $\sigma < 0$.

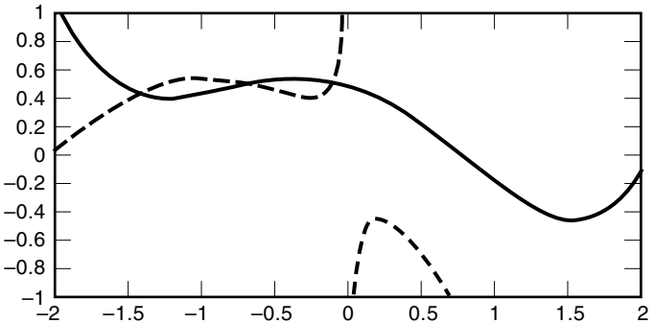


Figure 2. Graphs of $F(u)$ (solid) and $F^*(\sigma)$ (dashed).

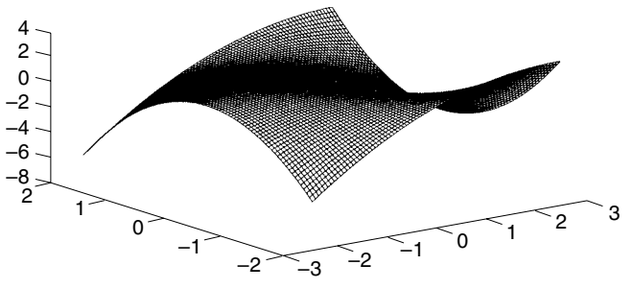


Figure 3. Lagrangian $L(u, \sigma)$.

CONCLUSIONS

Duality theory plays a crucial role in many natural phenomena. It can be used to study wider classes of problems in engineering and science. For geometrically linear systems, duality theory and methods are quite well understood. The excellent textbooks by Strang (1) and Wright (4) are highly recommended. An informal general result was proposed in Ref. 20.

General Duality Principle. For a given system $\mathscr{S}$, if there exists a geometrically linear operator $\Lambda : \mathscr{U} \to \mathscr{V}$ such that the primal system $\mathscr{S}_p = \{\mathscr{U}, \mathscr{V}; \Lambda\}$ and the dual system $\mathscr{S}_d = \{\mathscr{U}^*, \mathscr{V}^*; \Lambda^*\}$ are isomorphic, then

1. For each statement in the primal system $\mathscr{S}_p$, there exists a complementary statement, which is obtained by applying this statement to the dual system $\mathscr{S}_d$; and
2. For each valid theorem defined on the whole system $\mathscr{S} = \mathscr{S}_p \cup \mathscr{S}_d$, the dual theorem, which is obtained by changing all the concepts in the original theorem to their duals, is also valid on $\mathscr{S}$.

From the point of view of the *category theory* (see Ref. 21), the primal system $\mathscr{S}_p$ and the dual system $\mathscr{S}_d$ are said to be *isomorphic* if there exists a so-called *contravariant factor* $\mathscr{T}$ such that the map $\mathscr{T} : \mathscr{S}_p \to \mathscr{S}_d$ is *one-to-one and surjective.* The dual concepts include the paired variables ($\mathbf{u}$, $\mathbf{u}^*$), ($\mathbf{v}$, $\mathbf{v}^*$), conjugate functionals, as well as the dual operations (Λ, Λ^*), ($\geq$, $\leq$), (inf, sup), and so on.

In fully nonlinear systems, the one-to-one symmetrical relations between the primal and dual systems do not usually exist. The duality theory depends on the choice of the nonlinear operator Λ and the associated gap function. The triality theory reveals an intrinsic symmetry in fully nonlinear systems. For a given nonlinear system, the choice of Λ may not be unique, but a quadratic operator will make problems much easier. As long as the paired intermediate variables are defined correctly, the duality theory presented in this article can be used to develop both new theoretical results and powerful numerical methods. A comprehensive study and applications of the duality principle in nonconvex systems are given in Ref. 3. Primal–dual algorithms have been developed for both linear programming (see Ref. 4) and nonconvex problems (see Ref. 22). Triality theory can be used to develop algorithms for robust numerical solutions in fully nonlinear, nonconvex problems.

BIBLIOGRAPHY

1. G. Strang, *Introduction to Applied Mathematics,* Cambridge: Wellesley–Cambridge Univ. Press, 1986.
2. E. Tonti, A mathematical model for physical theories, *Rend. Accad. Lincei,* **LII**, I, 133–139; II, 350–356, 1972.
3. D. Y. Gao, *Duality Principles in Nonlinear Systems: Theory, Methods and Applications,* Dordrecht: Kluwer, 1999.
4. S. J. Wright, *Primal–Dual Interior-Point Methods,* Philadelphia: SIAM, 1996.
5. I. Ekeland and R. Temam, *Convex Analysis and Variational Problems,* Amsterdam: North-Holland, 1976.
6. R. T. Rockafellar, *Conjugate Duality and Optimization,* Philadelphia: SIAM, 1974.
7. M. J. Sewell, *Maximum and Minimum Principles,* Cambridge: Cambridge Univ. Press, 1987.
8. M. Walk, *Theory of Duality in Mathematical Programming,* Berlin: Springer-Verlag, 1989.
9. G. Auchmuty, Duality for non-convex variational principles, *J. Differ. Equ.,* **50**: 80–145, 1983.
10. J. F. Toland, A duality principle for non-convex optimization and the calculus of variations, *Arch. Rational Mech. Anal.,* **71**: 41–61, 1979.
11. D. Y. Gao and Strang G., Geometric nonlinearity: Potential energy, complementary energy, and the gap function, *Q. Appl. Math.,* **XLVII** (3): 487–504, 1989.
12. D. Y. Gao, Dual extremum principles in finite deformation theory with applications in post-buckling analysis of nonlinear beam model, *Appl. Mech. Rev., ASME,* **50** (11): S64–S71, 1997.
13. D. Y. Gao, Minimax and Triality Theory in Nonsmooth Variational Problems, in M. Fukushima and L. Q. Qi (eds.), *Reformulation—Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods,* Dordrecht: Kluwer, 1998, pp. 161–180.
14. I. Ekeland, *Convexity Methods in Hamiltonian Mechanics,* Berlin: Springer-Verlag, 1990.
15. B. Tabarrok and F. P. J. Rimrott, *Variational Methods and Complementary Formulations in Dynamics,* Dordrecht: Kluwer, 1994, p. 366.
16. R. W. Cottle, J. S. Pang, and R. E. Stone, *The Linear Complementarity Problems,* New York: Academic Press, 1992.
17. M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems,* Amsterdam: North-Holland, 1983.
18. D. Y. Gao, Duality, triality and complementary extremum principles in nonconvex parametric variational problems with applications, *IMA J. Appl. Math.,* 1998, in press.
19. D. Y. Gao, Duality in Nonconvex Finite Deformation Theory: A Survey and Unified Approach, in R. Gilbert, P. D. Panagiotopoulos, and P. Pardalos (eds.), *From Convexity to Nonconvexity, A Volume Dedicated to the Memory of Professor Gaetano Fichera,* Dordrecht: Kluwer, 1998, in press.
20. D. Y. Gao, *Complementary-duality theory in elastoplastic systems and pan-penalty finite element methods,* Ph.D. Thesis, Tsinghua University, Beijing, 1986.
21. R. Geroch, *Mathematical Physics,* Chicago: University of Chicago Press, 1985.
22. G. Auchmuty, Duality algorithms for nonconvex variational principles, *Numer. Funct. Anal. Optim.,* **10**: 211–264, 1989.

DAVID YANG GAO
Virginia Polytechnic Institute and State University

DUALITY OF MAGNETIC AND ELECTRIC CIRCUITS. See MAGNETIC CIRCUITS.

DUCTILE ALLOY SUPERCONDUCTORS. See SUPERCONDUCTORS, METALLURGY OF DUCTILE ALLOYS.

DUCTING. See REFRACTION AND ATTENUATION IN THE TROPOSPHERE.

DURATION MEASUREMENT. See TIME MEASUREMENT.

DVD-ROMS. See CD-ROMS, DVD-ROMS, AND COMPUTER SYSTEMS.

DYADIC GREEN'S FUNCTION. See GREEN'S FUNCTION METHODS.

Eigenvalues and Eigenfunctions

Standard Article

Yuri V. Makarov[1] and Zhao Yang Dong[2]

[1]Howard University, Washington, DC

[2]University of Sydney, Sydney, NSW, Australia

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2413

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (253K)

Abstract

The sections in this article are

Definition of Eigenvalue and Eigenfunction

Some Properties of Eigenvalues and Eigenvectors

Eigenvalue Analysis for Ordinary Differential Equations

Eigenvalues and Eigenfunctions for Integral Equations

Linear Dynamic Models and Eigenvalues

Eigenvalues and Stability

Eigenvalues and Bifurcations

Numerical Methods for the Eigenvalue Problem

Some Practical Applications of Eigenvalues and Eigenvectors

Acknowledgments

Keywords: eigenvalues; eigenvectors; eigenfunctions; singular decomposition and singular values; state matrix; jordan form; characteristic equation and polynomial; eigenvalue localization; modal analysis; participation factors; eigenvalue sensitivity; observability; QR transformation; bifurcations; small-signal stability; damping of oscillations

| | |
Copyright © 1999-2008 All Rights Reserved.

EIGENVALUES AND EIGENFUNCTIONS

DEFINITION OF EIGENVALUE AND EIGENFUNCTION

Many physical system models deal with a square matrix $A = [a_{i,j}]_{n\times n}$ and its eigenvalues and eigenvectors. The eigenvalue problem aims to find a nonzero vector $x = [x_1]_{1\times n}$ and scalar λ such that they satisfy the following equation:

$$Ax = \lambda x \tag{1}$$

where λ is the *eigenvalue* (or characteristic value or proper value) of matrix A, and x is the corresponding right *eigenvector* (or characteristic vector or proper vector) of A.

The necessary and sufficient condition for Eq. (1) to have a nontrivial solution for vector x is that the matrix $(\lambda I - A)$ is singular. Equivalently, the last requirement can be rewritten as a *characteristic equation* of A:

$$\det(\lambda I - A) = 0 \tag{2}$$

where I is the identity matrix. All n roots of the characteristic equation are all n eigenvalues $[\lambda_1, \lambda_2, \ldots, \lambda_n]$. Expansion of $\det(\lambda I - A)$ as a scalar function of λ gives the *characteristic polynomial* of A:

$$L(\lambda) = a_n\lambda^n + a_{n=1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 \tag{3}$$

where λ^k, $k = 1, \ldots, n$, are the corresponding kth powers of λ, and a_k, $k = 0, \ldots, n$, are the coefficients determined via the elements a_{ij} of A.

Each eigenvalue also corresponds to a left eigenvector l, which is the right eigenvector of matrix A^T where the superscript T denotes the transpose of A. The left eigenvector satisfies the equation

$$(\lambda I - A^T)l = 0 \tag{4}$$

The set of all eigenvalues is called the *spectrum* of A.

Eigenfunction is defined for an operator in the functional space. For example, oscillations of an elastic object can be described by

$$\varphi'' = L\varphi \tag{5}$$

where $L\varphi$ is some differential expression. If a solution of Eq. (5) has the form $\varphi = T(t)u(x)$, then with respect to function $u(x)$, the following equation holds:

$$L(u) + \lambda u = 0 \tag{6}$$

In a restricted region and under some homogenous conditions on its boundary, parameter λ is called eigenvalue, and non-

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

zero solutions of Eq. (6) are called eigenfunctions. More descriptions of this eigenfunction are given in the sequel (1–7).

Along with the eigenvalues, *singular values* are often used. If a matrix $(m \times n)$ can be transformed in the following form:

$$U^*AV = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix}, \text{ where } S = \text{diag}[\sigma_1, \sigma_2, \ldots, \sigma_r] \tag{7}$$

where U and V are $(m \times m)$ and $(n \times n)$ orthogonal matrix respectively, and all $\sigma_k \geq 0$, then expression (7) is called a singular value decomposition. The values σ_1, σ_2, . . ., σ_r are called singular values of A, and r is the rank of A. If A is a symmetric matrix, then matrices U and V coincide, and σ_k are equal to the absolute values of eigenvalues of A. The singular decomposition (7) is often used in the least square method, especially when A is ill conditioned (1), where condition number of a square matrix is defined as $k(A) = \|A^{-1}\| \cdot \|A\|$; a large $k(A)$ or ill-conditioned A is unwanted when solving linear equations, since a small variation in the system during computation causes a large displacement in the solution.

SOME PROPERTIES OF EIGENVALUES AND EIGENVECTORS

Eigenvectors corresponding to distinct eigenvalues are linearly independent.

Eigenvalues of a real matrix appear as real numbers or complex conjugate pairs.

A symmetric real matrix has all real eigenvalues.

The product of all eigenvalues of A is equal to the determinant of A; in other words,

$$\lambda_1\lambda_2, \ldots, \lambda_n = \det A \tag{8}$$

Eigenvalues of a triangle or diagonal matrix are the diagonal components of the matrix.

The sum of all eigenvalues of a matrix is equal to its *trace;* that is,

$$\lambda_1 + \lambda_2 + \cdots + \lambda_n = \text{tr}A = a_{11} + a_{22} + \cdots + a_{nn} \tag{9}$$

Eigenvalues for A^k are λ_1^k, λ_2^k, . . ., λ_n^k, e.g., eigenvalues for A^{-1} are λ_1^{-1}, λ_2^{-1}, . . ., λ_n^{-1}.

A symmetric matrix A can be put in a diagonal form with eigenvalues as the elements along the diagonal as shown below:

$$A = T\Lambda T^* = T\,\text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n]T^* \tag{10}$$

where $T = [t_{ij}]_{n\times n}$ is the *transformation matrix* and T^* is its complex conjugate transpose matrix, $T^* = [t_{ji}^*]_{n\times n}$.

However non-semi-simple matrices cannot be put into diagonal form, though, they can be put into the so-called *Jordan form.* For a non-semi-simple *multiple eigenvalue* λ, the eigenvector u^1 is dependent on $(m - 1)$ generalized eigenvectors u^2, . . ., u^m:

$$\begin{aligned} Au^1 &= \lambda u^1 \\ Au^2 &= \lambda u^2 + u^1 \\ &\cdots \\ Au^m &= \lambda u^m + u^{m-1} \\ Au^{m+1} &= \lambda_{m+1} u^{m+1} \\ &\cdots \\ Au^n &= \lambda_n u^n \end{aligned} \tag{11}$$

From these equations, the matrix form can be obtained as follows:

$$T^{-1}AT = J \tag{12}$$

where J is a matrix containing a *Jordan block,* and T is the *modal matrix* containing the generalized eigenvectors, $T = [u^1u^2, \ldots, u^m, \ldots, u^n]$. For example, when the number of multiple eigenvalues is 3, matrix J takes the form:

$$J = \begin{bmatrix} \lambda & \delta & & & & & \\ & \lambda & \delta & & & 0 & \\ & & \lambda & & & & \\ & & & \lambda_{m+1} & & & \\ & 0 & & & \lambda_{m+2} & & \\ & & & & & & \lambda_n \end{bmatrix} \tag{13}$$

where $\delta = 0$ or 1 (1).

EIGENVALUE ANALYSIS FOR ORDINARY DIFFERENTIAL EQUATIONS

The eigenvalue approach is applied to *solving the ordinary differential equations* (ODE) given in the following linear form:

$$\frac{dx}{dt} = Ax + Bu \tag{14}$$

where A is the *state matrix* and u is the vector of controls. When A is a matrix with all different eigenvalues λ_i and Eq. (14) is homogeneous (that is $u = 0$), then a solution of Eq. (14) can be found in the following general form:

$$x(t) = \sum_{l=1}^{n} c_i e^{\lambda_i t} \tag{15}$$

where c_i are coefficients that are determined by the initial conditions $x(0)$. For the case of $m < n$ different eigenvalues, the general solution of Eq. (14) for $u = 0$ is

$$x(t) = \sum_{i=1}^{m} \sum_{k=0}^{K_m - 1} c_{ik} t^k e^{\lambda_i t} \tag{16}$$

where K_m is the multiplicity of λ_1. If the system is inhomogeneous (that is u is nonzero), a solution of Eq. (14) can be found as a sum of a general solution for the homogeneous system (15) and (16) and a particular solution of the inhomogeneous system.

The elements of Eqs. (15) and (16) corresponding to each real eigenvalue $\lambda_i = \alpha_i$ or to each pair of complex conjugate eigenvalues $\lambda_i = \alpha_i \pm j\omega_i$ are called *aperiodic and oscillatory modes* of the system motion, respectively. The eigenvalue real part α_i is called damping of the mode i, and the imaginary part ω_i determines the frequency of oscillations.

When A is a matrix with all different eigenvalues, by substituting

$$x = Tx', u = Tu' \tag{17}$$

the original ODE can be transformed into

$$Tdx'/dt = ATx' + Tu' \tag{18}$$

If T is a nonsingular matrix chosen so that

$$T^{-1}AT = \Lambda = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n] \tag{19}$$

we get a modal form of ODE:

$$dx'/dt = T^{-1}ATx' + u' = \Lambda x' + u' \tag{20}$$

In the modal form, state variables x' and equations are independent, and T is the eigenvector matrix. Diagonal elements of the matrix Λ are eigenvalues of A, which can be used to solve ODE (1).

For a general nth order differential equation,

$$A_n \frac{d^n x}{dt^n} + A_{n-1}\frac{d^{n-1}x}{dt^{n-1}} + \cdots + A_1 \frac{dx}{dt} + A_0 x = 0 \tag{21}$$

Besides solving it through transferring it into a set of first order differential equations (1,5), it can also be solved using the original coordinate. The matrix polynomial of system (21) follows:

$$L(\lambda) = A_n\lambda^n + A_{n-1} + \cdots + A_1\lambda + A_0 \tag{22}$$

The solutions and eigenvalues as well as eigenvectors of the system (21) can be obtained by solving the eigenvalue equation:

$$L(\lambda)u = 0 \tag{23}$$

where $L(\lambda)$ is the matrix (22) containing an eigenvalue λ having the corresponding eigenvector u. If vectors u^1, u^2, . . ., u^m, where $m < n$, satisfies the equation:

$$\begin{aligned} &L(\lambda)u^1 = 0 \\ &L(\lambda)u^2 + \frac{1}{1!}\frac{dL(\lambda)}{d\lambda}u^1 = 0 \\ &\cdots \\ &L(\lambda)u^m + \frac{1}{1!}\frac{dL(\lambda)}{d\lambda}u^{m-1} + \cdots + \frac{1}{(m-1)!}\frac{d^{m-1}L(\lambda)}{d\lambda^{m-1}}u^1 = 0 \end{aligned} \tag{24}$$

then

$$x(t) = [t^{m-1}u^1/(m-1)! + \cdots + tu^{m-1}/1! + u^m]e^{\lambda_1} \tag{25}$$

is a solution to the ODE system (1). The set of equations (24) defines the *Jordan Chain* of the multiple eigenvalue λ and the eigenvector u^1.

EIGENVALUES AND EIGENFUNCTIONS FOR INTEGRAL EQUATIONS

An integral equation takes the following general form (1):

$$x(t) = f(t) + \lambda \int_0^t k(\zeta, t)x(\zeta)\, d\zeta \tag{26}$$

In eigenanalysis, we concentrate on the integral equation, which can be rewritten as

$$x(t) = f(t) + \lambda \int_a^b \sum_{i=1}^{n} r_i(\zeta)q_i(\zeta)x(\zeta)\, d\zeta \tag{27}$$

where

$$\int_a^b r_i(\zeta)q_l(\zeta)x(\zeta)\, d\zeta = \text{const.} \tag{28}$$

So Eq. (27) can be reduced to the following problem:

$$x(t) = f(t) + \lambda \sum_{j=1}^{n} x_j q_j(t) \tag{29}$$

By substitution, we have:

$$x_i = \int_a^b r_i(\zeta)[f(t) + \lambda \sum_{j=1}^{n} x_j q_j(t)]\, d\zeta, i = 1, 2, \ldots, n \tag{30}$$

The equation of the system can be obtained as

$$(I - \lambda A)x = b \tag{31}$$

where

$$\begin{aligned} x &= [x_1, x_2, \ldots, x_n]^T \\ A &= [a_{i,j}] = \left[\int_a^b r_l(\zeta)q_j(\zeta)\, d\zeta\right] \\ b &= [b_i] = \left[\int_a^b r_l(\zeta)q_j(\zeta)\, d\zeta\right] \end{aligned}$$

The values of λ which satisfy

$$\text{der}[I - \lambda A] = 0 \tag{32}$$

are the eigenvalues of the integral equation.

To find $x(t)$ by solving an integral equation similar to (26) except for the interval, which is $[a, b]$ instead of $[0, t]$, the eigenfunction approach can also be used. First, $x(t)$ is rewritten as

$$x(t) = f(t) + \sum_{n=1}^{\infty} a_n \phi_n(t) \tag{33}$$

where $\phi_1(t)$, $\phi_2(t)$, . . . are eigenfunctions of the system, and satisfy

$$\phi(t) = \lambda_n \int_a^b k(\zeta, t)\phi_n(\zeta)\, d\zeta \tag{34}$$

where λ_1, λ_2, . . . are eigenvalues of the integral equation. After substituting the eigenfunction into the integral equation and further simplification, the solution $x(t)$ is obtained as

$$x(t) = f(t) + \sum_{n=1}^{\infty} \frac{\lambda f_n}{\lambda_n - \lambda}\phi_n(t) \tag{35}$$

where $f_n = \int_a^b f(\zeta)\phi(\zeta)d\zeta$.

LINEAR DYNAMIC MODELS AND EIGENVALUES

State Space Modeling

In control systems, where the purpose of control is to make a variable adhere to a particular value, the system can be mod-

eled by using the *state space equation* and *transfer functions.* The state space equation is

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{36}$$

where x is the $(n \times 1)$ vector of *state variables,* $\dot{x}$ is its first-order derivative vector, u is the $(p \times 1)$ *control vector,* and y is the $(q \times 1)$ *output vector.* Accordingly, A is the $(n \times n)$ *state matrix,* B is a $(n \times p)$ matrix, C is a $(q \times n)$ matrix, and D is of $(q \times p)$ dimension.

Model Analysis on the Base of Eigenvalues and Eigenvectors

Model analysis is based on the *state space representation* (36). It also explores *eigenvalues, eigenvectors,* and *transfer functions* (8–10).

Consider a case where matrix D is a zero matrix. Then the state space model can be transformed using *Laplace transformation* in a *transfer function* that maps input into output:

$$G(s) = C(sI - A)^{-1}B \tag{37}$$

where s is the Laplace complex variable, and $G(s)$ is composed of denominator $a(s)$ and a numerator $b(s)$:

$$\begin{aligned} G(s) &= b(s)/a(s) \\ &= (b_0 s^n + b_1 s^{n-1} + \cdots + b_n)/(s^n + a_1 s^{n-1} + \cdots + a_n) \end{aligned} \tag{38}$$

The *closed-loop transfer function* for a *feedback system* is

$$G_c(s) = [I + G(s)H(s)]^{-1}G(s) \tag{39}$$

where $H(s)$ is a feedback transfer function.

The system model (36) can be analyzed using the observability and controllability concepts. *Observability* indicates whether all the system's modes can be observed by monitoring only the sensed outputs. *Controllability* decides whether the system state can be moved from an initial point to any other point in the state space within infinite time, and if every mode is connected to the controlled input. The concepts can be described more precisely as follows (8,9,11):

1. For a linear system, if within an infinite time interval, $t_0 < t < t_1$, there exists a piecewise continuous control signal $u(t)$, so that the system states can be moved from any initial mode $x(t_0)$ to any final mode $x(t_1)$, then the system is said to be controllable at the time t_0. If every system mode is controllable, then the system is state controllable. If at least one of the states is not controllable, then the system is not controllable.
2. For a linear system, if within an infinite time interval, $t_0 < t < t_1$, every initial mode $x(t_0)$ can be observed exclusively by the sensed value $y(t)$, then the system is said to be fully observable.

Matrix transformations are required to assess observability and controllability. To study controllability, it is necessary to introduce the control canonical form as follows:

$$A_c = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_n \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 \end{bmatrix}, B_c = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$
$$C_c = [b_1 \quad b_2 \quad \dots \quad b_n], \quad D_c = 0 \tag{40}$$

where the subscript c denotes that the associated matrix is in control canonical form.

For a linear time-invariant system, the necessary and sufficient condition for the system state controllability is the full rank of controllability matrix Q_c. The controllability matrix is

$$Q_c = [B \vdots AB \vdots \cdots \vdots A_{n-1}B] \tag{41}$$

and the system is controllable if and only if Rank $Q_c = n$.

When the linear time-invariant system has distinct eigenvalues, then after the modal transformation, the new system becomes

$$\dot{z} = T^{-1}ATz + T^{-1}Bu \tag{42}$$

where T^{-1}AT is diagonal matrix. Under such condition, the sufficient and necessary conditions for state controllability is that there are no rows in the matrix $T^{-1}B$ containing all zero elements.

When matrix A has multiple eigenvalues, and every multiple eigenvalue corresponds to the same eigenvector, then the system can be transformed into the new state space form, which is called the *Jordan canonical form:*

$$\dot{z} = Jz + T^{-1}Bu \tag{43}$$

where the matrix J is Jordan canonical matrix. Then the sufficient and necessary condition for state controllability is that not all the elements in the matrix $T^{-1}B$, corresponding to the last row of every Jordan sub-matrix in matrix J, are zero.

The *output controllability sufficient and necessary condition* for linear time-invariant system is that the matrix $[CB \vdots CAB \vdots \cdots \vdots CA^{n-1}B \vdots D]$ is full rank; that is,

$$\text{rank}[CB \vdots CAB \vdots \cdots \vdots CA^{n-1}B \vdots D] = n \tag{44}$$

Similarly, the sufficient and necessary observability condition for linear time-invariant system is that the observability matrix is full rank; that is,

$$Q_D = [C \vdots CA \vdots \cdots \vdots CA^{n-1}]^T \tag{45}$$

and *rank* $Q_0 = n$. When the system has distinct eigenvalues, then after a linear nonsingular transformation, the system takes the form (when control vector u is zero):

$$\begin{aligned} \dot{z} &= T^{-1}ATz \\ y &= CTz \end{aligned} \tag{46}$$

then the condition for observability is that there are no rows in the matrix CT which have only zero elements. Even though the system has multiple eigenvalues, and every multiple ei-

genvalue corresponds to the same eigenvector, the system after transformation looks like

$$\begin{aligned} \dot{z} &= Jz \\ y &= CTz \end{aligned} \tag{47}$$

where J is the Jordan matrix. The observability condition is that there are no columns corresponding to the first row of each Jordan submatrix having only zero elements.

EIGENVALUES AND STABILITY

Since the time-dependent characteristic of a mode corresponding to an eigenvalue λ_i is given by $e^{\lambda_i t}$, the stability of the system matrix can be determined by the eigenvalues of the system state matrix, as in the following (see Ref. 37).

A real eigenvalue corresponds to a nonoscillatory mode. A negative real eigenvalue represents a decaying mode. The larger its magnitude, the faster the decay. A positive real eigenvelue represents aperiodic instability.

Complex eigenvalues occur in conjugate pairs, and each pair corresponds to an oscillatory mode. The real component of the eigenvalues gives the damping, and the imaginary component defines the frequency of oscillation. A negative real part indicates a damped oscillation and a positive one represents oscillation of increasing amplitude. For a complex pair of eigenvalues, $\lambda = -\sigma \pm j\omega$, the frequency of oscillation in hertz can be calculated by

$$f = \omega/2\pi \tag{48}$$

which represents the actual or damped frequency. The damping ratio is given by

$$\zeta = \frac{\sigma}{\sqrt{\sigma^2 + \overline{\omega}^2}} \tag{49}$$

From the point of view of a system modeled by a transfer function, the concept of natural frequency is given based on complex poles which correspond to the complex eigenvalues of the state matrix A, as in Eq. (36). Let the complex poles be $s = -\sigma \pm j\omega$, and the denominator corresponding to them be $d(s) = (s + \sigma)^2 + \omega^2$. Then its transfer function is represented in polynomial form as $H(s) = \omega_n^2/(s^2 + 2\zeta\omega_n s + \omega_n^2)$, where $\sigma = \zeta\omega_n$ and $\omega = \omega_n\sqrt{(1 - \zeta^2)}$. This introduces the definition of the undamped natural frequency, ω_n, and again the damping ratio, ζ.

More fundamentally, the Lyapunov stability theory forms a basis for stability analysis. There are two approaches to evaluate system stability (4,8,9,11,12):

1. the first Lyapunov method and
2. the second Lyapunov method.

The *first Lyapunov method* is based on eigenvalue and eigenvector analysis for linearized systems and small disturbances. It finds its application in many areas, for example, in the area of power systems engineering.

To study small-signal stability, it is necessary to clarify some basic concepts regarding the following *Differential-Algebraic Equation* (DAE):

$$\begin{aligned} \dot{x} &= f(x, y, p) \quad f : R^{n+m+q} \to R^n \\ 0 &= g(x, y, p) \quad g : R^{m+n+q} \to R^m \end{aligned} \tag{50}$$

where $x \subset R^n$, $y \subset R^m$, $p \subset R^q$; x is the vector of dynamic state variables, y is the vector of static or instantaneous state variables, and p is a selected system parameter affecting the studied system behavior. Variable y usually represents a state variable whose dynamics is instantaneously completed as compared to that of the dynamic state variable x. Parameter p belongs to the system parameters which have no dynamics at all at least if modeled by Eq. (50) (13). For example, in power system engineering, typical dynamic state variables are chosen from the time-dependent variables such as machine angle and machine speed. The static variables are the load flow variables including bus voltages and angles. Parameter p can be selected from static load powers, or control system parameters.

A system is said to be in its *equilibrium condition* when the derivatives of its state variables are equal to zero, which means there is no variation of the state variables. For the system modeled in Eq. (50), this condition is given as follows:

$$\begin{aligned} 0 &= f(x, y, p) \\ 0 &= g(x, y, p) \end{aligned} \tag{51}$$

Solutions (x_0, y_0, p_0), of the preceding system are the system *equilibrium points*. Small-signal stability analysis uses the system represented in linearized form, which is done by differentiating the original system respect to the system variables and parameters around its equilibrium point (x_0, y_0, p_0). This linearization is necessary for the Lyapunov method and via computing system eigenvalues and eigenvectors.

For the original system (50), its linearized form is given in the following:

$$\begin{aligned} \Delta\dot{x} &= \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial y}\Delta y \\ 0 &= \frac{\partial g}{\partial x}\Delta x + \frac{\partial g}{\partial y}\Delta y \end{aligned} \tag{52}$$

For simplicity, system (52) is rewritten as

$$\begin{aligned} \Delta\dot{x} &= A\Delta x + B\Delta y \\ 0 &= C\Delta x + D\Delta y \end{aligned} \tag{53}$$

where matrices A, B, C, and D are the partial derivatives' matrices. If the algebraic matrix D is not singular (i.e., det $D \neq 0$), the state matrix A_s is given as

$$A_s = A - BD^{-1}C \tag{54}$$

which is studied in stability analysis using the eigenvalue and eigenvector approach.

The use of the first Lyapunov method involves the following steps (12,14,15):

1. linearization of the original system (50) as in (52);
2. elimination of the algebraic variables to form the reduced dynamic state matrix A_s;

3. computation of the eigenvalues and eigenvectors of the state matrix A_s;
4. stability study of the system (16):
   a. If eigenvalues of the state matrix are located in the left-hand side of the complex plane, then the system is said to be small-signal stable at the studied equilibrium point;
   b. If the rightmost eigenvalue is zero, the system is on the edge of small-signal aperiodic instability;
   c. If the rightmost complex conjugate pair of eigenvalues has a zero real part and a nonzero imaginary part, the system is on the edge of oscillatory instability depending on the transversality condition (17);
   d. If the system has eigenvalue with positive real parts, the system is not stable;
   e. For the stable case, analyze several characteristics including damping and frequencies for all modes, eigenvalue sensitivities to the system parameters, excitability, observability, and controllability of the modes.

More precise definitions for the first Lyapunov method have been addressed in the literature (16,18). The general time-varying or nonautonomous form is as follows:

$$\dot{x} = f(t, x, u) \tag{55}$$

where t represents time, x is the vector of state variables, and u is the vector of system input. In a special case of the system (55), f is not explicitly dependent on time t; that is,

$$\dot{x} = f(x) \tag{56}$$

and the system is said to be autonomous or time-invariant. Such a system does not change its behavior at different times (16).

An equilibrium point x_0 of the autonomous system (56) is

1. stable if, for each $\epsilon > 0$, there exists $\delta = \delta(\epsilon) > 0$ such that $\|x(0)\| < \delta \Rightarrow \|x(t)\| < \epsilon$, $\forall t \geq 0$;
2. unstable otherwise;
3. asymptotically stable, if it is stable, and δ can be chosen such that: $\|x(0)\| < \delta \Rightarrow \lim_{t\to\infty} x(0) = x_0$.

The definition can be represented in the form of eigenvalue approach as given in the Lyapunov first-method theorem.

Let x_0 be an equilibrium point for the autonomous system (54), where f: $D \to R^n$ is continuously differentiable and D is a neighborhood of the origin. Let the system Jacobian be $A = (\partial f/\partial x)(x)|_{x=x_0}$, and $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]$ be the eigenvalues of A, then the origin is asymptotically stable if Re $\lambda_i < 0$ for all eigenvalues of A, or the origin is unstable if Re $\lambda_i > 0$ for one or more eigenvalues of A.

Let us take one step further. The stability definition for a time-varying system, where the system behavior depends on the origin at the initial time t_0, is as follows. The equilibrium point x_0 for the system (55) is (16)

1. stable, if for each $\epsilon > 0$, there exists $\delta = \delta(\epsilon, t_0) > 0$ such that $\|x(t_0)\| < \delta \Rightarrow \|x(t)\| < \epsilon$, $\forall t \geq t_0 \geq 0$;
2. uniformly stable, if for each $\epsilon > 0$, there exists $\delta = \delta(\epsilon) > 0$, such that $\|x(t_0)\|\ \delta \Rightarrow \|x(t)\| < \epsilon$, $\forall t \geq t_0 \geq 0$;
3. unstable otherwise;
4. asymptotically stable, if it is stable and there is $c = c(t_0) > 0$ such that, for all $\|x(t_0)\| < c$, $\lim_{t\to\infty} x(t) = x_0$;
5. uniformly asymptotically stable if it is uniformly stable and there is a time invariant $c > 0$, such that for all $\|x(t_0)\| < c$, $\lim_{t\to\infty} x(t) = x_0$. This holds for each $\epsilon > 0$, if there is $T = T(\epsilon) > 0$, such that $\|x(t)\| < \epsilon$, $\forall t \geq t_0 + T(\epsilon)$, $\forall \|x(t_0)\| < c$;
6. globally uniformly asymptotically stable if it is uniformly stable and for each pair of positive numbers ϵ and c, there is a $T = T(\epsilon, c) > 0$, such that $\|x(t)\| < \epsilon$, $\forall t \geq t_0 + T(\epsilon, c)$, $\forall \|x(t_0)\| < c$.

The corresponding stability theorem follows.

Let $f(t, x, u)|_{(t^*,x^*,u^*)} = 0$ be an equilibrium point for the nonlinear time-varying system (55), where f: $[0, \infty) \times D \to R^n$ is continuously differentiable, $D = \{x \in R^n | \|x\|_2 < r\}$, the Jacobian matrix is bounded and Lipschitz on D, uniformly in t. $A(t) = (\partial f/\partial x)(t, x)|_{x=x_0}$ is the Jacobian; then the origin is exponentially stable for the nonlinear system if it is an exponentially stable equilibrium point for the linear system $\dot{x} = A(t)x$.

The second Lyapunov method isa potentially most reliable and powerful method for the original nonlinear and nonautonomous (or time-varying) systems. But it relys on the Lyapunov function, which is hard to find for many physical systems.

EIGENVALUES AND BIFURCATIONS

The bifurcation theory has a rich mathematical description and literature for various areas of applications. Many physical systems can be modeled by the general form

$$x' = f(x, p) \tag{57}$$

where x is vector of the system state variables, and p is the system's parameter, which may vary during system operation in normal as well as contingency conditions.

Bifurcations occur where, by slowly varying certain system parameters in some direction, the system properties change qualitatively or quantitatively at a certain point (14,19). Local bifurcations can be detected by monitoring the behavior of eigenvalues of the systems operation point. In some direction of parameter variation, the system may become unstable because of the singularity of the system dynamic state matrix associated with zero eigenvalue or because of a pair of complex conjugate eigenvalues crossing the imaginary axes of the complex plane. These two phenomena are saddle node and Hopf bifurcations, respectively. Other conditions that may drive the system state into instability may also occur. These include singularity-induced bifurcations, cyclic fold, period doubling, and blue sky bifurcations or even chaos (15,19,20).

For the general system (57), a point (x_0, p_0) is said to be a *saddle node bifurcation* point if it is an equilibrium point of the system; in other words, $f(x_0, p_0) = 0$, the system Jacobian matrix, $f_y(x_0, p_0)$ has a simple zero eigenvalue $\lambda(p_0) = 0$, and

the transversal conditions hold (17,21). More generally (24), the saddle bifurcation satisfy the following conditions:

1. The point is the system's equilibrium point [i.e., $f(x_0, p_0) = 0$].
2. The Jacobian matrix, $f_y(x_0, p_0)$ has a simple and unique eigenvalue $\lambda(p_0) = 0$ with the corresponding right and left eigenvectors l and r, respectively.
3. Transversality condition of the first-order derivative: $l^T f_y(x_0, p_0) \neq 0$
4. Transversality condition of the second-order derivative: $l^T[f_{yy}(x_0, p_0)r]r \neq 0$

A Hopf bifurcation occurs when the following conditions are satisfied:

1. The point is a system operation equilibrium point [i.e., $f(x_0, p_0) = 0$];
2. The Jacobian matrix $f_y(x_0, p_0)$ has a simple pair of pure imaginary eigenvalues $\lambda(p_0) = 0 \pm j\omega$ and no other eigenvalues with zero real part;
3. Transversality condition: $d[\text{Re } \lambda(p_0)]/dp \neq 0$.

The last condition guarantees the transversal crossing of the imaginary axis. The sign of $d[\text{Re } \lambda(p_0)]/dp$ determines whether there is a birth or death of a *limit cycle* at (x_0, p_0). Depending on the direction of transversal crossing the imaginary axis, Hopf bifurcation can be further categorized into supercritical and subcritical ones. The *supercritical Hopf bifurcation* happens when the critical eigenvalue moves from the left half plane to the right half plane. The *subcritical Hopf bifurcation* occurs when the eigenvalue moves from the left half plane to the right half plane and is unstable. The system transients are diverged into an oscillatory style at the vicinity of the subcritical Hopf bifurcation points.

Singularity-induced bifurcations occur when the system's equilibrium approaches singularity, and some of the system eigenvalues become unbounded along the real axis (i.e., $\lambda_i \to \infty$). In case of the DAE model (50), the singularity of the algebraic Jacobian $D = g_y$ causes the singularity-induced bifurcations. In that case, singular perturbations or noise techniques must be used to analyze the system dynamics (22). When singularity-induced bifurcation occurs, the system behavior becomes hardly predictable and may cause fast claps type instability (22).

A graphical illustration of these three major bifurcations is given in Fig. 1.

Methods of computing bifurcations can be categorized into direct and indirect approaches. The direct method has been practiced by many researchers in this area (13–15,17,19, 20,22–32). For example, the direct method computes the Hopf bifurcation condition by solving directly the set of equations (15,17,20,26):

$$f(x, p_0 + \tau\Delta p) = 0 \tag{58}$$

$$A_s'(x, p_0 + \tau\Delta p)l' + \omega l'' = 0 \tag{59}$$

$$A_s'(x, p_0 + \tau\Delta p)l'' - \omega l' = 0 \tag{60}$$

$$\|l\| = 1 \tag{61}$$

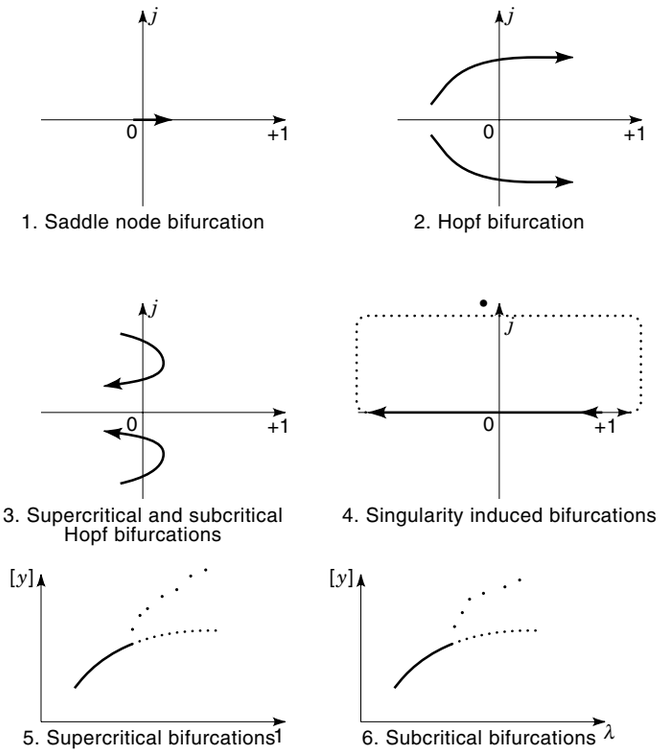


Figure 1. Bifurcation diagrams for different bifurcations. +1, real axis; j, imaginary axis; 1–4: eigenvalue trajectories as a result of system parameter variation; 5, 6: system state variable branch diagrams. The branching properties of the system state variable movement determine the type of bifurcations.

where $A_s = A - BD^{-1}C = f_x - f_y g_y^{-1} g_x$ is the state matrix, $0 + j\omega$ is its eigenvalue, $l = l' + jl''$ is the corresponding left eigenvector, $p = p_0 + \tau\Delta p$ is the system parameter vector varying from the point p_0 in direction Δp. By taking zero ω and l'', saddle node bifurcation can be computed as well.

Indirect methods are mainly Newton–Raphson type method using predictor and corrector to trace the bifurcation diagram. A detailed description of the continuation methods can be found in Refs. 17, 23, 25–27, and 33.

As an example of applied bifurcation analysis, let us consider a task from the area of power system analysis (20,26). The power system model is composed of two generators and one load bus. The system is shown in Fig. 2.

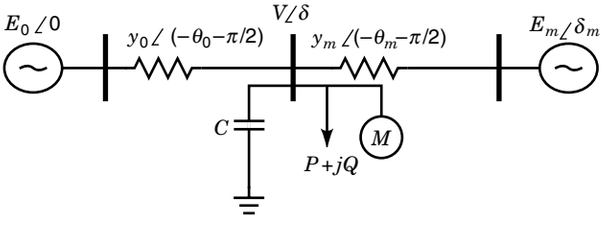


Figure 2. A simple power system model. The system dynamics are introduced mainly by the induction motor and generators.

Static and an induction motor load are connected with the load bus in the middle of the network. A capacitor device is also connected with the same bus to provide reactive power supply and control the voltage magnitude; E and δ_m are generator terminal voltage and angle, respectively; V is load bus voltage; δ is load bus voltage angle; Y is line conductance; and M stands for induction motor load. The system is modeled by the following equations:

$$\begin{aligned}
\delta'_m &= \omega \\
M\dot{\omega} &= -\delta_m\omega + P_m \\
&\quad + E_m y_m V \sin(\delta - \delta_m - \theta_m) \\
&\quad + E_m^2 y_m \sin\theta_m \\
K_{qw}\dot{\delta} &= -K_{qv2}V^2 - K_{qv}V + E'_0 y'_0 V \cos(\delta + \theta''_0) \\
&\quad + E_m y_m V \cos(\delta - \delta_m + \theta_m) \\
&\quad - (y'_0 \cos\theta'_0 + y_m \cos\theta'_m V^2 - Q_0 - Q_1 \\
TK_{qw}K_{pv}\dot{V} &= K_{qw}K_{qv}^2 V^2 + (K_{pw}K_{qv} - K_{qw}K_{pv})V \\
&\quad + \sqrt{(K_{qw}^2 + K_{pw}^2)}[-E'_0 y'_0 V \cos(\delta + \theta'_0 - h) \\
&\quad - E_m y_m V \cos(\delta - \delta_m + \theta_m - \eta) + (y'_0 \cos(\theta'_0 - \eta) \\
&\quad + y_m \cos(\theta_m - \eta))V^2] - K_{qw}(P_0 + P_1) \\
&\quad + K_{pw}(Q_0 + Q_1)
\end{aligned}$$

where $\eta = \tan^{-1}(K_{qw}/K_{pw})$. The active and reactive loads are featured by the following equations:

$$\begin{aligned}
P_d &= P_0 + P_1 + K_{pw}\delta + K_{pv}(V + TV') \\
Q_d &= Q_0 + Q_1 + K_{qw}\delta + K_{qv}V + K_{qv2}V^2
\end{aligned}$$

The system parameter Q_1 is selected as the bifurcation parameter to be increased slowly. Voltage V is taken as a dependent parameter for illustration. Figure 3 shows the dynamics for the system in the form of a Q–V curve (19).

The eigenvalue trajectory around a Hopf bifurcation point is given in Fig. 4 where both supercritical and subcritical Hopf bifurcations can be seen.

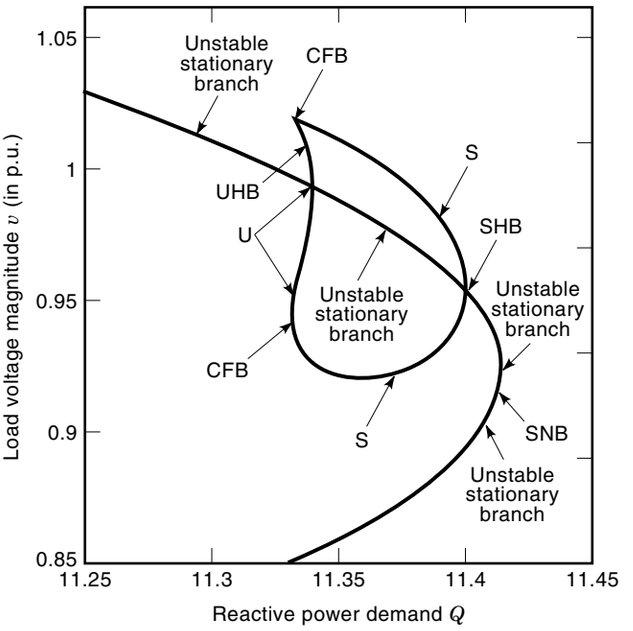


Figure 3. The Q–V curve branch diagrams. S—stable periodic branch; U—unstable periodic branch; SNB—saddle node bifurcation; SHB—stable (supercritical) Hopf bifurcation; UHB—unstable (subcritical) Hopf bifurcation; CFB—cyclic fold bifurcation. These bifurcations are associated with system eigenvalue behavior while the reactive load power Q_1 is consistently increased. This shows that for a simple dynamic system, as given in Fig. 3, stability-related phenomena are very rich.

NUMERICAL METHODS FOR THE EIGENVALUE PROBLEM

Computing Eigenvalues and Eigenvectors

Although roots of the characteristic polynomial $L(\lambda) = a_n\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0$ are eigenvalues of the matrix A, a direct calculation of these roots is not recommended because of the rounded errors and high sensitivity of the roots to coefficients a_i (1).

We start by introducing the *power method,* which locates the largest eigenvalue. Suppose a matrix A has eigenvalues $\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n]^T$ and the corresponding right eigenvectors $R = [r_1, r_2, \ldots, r_n]^T$, and $|\lambda 1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$. For any vector $x \neq 0$, we have

$$x = \sum_{i=1}^{n} c_i r_i \tag{62}$$

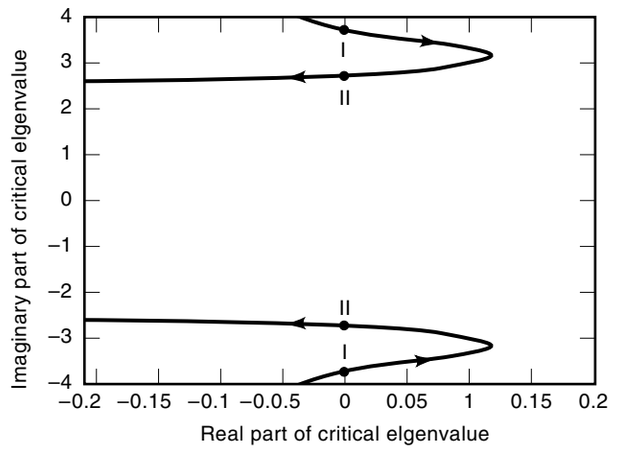


Figure 4. An illustration of subcritical (I) and supercritical (II) Hopf bifurcations. (I) corresponds to movement of the eigenvalue real part from the left to the right side of the s-plane; (II) indicates a reverse movement.

By multiplying (62) by A, A^2, . . ., it can be obtained that

$$\begin{aligned} x^{(1)} &= Ax = \sum_{i=1}^{n} c_i \lambda_1 r_i \\ x^{(2)} &= Ax^{(1)} = \sum_{i=1}^{n} c_i \lambda_i^2 r_i \\ &\cdots \\ x^{(m)} &= Ax^{(m-1)} = \sum_{i=1}^{n} c_i \lambda_i^m r_i \end{aligned} \tag{63}$$

After a number of iterations, $x^{(m)} \to \lambda_1^m c_1 r_1$. Therefore, λ_1 can be obtained by dividing the corresponding elements of $x^{(m)}$ and $x^{(m-1)}$ after a sufficient number of iterations, and the eigenvector can be obtained by scaling $x^{(m)}$ directly. Other eigenvalues and eigenvectors can be computed by applying the same method to the new matrix:

$$A_1 = A - \lambda_1 r_1 v^{1*} \tag{64}$$

where v^1 is the reciprocal vector of the first eigenvector r_1. It can be observed that the matrix A_1 has the same eigenvalues as A except the first eigenvalue, which is set to zero by the transformation. By applying the method successively, all eigenvalues and corresponding right eigenvectors of matrix A can be located. The applicability of this method is restricted by computational errors. Convergence of the method depends on separation of eigenvalues determined by the ratios $\|\lambda_i/\lambda_1\|$, $\|\lambda_i/\lambda_2\|$, etc. As evident, the method can compute only one eigenvalue and eigenvector at a time.

The *Schur algorithm* can also be used to locate eigenvalues from λ_2 while knowing λ_1 by applying the power method to A_1 after the following transformation (1):

$$\begin{bmatrix} \lambda_1 & B_1 \\ 0 & A_1 \end{bmatrix} \begin{bmatrix} 1 \\ y^{(2)} \end{bmatrix} = \lambda_2 \begin{bmatrix} 1 \\ y^{(2)} \end{bmatrix} \tag{65}$$

A general idea of the *inverse power method* is to use the power method determining the minimum eigenvalues. By shifts, any eigenvalue can be made the minimum one. The inverse method can compute eigenvectors accurately even when the eigenvalues are not well separated. The method implies the following. Let λ_i^* be an approximation of one of the eigenvalues λ_i of A. The steps involved follow:

- Obtain a tridiagonal matrix T by reduction of matrix A;
- Find zeros of $(T - \lambda_i^* I)^{-1} y_0$, i.e. $\{z_1 | (T - \lambda_i^* I) z_1 = y_0\}$;
- Set $y_1 = z_1/\|z_1\|$;
- Solve $(T - \lambda_i^* I)^{-1} z_2 = y_1$, etc.

The eigenvector l_i of λ_i is approximated by $y_k = z_k/\|z_k\|$ provided y_0 contains a nonzero term in l_i. If $|\lambda_i^* - \lambda_i|$ is sufficiently small, the inverse iteration method obtains the eigenvector associated with λ_i within only several iterations.

The *QR method* is one of the most popular algorithms for computing eigenvalues and eigenvectors. By using a factorization of the product of a unitary matrix Q and an upper-triangular matrix R, this method involves the following iteration process:

$$\begin{aligned} A_1 &= RQ = Q_1 R_1 \\ &\cdots \\ A_i &= R_{i-1} Q_{i-1} = Q_i R_i = Q_{i-1}^{-1} A_{l-1} Q_{l-1} \end{aligned} \tag{66}$$

where the tth unitary matrix Q_t is obtained by solving

$$Q_t^{\mathrm{T}} A = R_t$$

and Q_t^{T} is determined in a factorized form, such as the product of plane rotations or of elementary Hermitians. Then, the matrix $R_t Q_t$ is obtained by successive post-multiplication of R_t with the transposition of the factors of Q_t^{T} (5).

After a number of iterations, diagonal elements of R_m approximate eigenvalues for A (1). To reduce the number of iterations and speed up computations because less computational effort is required at each iteration, the studied matrix A is initially reduced to the Hessenberg form, which is preserved during iterations (1,5).

A more *general form of the eigenvalue problem* can be modeled as

$$Ax = \lambda Bx \tag{67}$$

If A and B are nonsingular, the problem can be transformed into the standard form of eigenvalue problems by expressing

$$B^{-1}Ax = \lambda x \quad \text{or} \quad A^{-1}Bx = \lambda^{-1}x \tag{68}$$

Then methods discussed earlier can be applied to solve the problem.

There are cases when the computation can be simplified (1). When both A and B are symmetric and B is positive definite, matrix B can be decomposed as $B = C^T\mathrm{C}$ where C is a nonsingular triangular matrix. Then the problem can be expressed as

$$Ax = \lambda C^T C x \tag{69}$$

If vector y is chosen so that $y = Cx$, the final transformation is obtained as

$$(C^{\gamma})^{-1} A C^1 y = Gy = \lambda y \tag{70}$$

and the problem is simplified into the eigenvalue problem with matrix G. Techniques dealing with other situations of the generalized eigenvalue problem can be found in Ref. 34.

In many cases, matrix A is a sparse matrix with many zero elements. Different techniques solving the *sparse matrix eigenvalue problem* are derived. The approaches can be categorized into two major branches: (1) problems where the *LU factorization* is possible, or (2) where it is impossible.

In the first case, after transformation of the generalized eigenvalue problem as $B^{-1}Ax = \lambda x$, or $y = L^T x$, so that $L^{-1}AL^{-T}y = \lambda y$, the resulting matrices may not necessarily be sparse. There are several aspects of the problem. First, the matrix should be represented in such a way that it dispenses with zero elements and allows new elements to be inserted as they are generated by the elimination process during the decomposition; second, pivoting must be performed during the elimination process to preserve sparsity and ensure numerical stability (31,35). The power method is sometimes used for large sparse matrix problems to compute eigenvalues.

When matrices A and B become very large, performing the LU factorization for the general eigenvalue problem becomes

more and more difficult. In this case, a function should be constructed so that it reaches its minimum at one or more of the eigenvectors, and the problem is to minimize this function with an appropriate numerical method (31). For example, the successive search method can be used to minimize this function. Also, other gradient methods can be employed.

Among all these computation methods for eigenvalue problems, many factors influence the efficiency of a particular method. For a large matrix, whether it is dense or sparse, the power method is suitable when only a few large eigenvalues and corresponding eigenvectors are required. The inverse iteration method is the most robust and accurate in calculating eigenvectors. Nevertheless, the most popular general method for eigenvalue and eigenvector computations is the QR method. However, in many cases, especially when the matrix is Hermitian or real symmetric, many methods can provide satisfactory results.

Localization of Eigenvalues

Along with the direct method based on computation of eigenvalues, there are several indirect methods to determine a domain in the complex plane where the eigenvalues are located. A particular interest for the stability studies is to decide whether all eigenvalues have negative real parts. Some methods can also count the number of stable and unstable modes without solving the general eigenvalue problem. Also, there are methods that determine a bounded region where the eigenvalues are located.

The following algebraic results can help to identify the stability of a matrix (4):

- If the matrix $A \in R_{n\times n}$ is stable and $W \in R_{n\times n}$ is positive (nonnegative) definite, then there exists a real positive (positive or nonnegative) definite matrix V such that $AV' + VA = -W$.
- Let $V \in R_{n\times n}$ be positive definite, define the real symmetric matrix W by $A'V + VA = -W$. Then A is stable if for the right eigenvector r associated with every distinct eigenvalue of A, there holds the relation $r^*Wr > 0$ where r^* means conjugate transpose of eigenvector r.
- If W is positive definite, then A is stable if $A'V + VA = -W$ has a positive definite solution matrix V.

Also, the stability problem can be studied by locating the eigenvalues using coefficients of the characteristic polynomial $\det(\lambda I - A) = 0$ rather then the matrix itself. The *Routh–Hurwitz criterion* is one of these approaches. For the monic polynomial with real coefficients,

$$f(z) = z^n + a_1 z^{n-1} + \cdots + a_n \tag{71}$$

and the *Hurwitz matrices* are defined as

$$\begin{aligned} H_1 &= a_1 \\ H_2 &= \begin{bmatrix} a_1 & 1 \\ a_3 & a_2 \end{bmatrix} \\ &\cdots \\ H_n &= \begin{bmatrix} a_1 & 1 & 0 & 0 & & 0 \\ a_3 & a_2 & a_1 & 1 & & 0 \\ a_3 & a_4 & a_3 & a_2 & & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ a_{2n-1} & \cdot & \cdot & & a_{n+1} & a_n \end{bmatrix} \end{aligned} \tag{72}$$

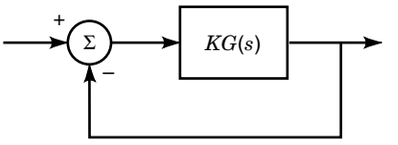


Figure 5. Block diagram for the feedback system: $Y(s)/R(s) = H(s) = KG(s)/[1 + KG(s)]$.

The criteria say that all the zeros of the polynomial $f(z)$ have negative real parts iff $\det H_i > 0$, for $i = 1, 2, \ldots, n$. This also indicates that the eigenvalues λ of the matrix associated with the characteristic polynomial $f(\lambda)$ have all negative real parts, so the matrix is stable.

The *Nyquist stability criterion* is another indirect approach to evaluating stability conditions. For the feedback system given in Fig. 5, it relates the system open loop frequency response to the number of closed-loop poles in the right half of the complex plane (8).

Stability of the system is analyzed by studying the Nyquist plot (polar plot) of the open loop transfer function $KG(j\omega)$. Because it is based on the poles of the closed-loop system, which is decided by $1 + KG(s) = 0$, the point -1 is the critical point for study of the curve $KG(j\omega)$ in the polar plot. The following steps are involved. First, draw the magnitude and angle of $KG(j\omega)$. Second, count the number of clockwise encirclements of -1 as N. Third, find the number of unstable poles of $G(s)$, which is P. The system is stable if the number of unstable closed-loop roots $Z = N - P$ is zero, which means that there are no closed-loop poles in the right half plane. There are other methods exploiting godographs of the system transfer function as functions of ω.

The *Gershgorin's theorem* is also used in eigenvalue localization. It states that any of the eigenvalues of a matrix $A = [a_{i,j}]_{n\times n}$ lies in at least one of the circular discs with centers $a_{i,i}$ and radii as sum of $\|a_{i,j}\|$ for all $i \neq j$. If there are s such circular discs forming a connected domain isolated from other discs, then A has exactly s eigenvalues within this domain (5,36). This theorem finds its application in perturbation analysis of eigenvalues.

Mode Identification

Identification of a mode of a system finds its application in many engineering tasks. Based on nonlinear simulations or measurements, system identification techniques can be used for this purpose. The least-squares method is among those widely used. The major approaches in system modeling and identification include system identification based on an FIR (MA) system model, system identification based on all All-Pole (AR) system model, and system identification based on a Pole-Zero (ARMA) system model. As one of the typically used methods in identfiying modes of a dynamic system, Prony's method is a procedure for fitting a signal $y(t)$ to a weighted sum of exponential terms of the form:

$$\hat{y}(t) = \sum_{i=1}^{n} R_i e^{\lambda_i t} \tag{73a}$$

or in a discrete form:

$$\hat{y}(k) = \sum_{i=1}^{n} R_i z_i^k \tag{73b}$$

where $\hat{y}(t)$, $\hat{y}(k)$ are the Prony approximation to $y(t)$, R_i is signal residue, λ_i is the s-plane mode, z_i is the z-plane mode, and n is the Prony fit order. Supposing that the signal y is a linear function of past values, the modes and signal residues can be calculated by the following equation:

$$y(k) = a_1 y(k-1) + a_2 y(k-1) + \cdots + a_n y(k-n) \tag{74}$$

which can be applied repeatedly to form the linear set of equations as shown in Eq. (75), where N is the number of sample points:

$$\begin{bmatrix} y(n+0) & y(n-1) & \cdots & y(1) \\ y(n+1) & y(n+0) & \cdots & y(2) \\ & & \cdots & \\ y(N-1) & y(N-2) & \cdots & y(N-n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_n \end{bmatrix} = \begin{bmatrix} y(n+1) \\ y(n+2) \\ \cdots \\ y(N) \end{bmatrix} \tag{75}$$

From Eq. (75), the coefficients a_i can be calculated. The modes z_i are the roots of the polynomial: $z^n - a_1 z^{n-1} - \cdots - a_n = 0$. The signal residues R_i can be calculated by solving the linear equations:

$$\begin{bmatrix} z_1^1 & z_2^1 & \cdots & z_n^1 \\ z_1^2 & z_2^2 & \cdots & z_n^2 \\ & & \cdots & \\ z_1^N & z_2^N & \cdots & z_n^N \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ \\ R_n \end{bmatrix} = \begin{bmatrix} y(1) \\ y(2) \\ \\ y(N) \end{bmatrix} \tag{76}$$

from which the s-plane modes λ_i can be computed by $\lambda_i = \log_e(z_i)/\Delta t$, where Δt is the sampling time interval (36a). A similar estimation method is the Shanks' method, which employs a least-squares criterion (36b).

SOME PRACTICAL APPLICATIONS OF EIGENVALUES AND EIGENVECTORS

Some Useful Comments

In the area of stability and control, eigenvalues give such important information as *damping, phase,* and *magnitude* of oscillations (15,20,37). For example, for the system dynamic state matrix A_s critical eigenvalue $\lambda_i = \alpha_i \pm j\omega_i$, which is the eigenvalue with the largest real part α_i, the damping constant is $\sigma = \alpha_i$, and frequency of oscillation is ω_i in radius per second unit, or $\omega_i/2\pi$ in hertz.

The *eigenvalue sensitivity analysis* is often needed to assess the influence of certain system parameters p on damping and enhance system stability (2,15):

$$\frac{\partial \alpha_j}{\partial p_i} = \mathrm{Re}\left\{ \frac{l_j^T \dfrac{\partial A_s}{\partial p_i} r_j}{l_j^T r_j} \right\} \tag{77}$$

l_j and r_j are the corresponding left and right eigenvectors for the jth eigenvalue α_j, and $\partial A_s/\partial p_i$ is the sensitivity of the dynamic state matrix to the ith parameter p_i.

The left and right vectors are also associated with important features of the system dynamics.

The left eigenvector is a normal vector to the equal damping surfaces, and the right eigenvector shows the initial dynamics of the system at a disturbance (15). They also provide an efficient mathematical approach to locating these equal damping surfaces in the parameter spaces (15,20).

The elements of the right and left eigenvectors are dependent on units and scaling associated with the state variables. This may cause difficulties when these eigenvectors are applied individually for identification of the relationship between the states and the modes. The participation matrix P is needed to solve the problem. The participation matrix combines the right and left eigenvectors and can serve as a measure of the association between the state variables and the modes. It is defined as

$$P = [P_1 P_2, \ldots, P_n] \qquad \text{with} \qquad P_i = \begin{bmatrix} P_{1i} \\ P_{2i} \\ \vdots \\ P_{ni} \end{bmatrix} = \begin{bmatrix} \rho_{1i}\vartheta_{i1} \\ \rho_{2i}\vartheta_{i2} \\ \vdots \\ \rho_{ni}\vartheta_{in} \end{bmatrix} \tag{78}$$

where ρ_{ki} is the kth entry of the right eivengector r_i, and θ_{ik} is the kth entry of the left eigenvector l_i. The element is the participation factor, which measures the relative participation of the kth state variable in the ith mode, and vice versa. Regarding the eigenvector normalization, the sum of the participation factors associated with any mode ($\sum_{i=1}^{n} P_{ki}$) or with any state variable ($\sum_{k=1}^{n} P_{ki}$) is equal to 1 (37).

A Power System Example

Let us take a power system example in DAE form, using a comprehensive numerical method (15) to calculate the following important small-signal stability characteristic points:

- load flow feasibility points, beyond where there exists no solution for the system load flow equations;
- aperiodic and oscillatory stability points;
- min/max damping points.

The method employs the following constrained optimization problem:

$$a^2 \Rightarrow \min/\max \tag{79}$$

subject to

$$f(x, p_0 + \tau\Delta p) = 0 \tag{80}$$

$$As(x, p_0 + \tau\Delta p)l' - al' + \omega l'' = 0 \tag{81}$$

$$As(x, p_0 + \tau\Delta p)l'' - al'' - \omega l' = 0 \tag{82}$$

$$l_l' - 1 = 0 \tag{83}$$

$$l_i'' = 0 \tag{84}$$

where a is the real part of system eigenvalue of interest, ω is the imaginary part; l' and l'' are real and imaginary parts of the corresponding left eigenvector l; $l_i' + jl_i''$ is the ith element of the left eigenvector l; $p_0 + \tau\Delta p$ specifies a ray in the space of p; and A_s stands for the state matrix. In the preceding set,

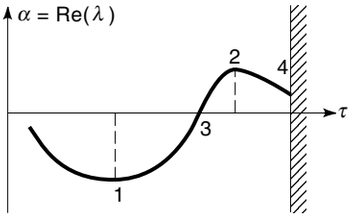


Figure 6. Different solutions of the problem: 1, 2—minimum and maximum damping; 3—saddle ($\omega = 0$) or Hopf ($\omega \neq 0$) bifurcations; 4—load flow feasibility boundary. $\alpha = \mathrm{Re}(\lambda)$: real part of system eigenvalue; τ = system parameter variation factor. These characteristic points can be located in one approach using a general method, as described in text.

(80) is the load flow equation and conditions (81)–(84) provide an eigenvalue with the real part of a and the corresponding left eigenvector.

The problem may have a number of solutions, and each one of them presents a different aspect of the small-signal stability problem as shown in Fig. 6.

The minimum and maximum damping points correspond to zero derivative $da/d\tau$. The constraint set (80)–(84) gives all unknown variables at these points. The minimum and maximum damping, determined for all oscillatory modes of interest, provides essential information about damping variations caused by a directed change of power system parameters.

The saddle node or Hopf bifurcations correspond to $a = 0$. They indicate the small-signal stability limits along the specified loading trajectory $p_0 + \tau\Delta p$. Besides revealing the type of instability (aperiodic for $\omega = 0$ or oscillatory for $\omega \neq 0$), the constraint set (80)–(84) gives the frequency of the critical oscillatory mode. The left eigenvector $l = l' + jl''$ (together with the right eigenvector $r = r' + jr''$ which can be easily computed in turn) determines such essential factors as sensitivity of a with respect to p, the mode, shape, participation factors, observability, and excitability of the critical oscillatory mode (29,37,38).

The load flow feasibility boundary points (80) reflect the maximal power transfer capabilities of the power system. Those conditions play a decisive role when the system is stable everywhere on the ray $p_0 + \tau\Delta p$ up to the load flow feasibility boundary. The optimization procedure stops at these points because the constraint (80) cannot be satisfied anymore.

The problem (79)–(84) takes into account only one eigenvalue each time. The procedure must be repeated for all eigenvalues of interest. The choice of eigenvalues depends upon the concrete task to be solved. The eigenvalue sensitivity, observability, excitability, and controllability factors (29,37,39) can help to determine the eigenvalues of interest and to trace them during optimization.

The result of optimization depends on the initial guesses for all variables in (79)–(84). To get all characteristic points for a selected eigenvalue, different initial points may be computed for different values of τ.

ACKNOWLEDGMENTS

The authors thank Professor Sergey M. Ustinov of Information and Control System Department, Saint-Petersburg State Technical University, Russia, for his substantial help in writing the article. Z. Y. Dong's work is supported by a Sydney University Electrical Engineering Postgraduate Scholarship.

BIBLIOGRAPHY

1. A. S. Deif, *Advanced Matrix Theory for Scientists and Engineers,* 2nd ed., New York: Abacus Press, New York: Gordon and Breach Science Publishers, 1991.
2. D. K. Faddeev and V .N. Faddeeva, *Computational Methods of Linear Algebra* (translated by R. C. Williams), San Francisco: Freeman, 1963.
3. F. R. Gantmacher, *The Theory of Matrices,* New York: Chelsea, 1959.
4. P. Lancaster, *Theory of Matrices,* New York: Academic Press, 1969.
5. J. H. Wilkinson, *The Algebraic Eigenvalue Problem,* New York: Oxford Univ. Press, 1965.
6. J. H. Wilkinson and C. H. Reinsch, *Handbook for Automatic Computation. Linear Algebra,* New York: Springer-Verlag, 1971.
7. J. H. Wilkinson, *Rounding Errors in Algebraic Problem,* Englewood Cliffs, NJ: Prentice-Hall, 1964.
8. G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems,* 3rd ed., Reading, MA: Addison-Wesley, 1994.
9. K. Ogata, *Modern Control Engineering,* 3rd ed., Upper Saddle River, NJ: Prentice-Hall International, 1997.
10. B. Porter and T. R. Crossley, *Modal Control, Theory and Applications,* London: Taylor and Francis, 1972.
11. L. A. Zadeh and C. A. Desoer, *Linear System Theory,* New York: McGraw-Hill, 1963.
12. S. Barnett and C. Storey, *Matrix Methods in Stability Theory,* London: Nelson, 1970.
13. V. Venkatasubramanian, H. Schattler, and J. Zaborszky, Dynamics of large constrained nonlinear systems—A taxonomy theory, *Proc. IEEE,* **83**: 1530–1561, 1995.
14. V. Ajjarapu and C. Christy, The continuation power flow: A tool for steady-state stability analysis, *IEEE Trans. Power Syst.,* **7**: 416–423, 1992.
15. Y. V. Makarov, V. A. Maslennikov, and D. J. Hill, Calculation of oscillatory stability margins in the space of power system controlled parameters, *Proc. Int. Symp. Electr. Power Eng., Stockholm Power Tech., Vol. Power Syst.,* Stockholm, 1995, pp. 416–422.
16. H. K. Khalil, *Nonlinear Systems,* 2nd ed., Upper Saddle River, NJ: Prentice-Hall, 1996.
17. R. Seydel, *From Equilibrium to Chaos, Practical Bifurcation and Stability Analysis,* 2nd ed., New York: Springer-Verlag, 1994.
18. A. J. Fossard and D. Normand-Cyrot, *Nonlinear Systems,* vol. 2, London: Chapman & Hall, 1996.
19. C.-W. Tan et al., Bifurcation, chaos and voltage collapse in power systems, *Proc. IEEE,* **83**: 1484–1496, 1995.
20. Y. V. Makarov, Z Y. Dong, and D. J. Hill, A general method for small signal stability analysis, *Proc. Int. Conf. Power Ind. Comput. Appl. (PICA '97),* Columbus, OH, 1997, pp. 280–286.
21. E. H. Abed, Control of bifurcations associated with voltage instability, *Proc. Bulk Power Syst. Voltage Phenom. III, Voltage Stab., Security, Control,* Davos, Switzerland, 1994, pp. 411–419.
22. H. G. Kwatny, R. F. Fischl, and C. O. Nwankpa, Local bifurcation in power systems: Theory, computation, and application (invited paper), *Proc. IEEE,* **83**: 1456–1483, 1995.
23. V. Ajjarapu and B. Lee, Bifurcation theory and its application to nonlinear dynamical phenomena in an electrical power system, *IEEE Trans. Power Syst.,* **7**: 424–431, February, 1992.

24. C. A. Canizares et al., Point of collapse methods applied to AC/DC power systems, *IEEE Trans. Power Syst.,* **7**: 673–683, May, 1992.

25. C. Canizares, A. Z. de Souza, and V. H. Quintana, Improving continuation methods for tracing bifurcation diagrams in power systems, *Proc. Bulk Power Syst. Voltage Phenom. III, Voltage Stab., Security Control,* Davos, Switzerland, 1994.

26. H.-D. Chiang et al., On voltage collapse in electric power systems, *IEEE Trans. Power Syst.,* **5**: 601–611, May, 1990.

27. H.-D. Chiang et al., CPFLOW: A practical tool for tracing power system steady state stationary behavior due to load and generation variations, *IEEE Trans. Power Syst.,* **10**: 623–634, 1995.

28. J. H. Chow and A. Gebreselassie, Dynamic voltage stability analysis of a single machine constant power load system, *Proc. 29th Conf. Decis. Control,* Honolulu, Hawaii, 1990, pp. 3057–3062.

29. I. A. Hiskens, Analysis tools for power systems—Contending with nonlinearities, *Proc. IEEE,* **83**: 1484–1496, 1995.

30. P. W. Sauer, B. C. Lesieutre, and M. A. Pai, Maximum loadability and voltage stability in power systems, *Int. J. Electr. Power Energy Syst.,* **15**: 145–154, 1993.

31. G. Strang, *Linear Algebra and its Applications,* New York: Academic Press, 1976.

32. G. W. Stewart, A bibliography tour of the large, sparse generalized eigenvalue problem, in J. R. Bunch and D. J. Rose (eds.), *Sparse Matrix Computations,* New York: Academic Press, 1976, pp. 113–130.

33. G. B. Price, A generalized circle diagram approach for global analysis of transmission system performance, *IEEE Trans. Power Apparatus Syst.,* **PAS-103**: 2881–2890, 1984.

34. G. Peters and J. Wilkinson, The least-squares problem and pseudo-inverses, *Comput. J.,* **13**: 309–316, 1970.

35. J. R. Bunch and D. J. Rose, *Sparse Matrix Computations,* New York: Academic Press, 1976.

36. R. J. Goult et al., *Computational Methods in Linear Algebra,* London: Stanley Thornes, 1974.

36a. H. Okamoto et al., Identification of equivalent linear power system models from electromagnetic transient time domain simulations using Prony's method, *Proc. 35th Conf. Decision and Control,* Kobe, Japan, December 1996, pp. 3875–3863.

36b. J. G. Proakis et al., *Advanced Digital Signal Processing,* New York: Macmillan, 1992.

37. P. Kundur, *Power System Stability and Control,* New York: McGraw-Hill, 1994.

38. I. A. Gruzdev, V. A. Maslennikov, and S. M. Ustinov, Development of methods and software for analysis of steady-state stability and damping of bulk power systems, in *Methods and Software for Power System Oscillatory Stability Computations,* St. Petersburg, Russia: Publishing House of the Federation of Power and Electro-Technical Societies, 1992, pp. 66–88 (in Russian).

39. D. J. Hill et al., Advanced small disturbance stability analysis techniques and MATLAB algorithms, A final report of the work: *Collaborative Research Project Advanced System Analysis Techniques,* New South Wales Electricity Transmission Authority and Dept. Electr. Eng., Univ. Sydney, 1996.

YURI V. MAKAROV
Howard University

ZHAO YANG DONG
University of Sydney

EKG. See ELECTROCARDIOGRAPHY.

Elliptic Equations, Parallel Over Successive Relaxation Algorithm

Standard Article

Gerard G. L. Meyer[1] and Michael V. Pascale[2]

[1]Johns Hopkins University, Baltimore, MD

[2]Northrop Grumman, Baltimore, MD

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2414

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (229K)

Abstract

The sections in this article are

Architecture and Architectural Parameters

The Parameterized Family of SOR Algorithms

Latency Analysis

Conclusions

| | |

Copyright © 1999-2008 All Rights Reserved.

arrays. In this article, the following two-dimensional, second-order linear partial differential equation (PDE)

$$a(x,y)\frac{\partial^2 u}{\partial x^2} + b(x,y)\frac{\partial u}{\partial x} + c(x,y)\frac{\partial^2 u}{\partial y^2} + d(x,y)\frac{\partial u}{\partial y} + e(x,y)\frac{\partial^2 u}{\partial x \partial y} + f(x,y)u = g(x,y) \quad (1)$$

and its numerical solution via Successive Over-Relaxation (SOR) methods is considered. Given an initial estimate $u^{(0)}$, the SOR methods (7) obtain a refined estimate $u^{(R)}$ of the solution of Eq. (1) discretized over an $M \times N$ grid by using R iterations which iteratively improve each of the discretized solution estimate components $u_{m,n}^{(r)}$ by combining the previous estimate $u_{m,n}^{(r-1)}$ with recent estimates of its northern, western, eastern, and southern neighbors. Thus

$$u_{m,n}^{(r)} = u_{m,n}^{(r-1)} - \omega^{(r)}[\beta_{m,n,N} u_{m-1,n}^{(*N)} + \beta_{m,n,W} u_{m,n-1}^{(*W)} + u_{m,n}^{(r-1)} + \beta_{m,n,E} u_{m,n+1}^{(*E)} + \beta_{m,n,S} u_{m+1,n}^{(*S)} - \gamma_{m,n}] \quad (2)$$

for $r = 1, 2, \ldots, R$ and for all $(m, n) \in \Omega^\circ$, given the relaxation sequence $\omega^{(r)}$ for $r = 1, 2, \ldots, R$, an initial discretized solution estimate $u_{m,n}^{(0)}$ for all $(m, n) \in \Omega^\circ$ and boundary conditions $u_{m,n}^{(R)} = u_{m,n}^{(0)}$ for all $(m, n) \in \partial\Omega$ where

$$\Omega = \left\{ (m,n) \middle| \begin{matrix} m \in \{0, 1, \ldots, M+1\} \\ n \in \{0, 1, \ldots, N+1\} \end{matrix} \right\}$$

where Ω° and $\partial\Omega$ denote the interior and boundary of Ω, respectively, and where each sweeping ordering parameter $*_N$, $*_W$, $*_E$, and $*_S$ takes a value of r or $(r - 1)$ and implies a sequence of precedence among the computations of $u_{m,n}^{(r)}$. A family of parallel SOR algorithms is obtained by segmenting the SOR algorithms into arithmetic grains, parameterizing the assignment of the arithmetic grains to at most P parallel processes intended for execution on P processors, and parameterizing the number of arithmetic grains computed between communications events. To evaluate the complexity and performance of the parallel algorithms presented here, it is assumed that $\omega^{(r)}$ and R are known and that the discretization grid is static.

Because the numerical performance and parallelism of a given algorithm depend on the ordering parameters (8–11), the Jacobi (J), red–black Gauss–Seidel (RB), and natural Gauss–Seidel (GS) orderings are considered. In the Jacobi ordering, $*_N = *_W = *_E = *_S = r - 1$. Thus with the J ordering, all components at iteration r may be computed in parallel. In

ELLIPTIC EQUATIONS, PARALLEL OVER SUCCESSIVE RELAXATION ALGORITHM

Numerous numerical parallel techniques exist for solving elliptic partial differential equation discretizations (1–4). The most popular among these are parallel Successive Over-Relaxation (SOR) (5) and parallel multigrid methods (6) for a variety of parallel architectures, including shared memory machines, vector processors, and one- and two-dimensional

Table 1. Ordering Parameters

Sweeping Order		$*N$	$*W$	$*E$	$*S$
Jacobi (J)		$r-1$	$r-1$	$r-1$	$r-1$
Red–black (RB)	red	$r-1$	$r-1$	$r-1$	$r-1$
	black	r	r	r	r
Gauss–Seidel (GS)		r	r	$r-1$	$r-1$

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

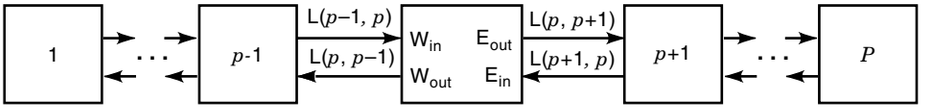


Figure 1. Linear array with bidirectional communication links.

the RB ordering, the components of u are divided into two groups; $u_{m,n}$ is red if $(m + n)$ is even, and black if $(m + n)$ is odd. Red components at iteration r are updated using black components from iteration $(r - 1)$, that is, $*_N = *_W = *_E = *_S = r - 1$. Black components at iteration r are updated using red components from iteration r, that is, $*_N = *_W = *_E = *_S = r$. Thus with the RB ordering, all red components may be computed in parallel followed by the computation of all black components in parallel. In the GS ordering, $*_N = *_W = r$, and $*_E = *_S = r - 1$, and thus all components with identical values of $(m + n)$ may be computed in parallel. These orderings are summarized in Table 1.

If the number of iterations R, which guarantee a solution of desired accuracy is not known, then a dynamic stop rule can be implemented by redefining an arithmetic grain to include accumulating the magnitudes of the terms in the parenthesis of Eq. (2) for each iteration and comparing the accumulation to a threshold.

The number of iterations R which guarantee a solution of desired accuracy depend on the relaxation sequence $\omega^{(r)}$. There are many relaxation schemes including static (7), unadaptive dynamic (7,12), global adaptive dynamic (13), and local adaptive dynamic (14–16). In the static and unadaptive dynamic cases, the relaxation sequence $\omega^{(r)}$ is known before execution of the SOR and therefore the evaluation of the SOR requires no computations other than those in Eq. (2). In the global adaptive dynamic and local adaptive dynamic cases, the relaxation sequence is computed as the SOR iterations proceed. In these cases, again an arithmetic grain can be redefined to incorporate the computations of such adaptive strategies.

The use of an adaptive grid is another strategy that can enhance SOR algorithm performance (17). This strategy computes an initial, crude, approximate solution on a coarse mesh with a low-order numerical method that is enriched until a prescribed accuracy is attained. Enrichment indicators, which are frequently estimates of the local discretization error, are used to control the adaptive process. Resources are introduced in regions having large enrichment indicators and are deleted from regions where indicators are low. This strategy can also be incorporated by redefining an arithmetic grain to include the calculation and usage of enrichment indicators.

To evaluate the complexity and performance of the parallel algorithms presented in this article, it is assumed that the relaxation sequence is known, that R is fixed and known, and that the discretization grid is static.

ARCHITECTURE AND ARCHITECTURAL PARAMETERS

The target architecture and associated software protocol consists of P processors connected in a linear array with bidirectional communication links, as shown in Fig. 1. The linear array was chosen for several reasons. First, it is among the least complex of all parallel architectures. If a parallel algorithm can be devised to execute efficiently on a linear array, then it is not necessary to consider more complicated architectures. Second, an algorithm developed for a linear-array topology is portable among architectures because it can be executed on topologies which include the linear array. Third, linear arrays require less hardware, are physically smaller, consume less power, require less cabling and backplane wiring, and are less expensive than more heavily connected topologies.

There are two communication links between processor p and processor $(p - 1)$ designated W_{in} (West in) and W_{out} (West out) on processor p. Likewise, there are two communication links between processor p and processor $(p + 1)$ designated E_{in} (East in) and E_{out} (East out) on processor p. The unidirectional link from processor p to processor q is designated $L(p, q)$. Each processor executes an instruction stream consisting of arithmetic and message initiation instructions. Input and output message initiations must be paired for two processors to communicate and exchange data. Communication between processors is synchronized. When data is passed between two processors, the output processor is blocked until the input processor is ready and vice versa (18). Furthermore, output messages are not initiated until the last word of a message has been computed.

Total latency is a combination of arithmetic latency and communication latency. Each processor requires time τ_a to execute an arithmetic instruction where τ_a includes the cost

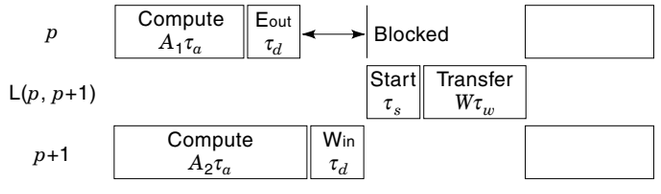


Figure 2. Nonconcurrent message startup blockage from p to $(p + 1)$.

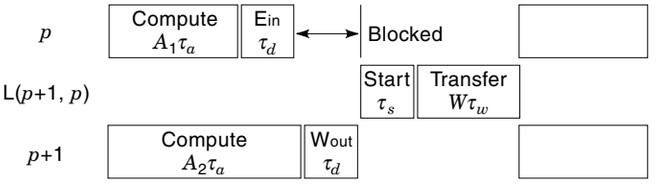


Figure 3. Nonconcurrent message startup blockage from $(p + 1)$ to p.

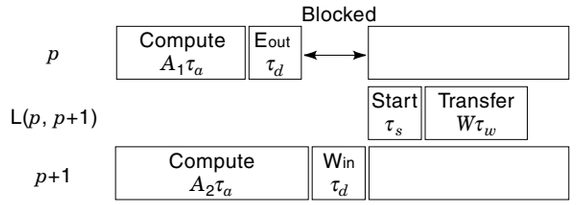


Figure 4. Concurrent message startup blockage.

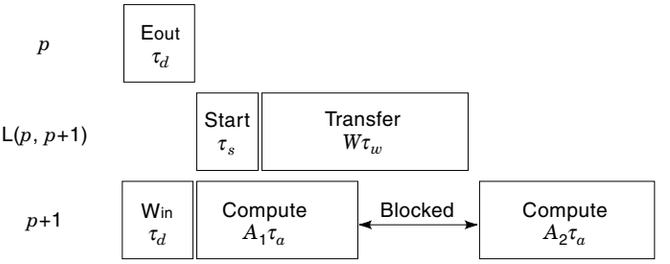


Figure 6. Data-dependency blockage.

of instruction fetch and decode, operand fetch and save, caching, operand index calculation, loop overhead, etc.. Each processor requires overhead time τ_d to initiate a message, where τ_d includes the cost of initializing source address, destination address, and message length registers, possible buffer allocation, etc. A communication link requires time $\tau_c(W) = \tau_s + W\tau_\omega$ to transfer a W-word message across a link where τ_s is the message start-up time and τ_ω is the per word transfer time if the other processor participating in the communication is ready for the message transfer. If the other processor is not ready, then the link blocks and transfer of the message is delayed. Message startup time τ_s includes the time to synchronize clocks, transfer header information, etc.

The capabilities of the P processors classify the architecture as either *nonconcurrent* or *concurrent.* The presence or absence of concurrency is usually determined by the presence or absence of a direct memory access (DMA) unit.

Nonconcurrent Architecture

In nonconcurrent architecture, the P processors perform either the execution of arithmetic instructions, message initiation instructions, or unidirectional communications across one communication link at any given time. When a (synchronized) communication takes place from processor p to processor $(p + 1)$ across a communication link, the processor which finishes its corresponding message initiation first, say p, blocks as seen in Fig. 2. When the other processor $(p + 1)$ finishes its message initiation, the link unblocks and message startup occurs on the communication link for a duration τ_s. At the conclusion of startup, words are transferred across the communication link with a latency of τ_ω for each word until the message transfer is complete. Arithmetic and message-initiation processing remains blocked throughout the message transfer. At the conclusion of the message transfer, arithmetic or message-initiation processing resumes on both processors (dashed boxes). Figure 3 shows a similar situation with the direction of communication reversed, that is, data is transferred from $(p + 1)$ to p. In this case, it is still the processor that finishes message initiation first (p) which blocks.

Concurrent Architecture

In concurrent architecture, the processors are capable of executing arithmetic instructions or message-initiation instructions simultaneously with bidirectional communications on all communication links. A processor that finishes message initiation first blocks, as in the nonconcurrent case. However, after the second processor finishes message initiation, execution of arithmetic instructions or message initiation may resume, as shown in Fig. 4 in addition to the unblocking of the communication link.

Initiation of a message on a communication link is blocked until any message in progress on that link completes. For example, message initiations from processor p to processor $(p + 1)$ are blocked on both p and $(p + 1)$ until the transfer from p to $(p + 1)$ is complete, as shown in Fig. 5.

If arithmetic instructions depend on message data, processing is blocked until message completion. For example in Fig. 6, processor $(p + 1)$ executes A_1 arithmetic instructions, blocks until the message transfer is complete, and then executes A_2 arithmetic instructions which are assumed to depend on message data.

Note that if instructions are properly coordinated among processors, then it is possible for a processor to execute arithmetic instructions simultaneously with the transfer of messages on all communications links, as shown in Fig. 7.

THE PARAMETERIZED FAMILY OF SOR ALGORITHMS

An *arithmetic grain,* denoted by its output $u^{(r)}_{m,n}$, consists of the operations of Eq. (2) for fixed m, n, and r. The arithmetic complexity and communication among grains are summarized in Table 2. Thus R iterations of SOR consist of MNR arithmetic grains whose execution require $11MNR$ operations.

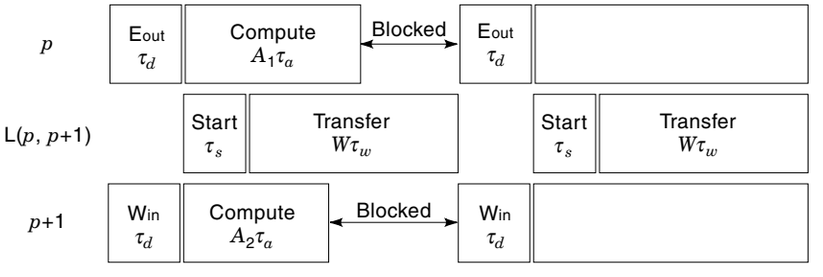


Figure 5. Message-initiation blockage.

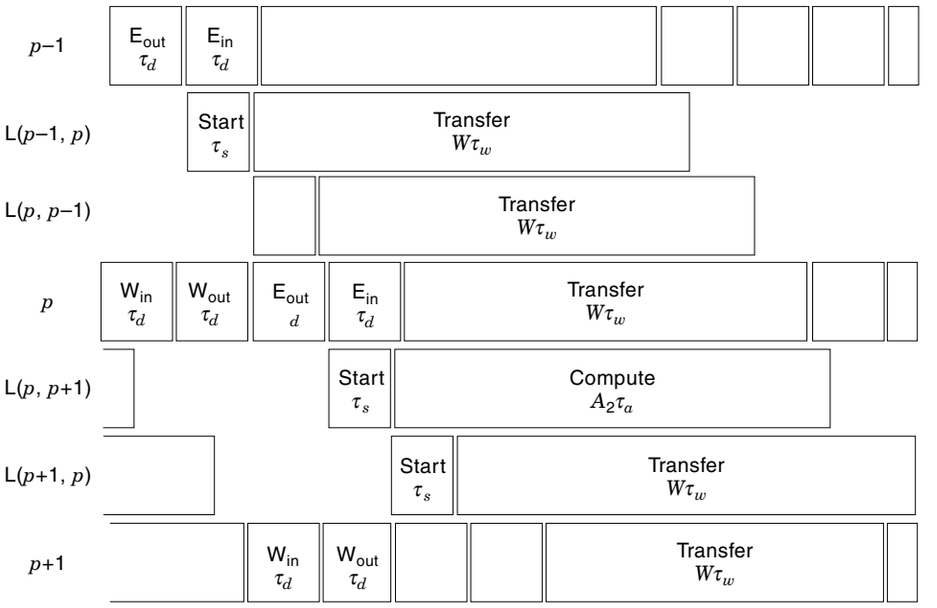


Figure 7. Maximum arithmetic and communications concurrency.

The assignment of the MNR grains to the P processes is dictated by P *arithmetic grain aggregation coefficients* $h_1, h_2, \ldots, h_P$, where

$$\left\lfloor \frac{N}{P} \right\rfloor \le h_p \le \left\lceil \frac{N}{P} \right\rceil, p = 1, 2, \ldots, P, \text{ and } \sum_{p=1}^{P} h_p = N$$

Then the arithmetic grain aggregation coefficients are used to define the *cumulative arithmetic grain aggregation coefficients* $H_0, H_1, \ldots, H_P$, where

$$H_0 = 0, H_p = H_{p-1} + h_p, p = 1, 2, \ldots, P$$

The arithmetic grains assigned to process p for $p = 1, 2, \ldots, P$ are $u^{(r)}_{m,n}$ for all $m = 1, 2, \ldots, M$, for all $n = H_{p-1} + 1, H_{p-1} + 2, \ldots, H_p$, and for all $r = 1, 2, \ldots, R$. The relationship between the discretization grid and the processing array is shown in Fig. 8.

Because the number of arithmetic grains assigned to process p is h_pMR, the number of arithmetic operations executed by process p is $11h_pMR$. For each $r = 1, 2, \ldots, R$, each process p depends on receiving a western boundary of M words consisting of $u^{(*W)}_{m,H_p-1}$ for $m = 1, 2, \ldots, M$ and an eastern boundary of M words consisting of $u^{(*E)}_{m,H_p+1}$ for $m = 1, 2, \ldots, M$. In addition, for each $r = 1, 2, \ldots, R$, each process p must send a western boundary of M words consisting of $u^{(r)}_{m,H_{p-1}+1}$ for $m = 1, 2, \ldots, M$ and an eastern boundary of M words consisting of $u^{(r)}_{m,H_p}$ for $m = 1, 2, \ldots, M$. The total arithmetic and communication complexities for process p are summarized in Table 3.

The order in which arithmetic grains are executed by each process must take into account their interprocess dependencies. Because the GS sweeping dependencies are supersets of the RB dependencies, which in turn are supersets of the J sweeping dependencies, the arithmetic grain ordering for GS sweeping is chosen. For RB sweeping, the indices are relabeled so that all red arithmetic grains precede black arithmetic grains. The execution of arithmetic grain $u^{(r)}_{m,n}$ depends on the input variables given in Table 2. To satisfy these dependencies, the arithmetic grains are executed from top to bottom among rows and from left to right within a row (see Fig. 9).

A *communication grain* is the communication of a single word of boundary information by any process p to the western process $(p - 1)$ or to the eastern process $(p - 1)$. There is an input and output communication grain associated with each arithmetic grain on the left and right edges of Fig. 9. The order in which the communication grains are executed is chosen as the order in which the corresponding boundary information is needed and generated by each process according to the arithmetic grain execution ordering described before. Communication grains between process p and western process $(p - 1)$ are executed from top to bottom, and communication grains between process p and eastern process $(p + 1)$ are also executed from top to bottom.

Let an *arithmetic step* be the contiguous arithmetic grains executed between communications events, and let U be the number of communication grains in any message. The choice of U induces the number of arithmetic grains in each arithmetic step. Because the time to communicate a W-word message is $\tau_c(W) = \tau_s + W\tau_w$, longer messages, that is, large U, result

Table 2. Arithmetic Grain Computation and Communication Complexities

Operations			Input		Output	
+	×	Total	Variables	Words	Variables	Words
6	5	11	$u^{(*N)}_{m-1,n}, u^{(*W)}_{m,n-1}, u^{(r-1)}_{m,n}, u^{(*E)}_{m,n+1}, u^{(*S)}_{m+1,n}$	5	$u^{(r)}_{m,n}$	1

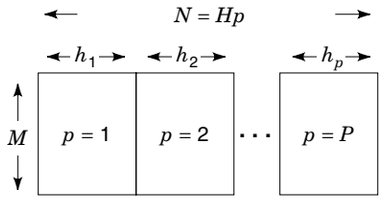


Figure 8. Discretization grid and processing-array relationship.

in a smaller average per word transfer time that reduces overall latency. However, longer messages also contribute to delaying computations on processors that depend on message data; thereby increasing overall latency. Thus expressing latency as a function U affords a means to determine this tradeoff optimally.

The number of input and output words to each process is MR, and therefore, the number of messages to and from each process is given by

$$S = \left\lceil \frac{MR}{U} \right\rceil$$

The reception of each U-word message by process p from process $(p - 1)$ with the corresponding message from process $(p + 1)$ enables computing Uh_p arithmetic grains which generate a pair of U word messages, one needed by process $(p - 1)$ and the other needed by process $(p + 1)$. Thus the execution of Uh_p arithmetic grains is bracketed by communications events which define an arithmetic step and therefore the number of arithmetic steps is S.

Five subroutines common to both concurrent and nonconcurrent parallel implementations of the SOR algorithm are now defined. Each subroutine call of `Comp(s)`, `Wout(s)`, `Eout(s)`, `Win(s)`, and `Ein(s)`, executes approximately $1/S$ of the total of arithmetic grains, western output grains, eastern output grains, western input grains, or eastern input grains, respectively where the argument $s \in [1, 2, \ldots, S]$ specifies which $1/S$ of the total grains is executed for a particular subroutine call. For instance, when $u_{i,j} = u^{(r)}_{m,n}$ with $i = M(r - 1) + m$ and $j = n$, then `Comp(s)` executes $u_{i,j}$ for $i = U(s - 1) + 1, U(s - 1) + 2, \ldots, Us$ and $j = H_{p-1} + 1, H_{p-1} + 2, \ldots, H_p$.

When the algorithm is executed, each process p has computed S arithmetic steps for $s = 1, 2, \ldots, S$, totaling g_pMR grains, sent S messages for $s = 1, 2, \ldots, S$ to process $(p - 1)$, totaling MR words, received S messages for $s = 1, 2, \ldots, S$ messages from process $p - 1$, totaling MR words, sent S messages for $s = 1, 2, \ldots, S$ to process $(p + 1)$, totaling MR words and received S messages for $s = 1, 2, \ldots, S$ from process $(p + 1)$ totaling MR words, excluding the boundary processes $p = 1$ and $p = P$, where the communication to process $(p - 1)$ and $(p + 1)$, respectively, is null.

Table 3. Grain Computation and Communication Complexities for Process p

Arithmetic		Input to Process p, Words		Output from Process p, Words	
Grains	Operations	$p - 1$	$p + 1$	$p - 1$	$p + 1$
h_pMR	$11h_pMR$	MR	MR	MR	MR

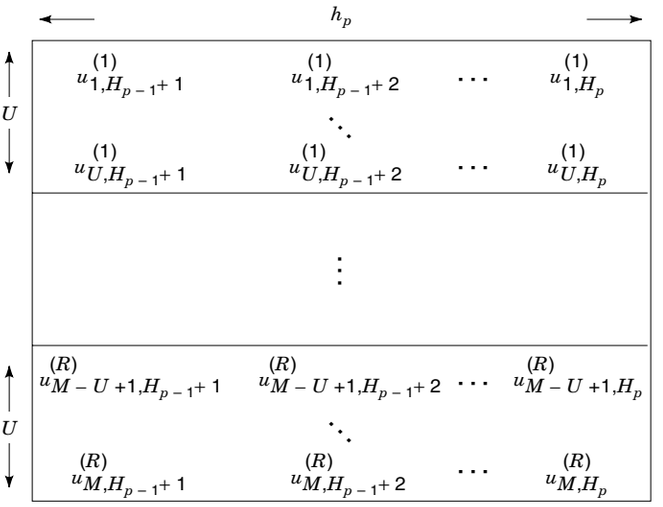


Figure 9. Arithmetic steps for process p.

LATENCY ANALYSIS

In this section, upper bounds on the overall latencies of the parameterized SOR algorithms are quantified for execution on a linear array of processors. In each case, the bounds are computed by evaluating the latency of the process q that has the maximum number of arithmetic grains associated with it, that is, $h_q = \lceil N/P \rceil$, and adding the latency of those processes or portions of processes required before and after execution process q. For convenience, the execution time of an arithmetic grain is denoted τ_g, and thus $\tau_g = 11\tau_a$.

Nonconcurrent SOR

When arithmetic computations and communications cannot be done simultaneously, the algorithm described in Fig. 10 is used for the J and RB sweepings, and the algorithm described in Fig. 11 is used for the GS sweeping, where the parallel execution of instructions $1, 2, \ldots, n$ is indicated by

instruction 1//instruction 2//. . .//instruction n

To satisfy dependency constraints, it is required that $U \leq M$ in the J case, and $U \leq M/2$ in the RB and GS cases.

In the J and RB cases, dependencies allow executing the worst-case process to begin immediately, and then execution proceeds without blocking because its western and eastern neighbors have at most the same number of grains to compute at each arithmetic step. Thus the latency in the J and RB cases is bounded from above by L_{Jn} and L_{RBn} with

$$\begin{aligned} L_{Jn} = L_{RBn} &= S(4(\tau_d + \tau_s + U\tau_w) + h_q U\tau_g) \\ &= \left\lceil \frac{MR}{U} \right\rceil (4\tau_d + 4\tau_s + 4U\tau_w + \lceil N/P \rceil U\tau_g) \end{aligned}$$

```
DO IN PARALLEL FOR p = 1, . . ., P
  for s = 1, 2, . . ., S
    Wout(s)
    Eout(s)
    Win(s)
    Ein(s)
    Comp(s)
  end
END
p = EVEN
```

```
DO IN PARALLEL FOR p = 1, . . ., P
  for s = 1, 2, . . ., S
    Ein(s)
    Win(s)
    Eout(s)
    Wout(s)
    Comp(s)
  end
END
  p = ODD
```

Figure 10. Nonconcurrent 1-D Jacobi/red–black algorithm.

Minimizing over the communication granularity parameter U in the J case gives $U = M$ and latency bound

$$L_{\mathrm{Jn}} = 4R\tau_{\mathrm{d}} + 4R\tau_{\mathrm{s}} + 4MR\tau_{\mathrm{w}} + MR\left\lceil \frac{N}{P} \right\rceil \tau_{\mathrm{g}}$$

and in the RB case gives $U = M/2$ and latency bound

$$L_{RBn} = 8R\tau_{\mathrm{d}} + 8R\tau_{\mathrm{s}} + 4MR\tau_{\mathrm{w}} + MR\left\lceil \frac{N}{P} \right\rceil \tau_{\mathrm{g}}$$

In the nonconcurrent GS case, dependencies block the execution of the worst case process q until processes $p = 1, 2, \ldots, q-1$ have executed their respective first triplet of input communications, arithmetic step, and output communications. Then execution of the worst case process proceeds unblocked because its western and eastern neighbors have at most the same number of grains to compute at each arithmetic step. When the worst case process concludes, processes $p = q+1, q+2, \ldots, P$ must execute their final triplet of input communications, arithmetic step, and output communications. The latency incurred before the process q loop can begin executing is expressed by

$$[1 + 2(q-2)](\tau_{\mathrm{d}} + \tau_{\mathrm{s}} + U\tau_{\mathrm{w}}) + \sum_{p=1}^{q-1} h_{\mathrm{p}} U \tau_{\mathrm{g}}$$

the latency incurred executing the process q loop is given by

$$4(S-1)(\tau_{\mathrm{d}} + \tau_{\mathrm{s}} + U\tau_{\mathrm{w}}) + (S-1)h_{\mathrm{q}} U \tau_{\mathrm{g}}$$

```
DO IN PARALLEL FOR p = 1, . . ., P
    Wout(1)
    Ein(1)
    for s = 1, 2, . . ., S − 1
      Win(s)
      Winout(s + 1)
      Comp(s)
      Eout(s)
      Ein(s + 1)
    end
    Win(S)
    Comp(S)
    Eout(S)
END
```

Figure 11. Nonconcurrent 1-D Gauss–Seidel algorithm.

and the latency incurred following the process q loop is given by

$$2(P-q)(\tau_{\mathrm{d}} + \tau_{\mathrm{s}} + U\tau_{\mathrm{w}}) + \sum_{p=q}^{P} h_{\mathrm{p}} U \tau_{\mathrm{g}}$$

Thus the latency in the GS case is bounded from above by L_{GSn} with

$$L_{\mathrm{GSn}} = \left(4\left\lceil \frac{MR}{U} \right\rceil + 2P - 7\right)(\tau_{\mathrm{d}} + \tau_{\mathrm{s}} + U\tau_{\mathrm{w}}) + \left[\left(\left\lceil \frac{MR}{U} \right\rceil - 1\right)\left\lceil \frac{N}{P} \right\rceil U + NU\right]\tau_{\mathrm{g}}$$

Given an SOR algorithm, architectural parameters P, τ_{a}, τ_{d}, τ_{w}, and τ_{s}, and problem parameters M, N, and R, the corresponding latency bound can be plotted as a function of the communication granularity U, and the optimal U may be obtained from the plot. For example, in the nonconcurrent GS case with architectural parameters $P = 8$, $\tau_{\mathrm{a}} = 1.34$ μs, $\tau_{\mathrm{d}} = 120.0$ μs, $\tau_{\mathrm{w}} = 9.0$ μs, and $\tau_{\mathrm{s}} = 12.2$ μs, and problem parameters $M = N = 90$ and $R = 10$, L_{GSn} is plotted as a function of U in Fig. 12. One sees that $U = 1$ yields 669.0 ms for the latency bound and that $U = 20$ yields 240.7 ms, the minimum

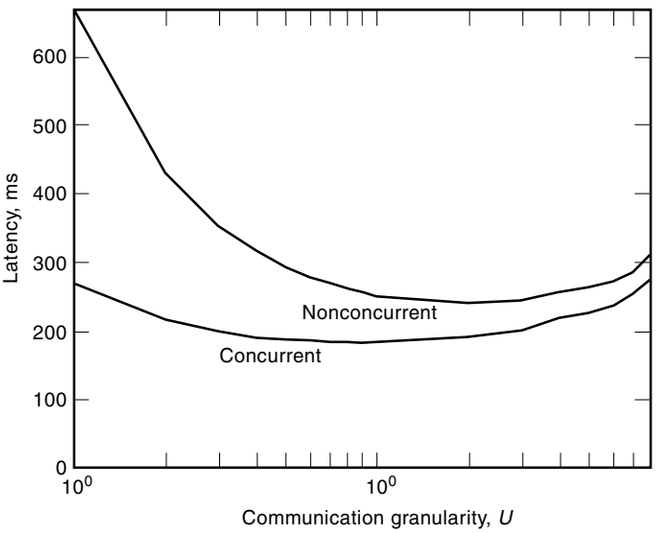


Figure 12. Gauss–Seidel latency vs. communication granularity for $P = 8$, $\tau_{\mathrm{a}} = 1.34$ μs, $\tau_{\mathrm{d}} = 120.0$ μs, $\tau_{\mathrm{w}} = 9.0$ μs, $\tau_{\mathrm{s}} = 12.2$ μs, $M = N = 90$, $R = 10$.

```
DO IN PARALLEL FOR p = 1, . . ., P
  Wout(1) // Eout(1) // Win(1) // Ein(1)
  for s = 1, 2, . . ., S − 1
    Wout(s + 1) // Eout(s + 1) // Win(s + 1) // Ein(s + 1) // Comp(s)
  end
  Comp(S)
END
```

Figure 13. Concurrent 1-D Jacobi/red-black algorithm.

latency bound. This may be compared to the single processor latency of 1193.8 ms, and we may conclude that the use of the customary U produces an efficiency of 22%, and the use of the optimal U produces an efficiency of 62%.

Concurrent SOR

When arithmetic computations and communications can be done simultaneously, the algorithm given in Fig. 13 is used for the J and RB cases, the algorithm given in Fig. 14 is used for the GS case. To satisfy dependency constraints and to permit the concurrent execution of arithmetic grains and communication grains, it is required that $U \le M/2$ in the J case and $U \le M/4$ in the RB and GS cases.

As in the nonconcurrent situation, dependencies in both the J and RB cases, allow execution of the worst case process to begin immediately and to proceed without blocking because its western and eastern neighbors have at most the same number of grains to compute at each arithmetic step. Thus the latency in the J and RB cases is bounded from above by L_{Jc} and L_{RBc} with

$$\begin{aligned} L_{\mathrm{Jc}} = L_{\mathrm{RBc}} &= (\tau_{\mathrm{d}} + \tau_{\mathrm{s}} + U\tau_{\mathrm{w}}) \\ &\quad + (S-1)\max\{\tau_{\mathrm{s}} + U\tau_{\mathrm{w}}, h_q U\tau_{\mathrm{g}} + \tau_{\mathrm{d}}\} + h_{\mathrm{q}} U\tau_{\mathrm{g}} \\ &= (\tau_{\mathrm{d}} + \tau_{\mathrm{s}} + U\tau_{\mathrm{w}}) + \left(\left\lceil \frac{MR}{U} \right\rceil - 1\right) \\ &\quad \max\left\{\tau_{\mathrm{s}} + U\tau_{\mathrm{w}}, \left\lceil \frac{N}{P} \right\rceil U\tau_{\mathrm{g}} + \tau_{\mathrm{d}}\right\} + \left\lceil \frac{N}{P} \right\rceil U\tau_{\mathrm{g}} \end{aligned}$$

If $\tau_{\mathrm{s}} + U\tau_{\mathrm{w}} \le \lceil N/P \rceil (U\tau_{\mathrm{g}} + \tau_{\mathrm{d}})$, then

$$L_{\mathrm{Jc}} = L_{\mathrm{RBc}} = (\tau_{\mathrm{s}} + U\tau_{\mathrm{w}}) + \left\lceil \frac{MR}{U} \right\rceil \left(U\left\lceil \frac{N}{P} \right\rceil \tau_{\mathrm{g}} + \tau_{\mathrm{d}}\right)$$

In the concurrent GS case, dependencies block the execution of the worst case process q until processes $p = 1, 2, \ldots, q-1$ have executed their respective first triplet of input communications, arithmetic step, and output communications. Then execution of the worst case process proceeds unblocked because its western and eastern neighbors have at most the same number of grains to compute at each arithmetic step. When the worst case process concludes, processes $p = q+1, q+2, \ldots, P$ must execute their final triplet of input communications, arithmetic step, and output communications. The latency incurred before process q can begin executing is expressed by

$$\sum_{p=1}^{q-1} (\tau_{\mathrm{s}} + U\tau_{\mathrm{w}} + \max\{\tau_{\mathrm{s}} + U\tau_{\mathrm{w}}, h_{\mathrm{p}} U\tau_{\mathrm{g}} + \tau_{\mathrm{d}}\})$$

the latency incurred executing process q is expressed by

$$(\tau_{\mathrm{s}} + U\tau_{\mathrm{w}}) + S\max\{\tau_{\mathrm{s}} + U\tau_{\mathrm{w}}, h_{\mathrm{g}} U\tau_{\mathrm{g}} + \tau_{\mathrm{d}}\}$$

and the latency incurred following process q is expressed by

$$\sum_{p=q+1}^{P} (\tau_{\mathrm{s}} + U\tau_{\mathrm{w}} + \max\{\tau_{\mathrm{s}} + U\tau_{\mathrm{w}}, h_{\mathrm{p}} U\tau_{\mathrm{g}} + \tau_{\mathrm{d}}\})$$

Thus the latency in the GS case is bounded from above by L_{GSc} where

$$\begin{aligned} L_{\mathrm{GSc}} &= \sum_{p=1}^{P} (\tau_{\mathrm{s}} + U\tau_{\mathrm{w}} + \max\{\tau_{\mathrm{s}} + U\tau_{\mathrm{w}}, h_{\mathrm{p}} U\tau_{\mathrm{g}} + \tau_{\mathrm{d}}\}) \\ &\quad + (S-1)\max\{\tau_{\mathrm{s}} + U\tau_{\mathrm{w}}, \left\lceil \frac{N}{P} \right\rceil U\tau_{\mathrm{g}} + \tau_{\mathrm{d}}\} \end{aligned}$$

If $\tau_{\mathrm{s}} + \mathrm{U}\tau_{\mathrm{w}} \le h_{\mathrm{p}}\, U\tau_{\mathrm{g}}$ for all p, then

$$\begin{aligned} L_{\mathrm{GSc}} &= P(\tau_{\mathrm{d}} + \tau_{\mathrm{s}} + U\tau_{\mathrm{w}}) + \left(\left\lceil \frac{MR}{U} \right\rceil - 1\right) \left(\left\lceil \frac{N}{P} \right\rceil U\tau_{\mathrm{g}} + \tau_{\mathrm{d}}\right) \\ &\quad + NU\tau_{\mathrm{g}} \end{aligned}$$

```
DO IN PARALLEL FOR p = 1, . . ., P
  Wout(1)
  Wout(2) // Ein(1)
  Wout(3) // Win(1)
  Wout(4) // Win(2) // Ein(2) // Comp(1)
  for s = 2, 3, . . ., S − 3
    Wout(s + 3) // Win(s + 1) // Ein(s + 1) // Comp(s) // Eout(s − 1)
  end
  Win(S − 1) // Ein(S − 1) // Comp(S − 2) // Eout(S − 3)
  Win(S) // Ein(S) // Comp(S − 1) // Eout(S − 2)
  Comp(S) // Eout(S − 1)
  Eout(S)
END
```

Figure 14. Concurrent 1-D Gauss–Seidel algorithm.

Once again, given architectural parameters P, τ_a, τ_d, τ_w, and τ_s, and problem parameters M, N, and R, the corresponding latency bound can be plotted as a function of the communication granularity U, and the optimal U may be obtained from the plot. For example, in the concurrent GS case with architectural parameters $P = 8$, $\tau_a = 1.34$ μs, $\tau_d = 120.0$ μs, $\tau_w = 9.0$ μs, and $\tau_s = 12.2$ μs, and problem parameters $M = N = 90$ and $R = 10$, L_{GSc} is plotted as a function of U in Fig. 12. One sees that $U = 1$ yields 269.0 ms for the latency bound and that $U = 9$ yields 183.5 ms, the minimum latency bound. This may be compared to the single processor latency of 1193.8 ms, and we may conclude that the use of the customary U produces an efficiency of 55% and the use of the optimal U produces an efficiency of 81%.

CONCLUSIONS

Whenever a W-word message has to be transferred from one processor to another, one incurs a computational cost τ_d to initiate and synchronize the message transfer and also a communication link cost $\tau_c(W) = \tau_s + W\tau_w$. It follows that longer messages result in a smaller average per word computational overhead and a smaller average per word communication transfer time that reduces overall latency. However, longer messages delay computations on processors which depend on message data, increasing latency. Using parameterized algorithms in which message size can be adjusted allows balancing message overhead against delays due to computational dependencies. Then, expressing latency as a function of *communication granularity,* which is related to message length, allows the optimally determining the necessary tradeoff. Parameterizing algorithms also has the advantage that high efficiencies are more easily maintained when hosted on a multiplicity of architectures because parameters may be adjusted for each architecture.

In this article, it has been shown that the relationship between latency and communication granularity U for a family of parametrized parallel SOR algorithms is pronounced and that the reduction in latency with an optimal choice of U is significant. The efficiencies of these algorithms are high whenever the corresponding optimal communication granularity is used, suggesting that architectures which are more complicated than the linear array need not be considered. Given a problem, one can determine or estimate the number of iterations R_J, R_{RB}, and R_{GS} required to achieve a desired accuracy for each sweeping order J, RB, and GS, find the optimal U and the corresponding latency for each case, and then choose the best SOR algorithm. The GS sweeping order is of the most interest, however, because it has a generally superior rate of convergence and because of its amenability to the enhancements mentioned in the introduction.

BIBLIOGRAPHY

1. R. E. Boisvert, Algorithms for special tridiagonal systems, *SIAM J. Sci. Stat. Comput.,* **12**: 423–442, 1991.
2. C. J. Ribbens, L. T. Watson, and C. Desa, Toward parallel mathematical software for elliptic partial differential equations, *ACM Trans. Math. Softw.,* **19**: 457–473, 1993.
3. G. Rodrigue, *Parallel Processing for Scientific Computations,* Philadelphia: SIAM, 1989.
4. H. A. Van der Vorst, High performance preconditioning, *SIAM J. Sci. Stat. Comput.,* **10**: 1174–1185, 1989.
5. A. Asenov, D. Reid, and J. R. Barker, Speed-up of scalable iterative linear solvers implemented on an array of transputers, *Parallel Comput.,* **20**: 375–387, 1994.
6. N. H. Naik and J. Van Rosendale, The improved robustness of multigrid elliptic solvers based on multiple semicoarsened grids, *SIAM J. Numer. Anal.,* **30**: 215–229, 1993.
7. W. H. Press et al., *Numerical Recipes in C,* Cambridge Univ. Press, 1992, Chap. 19.
8. L. M. Adams and H. F. Jordan, Is SOR color-blind?, *J. Sci. Stat. Comput.,* **7**: 490–506, 1986.
9. C.-C. J. Kuo, T. F. Chan, and C. Tong, Two color Fourier analysis of iterative algorithms for elliptic problems with red/black ordering, *SIAM J. Sci. Stat. Comput.,* **11**: 767–794, 1990.
10. C.-C. J. Kuo and B. C. Levy, Discretization and solution of elliptic PDEs—a digital signal processing approach, *Proc. IEEE,* **12**: 1808–1842, 1990.
11. J. M. Ortega and R. G. Voigt, Solution of partial differential equations on vector and parallel computers, *SIAM Rev.,* **27**: 149–240, 1985.
12. R. S. Varga, *Matrix Iterative Analysis,* Englewood Cliffs, NJ: Prentice-Hall, 1962.
13. L. A. Hageman and D. M. Young, *Applied Iterative Methods,* New York: Academic Press, 1981.
14. E. F. Botta and A. E. P. Veldman, On local relaxation methods and their application to convection-diffusion equations, *J. Comput. Phys.,* **48**: 127–149, 1981.
15. L. W. Ehrlich, An ad hoc SOR method, *J. Comput. Phys.,* **44**: 31–45, 1981.
16. C.- C. J. Kuo, B. C. Levy, and B. R. Musicus, A local relaxation method for solving elliptic PDEs on mesh-connected arrays, *SIAM J. Sci. Stat. Comput.,* **8**: 550–573, 1987.
17. R. Biswas, J. E. Flaherty, and M. Benantar, Advances in adaptive parallel processing for field applications, *IEEE Trans. Magn.,* **27**: 3768–3773, 1991.
18. D. P. O'Leary and P. Whitman, Parallel QR factorization by Householder and modified Gram–Schmidt algorithms, *Parallel Comput.,* **16**: 99–112, 1990.
19. G. G. L. Meyer and M. Pascale, A family of Parallel QR Factorization Algorithms, *High Performance Comput. Symp. '95,* 1995, pp. 95–106.
20. M. A. Pirozzi, The fast numerical solution of mildly nonlinear elliptic boundary value problems on multiprocessors, *Parallel Comput.,* **19**: 1117–1128, 1993.

GERARD G. L. MEYER
Johns Hopkins University
MICHAEL V. PASCALE
Northrop Grumman

Equation Manipulation
Standard Article
Deepak Kapur[1]
[1]University of New Mexico
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2456
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (519K)

Abstract

The sections in this article are

Equational Inference

Equation Solving Over Terms: Unification

Polynomial Equation Solving

Acknowledgment

| | |

Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

EQUATION MANIPULATION

Formal symbol manipulation is ubiquitous. In a broad sense, most of computer science, artificial intelligence, symbolic logic, and even mathematics can be viewed as nothing but symbol manipulation. We focus on a very narrow but useful aspect of symbol manipulation: *reasoning about equations.* Equations arise in all aspects of modeling and computation in many fields of engineering and their applications. We show how equations can be deduced from other equations (using the properties of equality) as well as how equations can be solved, both in a general framework and in some concrete cases.

In the next section, we discuss a rewrite-rule-based approach for inferring equations from other equations. Equational hypotheses are transformed into unidirectional simplification rules, and these rules are then used for rewriting. The concept of a *completion procedure* is introduced for generating a *canonical* rewrite system from a given rewrite system. A canonical rewrite system has the useful property that every object has a *unique normal form* (*canonical form*) using the rewrite system. Objects equivalent by an equality relation have the same canonical form. To infer an equation, it is necessary and sufficient to check whether the two sides of the equation have identical canonical forms.

In the third section, equation solving is reviewed. Given a set of equations in which function symbols are assumed to be uninterpreted (i.e., no special meaning of the symbols is assumed), a method is given for finding substitutions for variables that make the two sides of every equation identical. This unification problem arises in many diverse subfields of computer science and artificial intelligence, including automated reasoning, expert systems, natural language processing, and programming. In the same section, we also discuss how the algorithm is changed to exploit the semantics of function symbols when possible. In particular, we discuss changes to the unification algorithm when some function symbols are commutative, or both associative and commutative, as is often the case for many applications.

The fourth section is the longest. It reviews three different approaches for solving polynomial equations over complex numbers—*resultants*, the *characteristic set* method, and the *Gröbner basis* method. For polynomial equations with parameters, these methods can be used to identify conditions on parameters under which a system of polynomial equations has a solution. This problem comes up in many application domains in engineering, including CAD/CAM, solid modeling, robot kinematics, computer vision, and chemical equilibria. These approaches are compared on a wide variety of examples.

Equational Inference

Consider the following equations:

$$x + 0 = x \tag{1}$$

$$x + -(x) = 0 \tag{2}$$

$$(x + y) + z = x + (y + z) \tag{3}$$

In the above equations, 0 is a constant symbol, standing for the identity for $+$, a binary function symbol, and $-$ is a unary function symbol. Each variable is universally quantified; that is, any expression involving variables, the operations $-$, $+$, 0, and other generators can be substituted for the free variables.

The reader may have noticed that these equations are the defining axioms of the familiar algebraic structure *groups*. It is easy to see that the equations

$$\begin{aligned} u &= u + 0 = u + (-(u) + -(-(u))) \\ &= (u + -(u)) + -(-(u)) = 0 + -(-(u)) \end{aligned}$$

follow from the above defining axioms. By appropriately substituting expressions for variables in the above axioms, this deduction follows by the properties of equality. However, proving other equations routinely given as homework problems in a first course on abstract algebra, such as

$$\begin{aligned} 0 + u &= u \\ -(u) + u &= 0 \\ -(u + v) &= -(v) + -(u) \end{aligned}$$

or showing that $-(u + v) \neq -(u) + -(v)$, is not that easy. It requires some effort. One has to find appropriate substitutions for variables in the above axioms, and then chain them properly to derive these properties.

In a more general setting, a natural question to ask is: given a finite set of equations as the defining axioms, such as the group axioms above, can another equation be deduced from them by substituting for universally quantified variables and using the well-known properties of equality, such as reflexivity, symmetry, transitivity, and replacement of equals by equals? This is a fundamental problem in equational deduction. The answer to this problem is, in the general case, negative, as the problem is undecidable (that is, there can never exist a computer program/algorithm that can solve this problem in general, even though specific instances of it can be solved).

In many cases, however, this question can be answered. Below we discuss a particular heuristic that is often useful in finding the answer. The basic idea is not to use the above equational axioms in both directions; in other words, if an instance of its left side can be replaced by the corresponding instance of its right side, then that instance of its right side cannot be replaced by the instance of its left side, and similarly, the other way around. Instead, using some uniform well-founded measure on expressions, we determine which of the two sides in an equation is more "complex," and view an equation as a unidirectional rewrite (simplification) rule that transforms more complex expressions into less complex. We often employ such a heuristic in solving problems. For instance, the axiom

$$x + 0 = x$$

is viewed as a simplification rule in which $x + 0$ (or any other instance of it) is simplified to x, and not the other way around; that is, x is never replaced by $x + 0$.

Let us precisely define *simplification* or *rewriting*. Given a rewrite rule

$$L \rightarrow R$$

where L and R are terms built from variables and function symbols, a term t can be simplified by the rule (at position p, a sequence of nonzero positive integers,[1] in t) if the subterm t/p of t *matches* L, that is there is a substitution σ for variables in L, written as $\{ x_1 \leftarrow s_1, x_2 \leftarrow s_2, \ldots, x_n \leftarrow s_n \}$, such that $t/p = \sigma(L)$, the result

of applying the substitution σ on L . The result of this simplification is then

$$t[p \leftarrow \sigma(R)]$$

—the term obtained by replacing the subterm at position p in t by $\sigma(R)$. This definition can be extended to consider rewriting (including multistep rewriting) by a system of rewrite rules. A term t is in *normal form* (also called *irreducible*) with respect to a rule $L \rightarrow R$ (respectively, a system of rewrite rules) if no subterm in t matches L (respectively, the left side of any rewrite rule in the system).

It should be noted that for rewriting to be meaningfully defined, the variables appearing in the right side R must also appear in the left side L, as otherwise substitutions for extra variables appearing in R cannot be determined. Henceforth, every rule is assumed to satisfy this property; i.e., all variables appearing in R must appear in L as well.

For example,

$$(u + -(u)) + -(-(u)) = u$$

can be proved easily by simplifying the left side: first using the axiom in Eq. (3), followed by the axiom in Eq. (2), and then by axiom in Eq. (1), using each axiom as a left-to-right rewrite rule.

For certain equations, it is not too difficult to determine which side is more complex (e.g., the first two axioms). But there are cases in which it is not easy.[2]

Consider the case of the associativity axiom. The left side is more complex if left-associative expressions are considered simpler; if right-associative expressions are considered simpler, then the left side is less complex than the right side. In this sense, there is sometimes a choice for certain equations. If all equational axioms can be oriented into terminating rewrite rules (meaning that simplification using such rewrite rules always terminates), then simplification by rewriting itself serves as a useful heuristic for checking whether the given two terms are equal.

Notice that there is another way to simplify the left side of the above equation, which is to first simplify using the axiom in Eq. (2), giving a new equation

$$0 + -(-(u)) = u$$

Clearly, both u and $0 + -(-(u))$ are normal forms of $(u + -(u)) + -(-(u))$ with respect to the above axioms considered as left-to-right rewrite rules. This would imply that $u = 0 + -(-(u))$ is an equation following from the above axioms as well.

For attempting proofs of other equations from equational axioms, the following two properties of equations are helpful. The first property is whether equational axioms can be collectively oriented into terminating rewrite rules. The second property is whether the rewrite rules thus obtained have the so-called *Church–Rosser* or *confluence* property: that is, given an expression, no matter how it is simplified using rewrite rules, if there is a normal form, that normal form is unique. Both of these properties are undecidable in the general case. However, for terminating rewrite rules, the confluence property is decidable.

As discussed earlier, the equational axioms of groups can be oriented as terminating rewrite rules (by going from left to right). These rewrite rules do not have the confluence property, as we saw above that the expression $(u + -(u)) + -(-(u))$ has two different normal forms.

If a set of rewrite rules is terminating and confluent, then every expression has a normal form, and further, that normal form is unique; unique normal forms are also called *canonical* forms. For such rewrite rules, it is easy to decide whether an equational formula follows from the axioms or not: compute the unique normal forms of the expressions on the two sides of the equality; if the normal forms are identical, the equational formula is a theorem; otherwise, it is not a theorem.

For groups, the following set of rewrite rules is terminating and confluent; this system thus serves as a decision procedure for equational formulas involving $+, -, 0$:

$$
\begin{aligned}
x + 0 &\rightarrow x && (1)\\
x + -(x) &\rightarrow 0 && (2)\\
(x + y) + z &\rightarrow x + (y + z) && (3)\\
-(0) &\rightarrow 0 && (4)\\
-(-(x)) &\rightarrow x && (5)\\
-(x) + x &\rightarrow 0 && (6)\\
0 + x &\rightarrow x && (7)\\
x + (-(x) + y) &\rightarrow y && (8)\\
-(x) + (x + y) &\rightarrow y && (9)\\
-(x + y) &\rightarrow -(y) + -(x) && (10)
\end{aligned}
$$

Given a set of equational axioms, it is sometimes possible to generate an *equivalent* terminating and confluent rewriting system from them. The generated system is equivalent to the input system in the sense that the equations provable from the equations corresponding to rewrite systems are precisely the equations provable from the original equational axioms. This can be done using *completion* procedures.

In the next subsection, we discuss heuristics for ensuring/checking termination of rewrite rules. In the following subsection, we discuss the concepts of *superpositions* and *critical pairs* for checking the confluence of terminating rewrite rules. This is followed by a discussion of completion procedures for generating an equivalent confluent and terminating rewrite system from a given terminating rewrite system. Finally, we discuss advanced concepts relating to generalizations of these techniques when semantic information of some function symbols must be exploited. The special focus is on systems in which certain function symbols have the *associative and commutative* (*AC*) properties.

These methods and heuristics can be implemented and tried on a variety of examples. We have built an automated reasoning program called *Rewrite Rule Laboratory* (*RRL*) for checking termination of a large class of rewrite systems, as well as for generating a confluent and terminating rewrite system from a given rewrite system using completion. This program has been used for solving many nontrivial problems in equational inference, and has also been used in a variety of applications, including: automatic verification of hardware arithmetic circuits, such as the SRT division algorithm widely believed to have been implemented in the Pentium chip; software verification; analysis of database integrity constraints; checking consistency; and completeness of behavioral specifications of abstract data types. For details and citations, an interested reader may refer to Refs. 1 and 2.

Termination of Rewriting. Checking for termination of a set of rewrite rules is undecidable, in general. In fact, the termination of a single rewrite rule is undecidable, as a Turing machine can be encoded using just one rewrite rule. However, for a large class of interesting rewrite systems, heuristics have been developed to check their termination.

As stated above, one approach for checking termination is to define a complexity measure on expressions by mapping them to a well-founded set (i.e., a set that does not admit any infinite descending chain) such that for every rule, its right side is of smaller measure than its left side. Since these rewrite rules are used for simplification, such a measure should satisfy some additional properties, namely, in any context (larger expression), whenever the left side of a rule is replaced by the right side, the measure reduces, and similarly,

the measure of every instance of the right side of a rule is always smaller than the measure of the corresponding instance of its left side.

In their seminal paper discussing a completion procedure, Knuth and Bendix 3 introduced a measure by associating *weights* with expressions by assigning weights to function symbols. They required that for an expression $s > t$, every variable in s must have at least as many occurrences as in t . This idea was extended by Lankford to design a complexity measure by associating polynomials with function symbols.

A more commonly used measure is *syntactic*, based on a precedence relation among function symbols. Assuming that function symbols can be compared, terms built using these function symbols are compared by recursively comparing their subterms. These *path* orderings are quite powerful in the sense that termination of all primitive recursive function definitions, as well as other definitions such as of Ackermann's function, which grows faster than any primitive recursive function, can be established using these orderings. For a function symbol that takes more than one argument, if two terms have that function symbol as their root, then the arguments can be compared without considering their order, left-to-right order, right-to-left order, or any other permutation. A commonly used path ordering based on these ideas is the *lexicographic recursive path ordering* (4); this is the ordering implemented in our theorem prover *RRL*.

Let $\succ_{\mathrm{f}}$ be a well-founded precedence relation on a set of function symbols F ; function symbols can also have equivalent precedence, written $f \sim_{\mathrm{f}} g$. The lexicographic recursive path ordering (with status) $\succ_{\mathrm{rpo}}$ extends $\succ_{\mathrm{f}}$ to a well-founded ordering on terms:

Definition 1. $s = f(s_1, \ldots, s_m) \succ_{\mathrm{rpo}} g(t_1, \ldots, t_n) = t$ iff one of the following holds.

(1) $f \succ_{\mathrm{f}} g$, and $s \succ_{rpo} t_j$ for all j $(1 \leq j \leq n)$.

(2) For some i $(1 \leq i \leq n)$, either $s_{\mathrm{i}} \sim_{\mathrm{rpo}} t$ or $s_i \succ_{rpo} t$

(3) $f \sim_{\mathrm{f}} g, f$ and g have multiset status, and $\{\{s_1, \ldots, s_m\}\} \succ_{mul} \{\{t_1, \ldots, t_n\}\}$

(4) $f \sim_{\mathrm{f}} g, f$ and g have left-to-right status, then $(s_1, \ldots, s_m) \succ_{lex} (t_1, \ldots, t_{\mathrm{n}})$, and $s \succ_{rpo} t_{\mathrm{j}}$ for all j $(1 \leq j \leq n)$; if f and g have right-to-left status, then $(s_m, \ldots, s_1) \succ_{lex} (t_n, \ldots, t_1)$ and $s \succ_{rpo} t_j$ for all j $(1 \leq j \leq n)$.

Of course, $s \succ_{rpo} x$ if s is nonvariable, and the variable x occurs in s . Term $s \sim_{\mathrm{rpo}} t$ if and only if $f \sim_{\mathrm{f}} g$ and

(1) f and g have multiset status, and $\{\{s_1, \ldots, s_{\mathrm{m}}\}\} = \{\{t_1, \ldots, t_{\mathrm{n}}\}\}$, or

(2) $f \sim_{\mathrm{f}} g, f \sim$ and g have left-to-right (similarly, right-to-left) status, and for each $1 \leq i \leq m$, $s_{\mathrm{i}} \succ_{rpo} t_{\mathrm{i}}$.

In the above definitions, $\succ_{rpo}$ is recursively defined using its multiset and sequence extensions. The multiset $M_1 \succ_{mul} M_2$ if and only if for every $t_i \in M_2 - M_1$, where $-$ is the multiset difference, there is an $s_{\mathrm{j}} \in M_1 - M_2$, $s_{\mathrm{j}} \succ_{rpo} t_i$. The sequence $(s_1, \ldots, s_m) \succ_{\mathrm{lex}} (t_1, \ldots, t_n)$ if and only if there is a $1 \leq i \leq m$ such that for all $1 \leq j < i$ one has $s_{\mathrm{j}} \succ_{rpo} t_j$ and $s_i \succ_{rpo} t_i$.

It can be shown that $\succ_{rpo}$ has the desired properties of a measure needed to ensure that rewrite rules used for simplification are indeed terminating. So if the left side of an equation is $\succ_{rpo}$ its right side, it can be oriented from left to right as a terminating rewrite rule. The properties are as follows:

- $\succ_{rpo}$ is well founded,
- $\succ_{rpo}$ is *stable* under substitutions, that is, if $s \succ_{rpo} t$, then for any substitution σ of variables, $\sigma(s) \succ_{rpo} \sigma(t)$, and
- $\succ_{rpo}$ is *preserved under contexts*, that is, $s \succ_{rpo} t$, then for any term c that has a position p, one has $c[p \leftarrow s] \succ_{rpo} c[p \leftarrow t]$ as well.

Local Confluence of Rewriting: Superposition and Critical Pairs. Since bidirectional equations are used as unidirectional rewrite rules for efficiently exploring search space, additional rewrite rules are needed to compensate for the lack of symmetric use of equations. For instance, we saw above an example of an expression, $(u + -(u)) + -(-(u))$, which could be simplified in two different ways—using the axiom in Eq. (3) or using the axiom in Eq. (2). Depending upon the axiom used, two different normal forms can be computed from the expression, thus showing that the rewrite system obtained from the axioms is not *confluent*. However, $0 + -(-(u)) = u$ can be inferred from the three axioms.

A rewrite system $\mathscr{R}$ is called *confluent* if for every term t, if t is simplified by $\mathscr{R}$ in *many* steps in two different ways to two different results, it is always possible to simplify the results to the same expression. Checking for confluence is, in general, undecidable. However, for a terminating rewrite system, the confluence check is equivalent to *local confluence*, which can be easily decided based on the concepts of *superpositions* and *critical pairs* generated by overlapping the left sides of rewrite rules. A rewrite system $\mathscr{R}$ is called *locally confluent* if for every term t, if t is simplified in a *single* step in two different ways to two different results, it is always possible to simplify the results to the same expression.

Theorem 2. If a terminating rewrite system $\mathscr{R}$ is locally confluent, then $\mathscr{R}$ is confluent.

Definition 3. Give two rules $L_1 \to R_1$ and $L_2 \to R_2$, not necessarily distinct, such that a nonvariable subterm of L_1 at position p unifies with L_2 with a most general substitution (unifier) σ, $\sigma(L_1)$ is called a superposition of $L_2 \to R_2$ into $L_1 \to R_1$ and

$$\langle \sigma(R_1), \sigma(L_1)[p \leftarrow \sigma(R_2)] \rangle$$

is called the associated critical pair.

(Unification of terms is defined in the next section.)

Theorem 4. Give a rewrite system $\mathscr{R}$ if for every critical pair $\langle c_1, c_2 \rangle$ generated from every pair of $L_1 \to R_1$, $L_2 \to R_2$ in $\mathscr{R}$, c_1 and c_2 have the same normal form, then $\mathscr{R}$ is locally confluent.

As an illustration, consider the axioms in Eqs. (2) and (3) above, viewed as left-to-right unidirectional rules. Using *unification* (see the next section), the superposition $(x + -(x)) + z$ [of rules 2 and 3 corresponding to the axioms in Eqs. (2) and (3), respectively] can be generated from the overlap of the two left sides; when axiom 2 is applied on it, the result is $0 + z$, but if the axiom in Eq. (3) is applied on the same expression, the result is $x + (-(x) + z)$. The pair $\langle x + (-(x) + z), 0 + z \rangle$ is the critical pair obtained from the superposition. Such pairs are critical in determining the confluence property; hence the name. In this example, the expressions in the above critical pair have different normal forms (both $0 + z$ and $x + (-(x) + z)$ are already in normal form). So the rewrite rules corresponding to the three group axioms are not confluent, even though they are terminating.

In contrast, the system of 10 rewrite rules given above is confluent; it can be shown that every superposition generated from possible overlaps between the left sides of these rewrite rules generates critical pairs in which the two terms have the same normal form. For instance, terms in the pair $\langle x + (-(x) + z), 0 + z \rangle$ reduce to the same normal form using the ten rewrite rules above.

Identifying which superpositions are essential and which are redundant for checking local confluence as well as in completion procedures has been a fruitful area of research in rewrite-rule-based automated deduction. Implementation of such results speeds up generation of canonical rewrite systems using the completion procedure as discussed below. As the reader might have observed, superpositions and critical pairs are defined by unifying a nonvariable subterm of the left side of a rule with the left side of another rule, since variable subterms always result in pairs that rewrite to the same expression. This research of identifying and discarding unnecessary inferences has recently been found useful in other approaches to automated deduction as well, including resolution-based calculi.

Completion Procedure: Making Rewrite Systems Canonical. As the reader might have guessed, if a terminating rewrite system is not confluent, it may be possible to make it confluent by augmenting it with additional rewrite rules obtained from normal forms of critical pairs. An equation between the two different normal forms of the terms in a critical pair obtained from a superposition is an equational consequence of the original rules. Including it in the original set of equations does not, in any way, change the equational theory of the original system. An equational theory associated with a set of equations is defined as the set of all equations that can be derived from the original set of equations using the rules of equality and instantiation of variables.

This process of adding new equational consequences generated from superpositions and critical pairs from rewrite rules is called *completion*. Every nontrivial new equation generated must be oriented into a terminating rewrite rule. If this process terminates successfully, then the result is a terminating and confluent rewrite system (also called a *canonical* or *complete* rewrite system), which serves as a decision procedure for the equational theory of the original set of equations. An equation $s = t$ is an equational consequence of the original system if and only if the normal forms of s and t with respect to the canonical rewrite system are the same. A canonical rewrite system thus associates canonical forms with congruence classes induced by a set of equations. For example, from the equational axioms defining groups, the above set of ten rewrite rules can be obtained by completion from the original set of three rewrite rules; this set can be generated by our theorem prover *RRL* in less than a second.

A completion procedure can be viewed as an implementation of an inference system consisting of the following inference steps (5). Each inference step transforms a pair consisting of a set $\mathscr{E}$ of equations and a set $\mathscr{R}$ of rules. The initial state is $\mathscr{E}_0$ and $\mathscr{R}_0$, with $\mathscr{R}_0$ usually being $\{\ \}$, the empty set. An inference step transforms $\langle \mathscr{E}_\mathrm{i}, \mathscr{R}_\mathrm{i} \rangle$ to $\langle \mathscr{E}_\mathrm{i} + 1, \mathscr{R}_{i+1} \rangle$ using a termination ordering $\succ$ on terms, as follows:

(1) *Process an Equation.* Given an $e_1 = e_2$ in $\mathscr{E}_\mathrm{i}$, let n_1 and n_2 be, respectively, normal forms of e_1 and e_2 using $\mathscr{R}_\mathrm{i}$.

(1) *Delete a Redundant Equation.* If $n_1 \equiv n_2$, then $\mathscr{E}_{i+1} = \mathscr{E}_\mathrm{i} - \{e_1 = e_2\}$ and $\mathscr{R}_{i+1} = \mathscr{R}_\mathrm{i}$ ($n_1 \equiv n_2$ stands for n_1 and n_2 being identical.)
(2) *Add a New Rule.*

(1) If $n_1 \succ n_2$, then $\mathscr{E}_{i+1} = \mathscr{E}_\mathrm{i} - \{e_1 = e_2\}$ and $\mathscr{R}_{i+1} = \mathscr{R}_\mathrm{i} \cup \{n_1 \to n_2\}$.
(2) If $n_2 \succ n_1$, then $\mathscr{E}_{i+1} = \mathscr{E}_i - \{e_1 = e_2\}$ and $\mathscr{R}_{i+1} = \mathscr{R}_i \cup \{n_2 \to n_1\}$.

Introduce a New Function. Let f be a new function symbol not in the theory, and let $x_1, \ldots, x_j$ be the common variables appearing in n_1 and n_2 .

(1) If $n_1 \succ f(x_1, \ldots, x_j)$, then $\mathscr{E}_{i+1} = \mathscr{E}_\mathrm{i} - \{e_1 = e_2\} \cup \{n_2 = f(x_1, \ldots, x_j)\}$ and $\mathscr{R}_{i+1} = \mathscr{R}_\mathrm{i} \cup \{n_1 \to f \sim (x_1, \ldots, x_j)\}$.
(2) If $f(x_1, \ldots, x_j) \succ n_1$, then $\mathscr{E}_{i+1} = \mathscr{E}_\mathrm{i} - \{e_1 = e_2\} \cup \{n_2 = f(x_1, \ldots, x_j)\}$ and $\mathscr{R}_{i+1} = \mathscr{R}_i \cup \{f(x_1, \ldots, x_j) \to n_1\}$. This choice should not be taken. Almost always, $f(x_1, \ldots, x_j)$ should be made the right side of introducing rules.

Add Critical Pairs. Given two (not necessarily distinct) rules $l_1 \to r_1$ and $l_2 \to r_2$ in $\mathscr{R}_i$,

$$\mathscr{R}_{i+1} = \mathscr{R}_i$$

and

$$\mathscr{E}_{i+1} = \mathscr{E}_i \cup \{c_1 = c_2 \mid \langle c_1, c_2 \rangle \text{ is a critical pair of } l_1 \to r_1 \text{ and } l_2 \to r_2\}$$

Normalize Rules. Given two distinct rules $l_1 \to r_1$ and $l_2 \to r_2$ in $\mathscr{R}_i$:

(1) If $l_1 \to r_1$ rewrites l_2, then $\mathscr{E}_{i+1} = \mathscr{E}_i \cup \{l_2 = r_2\}$ and $\mathscr{R}_{i+1} = \mathscr{R}_i - \{l_2 \to r_2\}$. The rule $l_2 \to r_2$ is thus deleted from $\mathscr{R}_i$ and inserted as an equation into $\mathscr{E}_i$.
(2) If $l_1 \to r_1$ rewrites r_2, then $\mathscr{E}_{i+1} = \mathscr{E}_i$ and $\mathscr{R}_{i+1} = \mathscr{R}_i - \{l_2 \to r_2\} \cup \{l_2 \to r'_2\}$, where r'_2 is a normal form of r_2 using $\mathscr{R}_i$.

The above steps can be combined in many different ways to generate a complete system when all critical pairs among rules have been considered. The resulting algorithm must be *fair* in the sense that critical pairs among all pairs of rules must eventually be considered. Many useful heuristics have been explored and developed to make completion faster. The order in which rules are considered for computing superpositions, the order in which critical pairs are processed, how new rules are used to simplify already generated rules, and so on, appear to affect the performance of completion considerably. For some early work on this, the reader may consult Ref. 6.

There are at least two ways a completion procedure can fail to generate a canonical rewrite system:

(1) An equational consequence is generated that cannot be oriented into a terminating rewrite rule.
(2) Even if all equational consequences generated from critical pairs during completion are orientable, the process of adding new rules does not appear to terminate.

The first condition could arise for many reasons. The new equation, when oriented either way, might result, in conjunction with other rules, in nontermination of rewriting. Or the ordering heuristics might not be powerful enough to establish the termination of the rewrite system when augmented with the new rewrite rule. It is also possible that the generated equation could not be made into a rewrite rule because either side has extra variables that the other side does not have.

In some cases, it is helpful to split such an equation by introducing a new function symbol (to stand for the operation corresponding to one of the sides of the equation); this is done in the "Introduce a New Function" step above. For instance, the following single axiom can be shown to characterize free groups using completion:

$$(x/((((x/x)/y)/z)/(((x/x)/x)/z))) = y$$

where/corresponds to the right division operation. During completion of the above axiom, an equation

$$x/x = y/y$$

is generated, which suggests that a new constant can be introduced. That constant turns out to satisfy the properties of the identity. Similarly, an inverse function symbol is introduced, finally leading to the following

canonical system:

$$y/y \rightarrow f2 \quad (11)$$
$$f1(y,x) \rightarrow f3(y) \quad (12)$$
$$x/f2 \rightarrow x \quad (13)$$
$$f3(f2) \rightarrow f2 \quad (14)$$
$$f3(f3(x)) \rightarrow x \quad (15)$$
$$f2/y \rightarrow f3(y) \quad (16)$$
$$((y/z)/f3(z)) \rightarrow y \quad (17)$$
$$((y/f3(y1))/y1) \rightarrow y \quad (18)$$
$$f3((y/x)) \rightarrow (x/y) \quad (19)$$
$$(y/(x/z)) \rightarrow ((y/f3(z))/x) \quad (20)$$

Above, $f2$ behaves as the identity 0; $f3$ behaves as the inverse $-$; $f1$ is an extra function symbol introduced during completion; $x + y$ can be defined as $x/f3(y)$. All the rewrite rules corresponding to the canonical system of free groups presented earlier are equational consequences of the above canonical system.

Another approach for handling nonorientable equations is discussed below in which such equations are processed semantically. Finally, an equation could be kept as is in $\mathscr{E}_i$, and its instances could be oriented and used for simplification as needed by including them in $\mathscr{R}_i$; this is the approach taken in *unfailing* completion (7).

The second condition above (non-termination of completion) cannot, in general, be avoided, since the decision problem of equational theories is unsolvable. In some cases, however, introducing a new function symbol to stand for an expression is helpful even though intermediate equations can be oriented as terminating rewrite rules. With the help of this new symbol, it may be possible to generate a finite canonical system though no such system can be generated without it (see Ref. 8 for such an example of a finitely presented semigroup).

Forward versus Backwards Reasoning. A completion procedure employs *forward reasoning* and *saturation*. That is, additional consequences are derived from the original set of axioms, without considering conjectures/goal(s) being attempted. And this process is continued until the resulting set of derivations is completely saturated.

A completion procedure can also be used in a goal-directed way. A conjecture to be attempted is negated, and a proof by contradiction is attempted. Axioms interact with each other by forward reasoning, generating new consequences. They can also interact with the negation of the goal, and attempt to generate a contradiction using *backward reasoning*. This approach based on the completion procedure can be shown to be semidecidable for determining membership in an equational theory. In other words, if a conjecture is provable, then barring any difficulties arising due to nonorientability of new equations,[3]

a proof by contradiction can always be generated by the completion procedure.

To illustrate using the group example again, one way to attempt a proof of $0 + x = x$ is to

(1) first complete the above axioms using a completion procedure that generates a complete set of rewrite rules, and then
(2) simplify both sides of the conjecture, and check for identity.

If a completion procedure does not terminate, then the second step will never be performed. To circumvent this problem, an obvious heuristic is to do step 2 whenever a new rule is generated, to see whether with the existing set of rules, the conjecture can be proved by rewriting. The same approach can be tried in a uniform

way by adding the negation of the (Skolemized) conjecture (for the example, $0 + a \neq a$), and then running completion on the axioms and the negated conjecture, and looking for a contradiction.

Unorientable Equations. So far, all equations are assumed to be oriented as terminating rewrite rules. In many applications, one often has to consider function symbols that are commutative and/or associative. Commutative axioms (and associative axioms in conjunction with commutative axioms) are nonterminating when used for simplification. For example, consider the axioms defining abelian groups, which, in addition to the three axioms in Eqs. (1, ,), include the commutative axiom as well:

$$x + y = y + x$$

One approach for handling axioms such as commutativity and associativity is to integrate them into the definitions of rewriting (simplification) and superpositions. In the definition of rewriting, instead of looking for an exact match of the left side of a rewrite rule $L \rightarrow R$, matching is done modulo such axioms. In the discussion below, we assume that certain function symbols have the associative and commutative properties.

A term t rewrites at position p using a rewrite rule $L \rightarrow R$ (where t and L could have occurrences of function symbols with associative and commutative properties) if there is a $t' =_{AC} t$ such that $t'/p =_{AC} \sigma(L)$, where $=_{AC}$ is the congruence relation defined by the associative–commutative properties of those function symbols on terms; t is then said to rewrite to $t[p \leftarrow \sigma(R)]$.[4]

For instance, $-(u) + (u + w)$ can be simplified by associative–commutative simplification using the axiom $x + -(x) = 0$ to $0 + w$, which simplifies further using the axiom $x + 0 = x$ to w . Notice that the term $-(u) + (u + w)$ is first rearranged using the associativity property of $+$ to the equivalent term $(-(u) + u) + w$. The subterm $-(u) + u$ then matches modulo commutativity to $x + -(x)$. Similarly, the result $0 + w$ matches modulo commutativity to $x + 0$.

Similarly, superpositions among the left sides of rewrite rules are defined using unification modulo axioms. Further, to ensure confluence of terminating rewrite rules, additional checks must be made (which can be viewed as considering specialized superpositions between each rule and the axioms to account for the semantics) (9,10). In the next section, we discuss unification modulo associative–commutative properties as well. Termination heuristics must be generalized appropriately also; termination of rewrite rules modulo axioms, such as associative–commutative, means defining a well-founded ordering on equivalence classes defined by the axioms (11).

Below is a list of rewrite rules of abelian groups which is canonical modulo AC.

$$\begin{aligned} x + 0 &\rightarrow x && (1)\\ x + -(x) &\rightarrow 0 && (2)\\ -(0) &\rightarrow 0 && (4)\\ -(-(x)) &\rightarrow x && (5)\\ -(x + y) &\rightarrow -(y) + -(x) && (10) \end{aligned}$$

This list is much smaller than the list for (general, abelian as well as nonabelian) groups, since one rewrite rule here, namely $x + 0 \rightarrow x$, is equivalent to both $x + 0 \rightarrow x$ and $0 + x \rightarrow x$ because of the commutativity property of $+$. Using the above rewrite rules, it can proved that $-(x + y) = -(x) + -(y)$, a property true for abelian groups but not for nonabelian groups.

Below is another list of canonical rewrite rules of abelian groups. Notice that the rule in Eq. (10) above is oriented in the opposite direction below:

$$x + 0 \rightarrow x \qquad (1)$$
$$x + -(x) \rightarrow 0 \qquad (2)$$
$$-(0) \rightarrow 0 \qquad (4)$$
$$-(-(x)) \rightarrow x \qquad (5)$$
$$x + -(x + y) \rightarrow -(y) \qquad (21)$$
$$-(y) + -(x) \rightarrow -(x + y) \qquad (22)$$
$$-(x + -(y)) \rightarrow y + -(x) \qquad (23)$$

The issue of redundant superpositions becomes very critical when confluence check and completion procedure modulo a set of axioms are considered. In general, there are a lot more superpositions to consider, because unlike the "ordinary" case, there can be many most general unifiers of two terms. Further, rewriting modulo axioms is a lot more expensive than "ordinary" rewriting. It was control over this redundancy which led us to easily prove ring commutative problems using *RRL* (12). In his EQP program, McCune extensively exploited heuristics for discarding redundant superpositions to establish that Robbins algebras are Boolean (13), thus solving a long-standing open problem in algebra and logic.[5]

Without these optimizations, it is unclear whether EQP would have been able to settle this long-standing conjecture.

The concept of a completion procedure and related properties generalize as well. For instance, from the first two axioms of groups discussed above (since the associativity axiom in addition to the commutativity axiom is semantically built-in), the canonical system consisting of the five rules in Eqs. (1), (2), (4), (5), (10) above can be generated using the completion procedure for associative–commutative function symbols. Our theorem prover *RRL* can generate this canonical system in a few seconds. For details about an associative–commutative completion procedure as well as for an $\mathscr{E}$completion procedure, where $\mathscr{E}$ is a first-order theory for which $\mathscr{E}$-unification and $\mathscr{E}$-rewriting are decidable, an interested reader can consult 9,10. In a later section, we discuss the Gröbner basis algorithm, which is a specialized completion procedure for finitely generated polynomial rings in which coefficients are from a field. This completion procedure has the nice property that it always terminates.

Extensions. We have discussed proofs in equational theories, in which only the properties of equality are used. When function symbols are defined by recursive equations on inductively (recursively) defined data structures, equations must be proved by induction as well. For example, if addition on numbers is defined recursively as

$$x + 0 = x, \qquad x + s(y) = s(x + y)$$

where s is the successor function on numbers, then $0 + x = x$ cannot be proved equationally from the definition. But $0 + x = x$ can be proved by induction; that is, no matter what ground term built using 0 and s is substituted for x in the above equation, the resulting equation is an equational inference of the two equations defining $+$. Proofs by induction have been found useful in verification and specification analysis.

There are many approaches for mechanizing proofs of equational formulae by induction, in which some of the variables are ranging over inductively (recursively) defined data structures such as natural numbers, integers, lists, sequences, or trees (14,15). In the following sub-subsection, we will briefly review the *cover-set* approach implemented in *RRL* (15), which is closely related to the approach in 14 in that the induction

scheme used in attempting a proof of a conjecture is derived from the well-founded ordering used to show the termination of the definitions of function symbols appearing in the conjecture.

We will not be able to discuss full first-order inference, in which properties of other logical connectives are used for deduction. This is a very active area of research, with many conferences and workshops. First-order theorem proving can be mechanized equationally as well. In fact, this was proposed by the author in collaboration with Paliath Narendran (16), in which quantifier-free first-order formulae are written as polynomials over a Boolean ring, using $+$ to represent exclusive or and $*$ to represent conjunction. Our theorem prover *RRL* includes an implementation of this approach to first-order theorem proving.

Inductive Inference: Cover-Set Method. The cover-set method for mechanizing induction is based on analysis of the definitions of function symbols appearing in a conjecture. The definition of a function symbol as a finite set of terminating rewrite rules is used to design an induction scheme based on the well-founded ordering used to show the termination of the function definition. The induction hypotheses are generated from the recursive calls in the definition, which are lower in the well-founded order. In the case of competing induction variables and induction schemes, heuristics are employed to pick the variable and the associated induction scheme most likely to succeed. Most induction theorem provers use backtracking so that if one particular choice fails, another choice can be attempted.

Let us start with a simple example. Assume that functions $+$, $*$, and x^y are defined on natural numbers, generated by 0 and s, where s is the successor operation to denote incrementing its argument by 1, as follows:

$$\begin{aligned} x + 0 &= x \\ x + s(y) &= s(x + y) \\ x * 0 &= 0 \\ x * s(y) &= x + (x * y) \\ x^0 &= s(0) \\ x^{s(y)} &= x * x^y \end{aligned}$$

Without assuming any additional properties of $+$, $*$, x^y, we wish to prove from the above definitions that

$$x^{(y+z)} = x^y * x^z$$

It is easy to see that the above defining equations, when oriented from left to right as terminating rewrite rules, are confluent as well. Further, since the two sides of the above conjecture are already in normal form, it is clear that the conjecture is not provable by equational reasoning; that is, the conjecture is not in the equational theory of the definitions.

The conjecture can, however, be proved by induction, using the property that every natural number can be generated *freely* by 0 and s . Our theorem prover *RRL*, for instance, can prove the above conjecture automatically without any guidance or help; it generates the needed intermediate lemmas, including the associativity and commutativity of $*$, $+$, and so on.

While attempting a proof by induction of a conjecture (by hand or automatically), a number of issues need to be considered:

(1) Which variable in a conjecture should be selected for performing induction? Associated with this choice is determining an induction scheme to be used.

(2) What mechanisms can be attempted for generalizing intermediate subgoals so as to have stronger lemmas, which are likely to be more useful?
(3) How does one ensure whether progress is being made while trying subgoals generated from following a particular approach, and when should an alternate approach be attempted instead?
(4) When should one give up?

In the above example, there are three candidates for an induction variable. By performing definition analysis, it can be easily determined that if x or y is chosen as a induction variable, a proof attempt will get stuck, since the definitions above are given using recursion on the second argument of $+$, $*$, and x^y . Thus the most promising variable to be used for doing induction is z .

The induction scheme to be used is suggested by the definitions of the function symbols $+$ and x^y, in which z appears as an argument. Since each of these definitions can be proved to be terminating, a well-founded ordering used to show termination of the definition can be used to design an induction scheme. For this example, the induction schemes suggested by the definitions of $+$ and x^y are precisely the principle of mathematical induction, namely, that to prove the above conjecture, it suffices to show that the conjecture can be proved for the case when

(1) $z = 0$—the *basis subgoal*, and
(2) $z = s(z')$, using the induction hypothesis obtained by making $z = z'$—the *induction step subgoal*.

In general, for each rewrite rule in the function definition used for designing the induction scheme, a subgoal is generated. A rewrite rule whose right side does not have a recursive call to the function gives a basis subgoal. A rewrite rule whose right side invokes recursive calls to the function gives an induction step, in which the changing arguments in the recursive calls generate substitutions for producing potentially useful induction hypotheses. In general, there can be many basis subgoals, and in an induction subgoal there can be many induction hypotheses.

For the above example, the basis subgoal can be easily proved by normalizing it using the rewrite rules. The induction subgoal after normalization produces

$$x * x^{(y+z')} = x^y * (x * x^{z'})$$

assuming the induction hypothesis

$$x^{(y+z')} = x^y * x^{z'}$$

Using the induction hypothesis, the conclusion in the subgoal simplifies to

$$x * (x^y * x^{z'}) = x^y * (x * x^{z'})$$

Most induction theorem provers, including *RRL*, would attempt to generalize this intermediate conjecture using a simple heuristic of abstracting common subexpressions on the two sides by new variables. For instance, the above intermediate conjecture would be generalized to

$$x * (u * v) = u * (x * v)$$

This conjecture cannot be proved equationally either, but can be proved by induction. The variable v is chosen as the induction variable by recursion analysis. The basis subgoal can be easily proved. The induction step subgoal leads to the following intermediate subgoal to be attempted:

$$x * (u + (u * v')) = u * (x + (x * v'))$$

assuming

$$x * (u * v') = u * (x * v')$$

In the conclusion above, it is not even clear how to use the induction hypothesis.

Another induction on the variable v' (which is an obvious choice based on recursion analysis) leads to a basis subgoal when $v' = 0$:

$$x * u = u * x$$

the commutativity of $*$. The induction step, after generalization, leads to an intermediate conjecture:

$$x + y = y + x$$

which can be easily established by induction. Using these intermediate conjectures, the original intermediate conjecture

$$x * (u + (u * v')) = u * (x + (x * v'))$$

is proved, from which the proof of the main goal follows.

As the reader might have guessed, the major challenge during proof attempts by induction is to generate/speculate lemmas likely to be found useful for the proof attempt to make progress.

More details about the cover-set induction method can be found in Ref. 15; see Ref. 2 for the use of the cover-set method for mechanically verifying parametrized generic arithmetic circuits of arbitrary data width. For mechanizing inductive inference about Lisp-like functions, an interested reader may consult Ref. 14.

Equation Solving Over Terms: Unification

In the last section, we discussed whether an equation can be deduced from a set of equations using the properties of equality, with the assumption that any expression can be substituted for a variable, that is, variables in hypothesis equations are assumed to be universally quantified. In equation solving, one is often interested in a particular instance of such variables; that is, variables in equations are assumed to be existentially quantified. In this section, we focus on this aspect of symbolic computation.

Given a set of equations over terms involving variables and function symbols, we are interested in solving these equations, that is, finding a substitution for variables appearing in the equations so that after the substitution is made, both sides of every equation become identical. At first, nothing is assumed about the meaning of the function symbols appearing in the equations (such symbols are called *uninterpreted*). In a later subsection, we assume that as in the case of $+$ in abelian groups, some of the function symbols have

the associative–commutative properties. In a subsequent section, we discuss solving polynomial equations, and assume even more about the operators, namely that $+$, $*$ are, respectively, addition and multiplication on numbers.

For example, consider an equation

$$f(a, g(x, y)) = f(x, z)$$

This equation can be solved, and it has many solutions. One solution is

$$\{x \leftarrow a,\ \ z \leftarrow g(a, y)\}$$

In fact, this solution is the most general solution, and any other solution can be obtained by instantiating the variables in it.

Another equation,

$$f(a, g(x, y)) = f(x, g(b, z))$$

does not have any solution, since x cannot be made equal to both a and b (recall that the function symbols are assumed to be uninterpreted, so it cannot be assumed that a could possibly be equal to b) .

In the previous section, we saw an application of unification for considering overlaps among the left sides of rewrite rules to compute superpositions and critical pairs. Later, we will summarize other applications of unification as well.

Simple Unification. Consider a finite set of equations

$$\{s_1 = t_1, \ldots,\ \ s_k = t_k\}$$

The goal is to find whether they have a common *solution*; i.e., whether there is a substitution σ for variables in $\{s_i, t_i \mid 1 \leq i \leq k\}$ such that for each i, $\sigma(s_i)$ and $\sigma(t_i)$ are the same. If so, what is a most general solution; that is, what is a solution from which all other solutions can be found by instantiating the variables? A solution of these equations is called a *unifier*, and a most general solution is called a *most general unifier* (mgu).

Just as in the case of solving linear equations (in linear algebra), it is necessary to be clear about the *simplest* equations for which there is a solution as well as for equations that do not have any solution.

Similarly to $x = 3$ in linear algebra, the equation $x = t$, where x does not appear in t, has a solution, and in fact, the most general solution is $\{x \leftarrow t\}$. Such an equation is said to be in *solved form.* Similarly to $3 = 0$ having no solution, the equation $f(\ldots) = g(\ldots)$, where f , g are different function symbols, has no solution. No substitution for variables can make the two sides equal, as no assumption can be made about the properties of distinct function symbols.

Also, an equation $x = t$, where t is a nonvariable term with an occurrence of x, has no solution, since no substitution for x can make the two sides of the equation equal. After the substitution, the size of the left side is not equal to the size of the right side.

The general problem of solving a finite set of equations can be transformed into those of the above simple equations by a sequence of transformation steps. During the transformation, if a simple equation $x = t$, with t having no occurrence of x, is identified, then the partial solution obtained so far can be extended by including

$\{x \leftarrow t\}$ (*solution extension* step). In addition, t is substituted for x in the remaining equations yet to be solved. Equations of the form $t = t$ can be deleted, as solutions are not affected (*deletion* step).

If a simple equation $x = t$, where t is nonvariable and includes an occurrence of x, is generated, there is no solution to the system of equations under consideration. Similarly, an equation $f \sim(\ldots) = g(\ldots)$, in which $f \sim$, g are different function symbols, has no solution either. These two cases are the *no-solution* steps.

An equation of the form $f(u_1, \ldots, u_i) = f(v_1, \ldots, v_i)$ is replaced by the set of equations $\{u_1 = v_1, \ldots, u_i = v_i\}$ (*decomposition* step), since every solution to the original equation is also a solution to the new set of equations and vice versa.

The above transformation steps (decomposition, solution extension, deletion, and no solution) can be repeated in any order (nondeterministically) until either the no-solution condition is detected or a *solved* (*triangular*) form $\{x_1 \leftarrow w_1, \ldots, x_j \leftarrow w_j\}$ is obtained, in which for each $1 \leq i \leq j$, x_i does not appear in w_h, $h \geq i$.

The termination of these steps follows from the following observations:

- If the system of equations has no solution, then during the transformation, an unsolvable equation is generated, which would eventually be recognized by the no-solution step,
- whenever a simple equation $x = t$, where x does not occur in t, is included in a solved form, the number of variables under consideration goes down (even though the size of the problem may increase because of the substitution, unless proper data structures with bookkeeping are chosen), and
- a decomposition step reduces the problem size.

A measure that lexicographically combines the number of unsolved variables and the problem size reduces with every transformation step.

The order in which these transformation steps are performed determines the complexity of the algorithm. An algorithm of linear time complexity is discussed in Ref. 17; it is rarely used, due to the considerable overhead in implementing it. A typical implementation is of n^2 (quadratic) or $n \log n$ complexity, where n is the sum of the sizes of all the terms in the original problem. The main trick is to keep track of variables that have the same substitution; this can be done using a union-find data structure.

It can be easily shown that a set of equations either does not have a solution, or has a solution, in which case, then the most general solution (mgu) is unique up to the renaming of the (independent) variables (variables not being substituted for).

Simple unification is fundamental in automated reasoning and other areas in artificial intelligence. For example, superposition and critical-pair construction used in checking confluence, as well as in the completion procedure discussed in the section on equational inference, use a unification algorithm for identifying terms that can be simplified in either of two different ways. Resolution theorem provers as well as provers based on other approaches also use unification as the main primitive. Unification is the main computation mechanism in logic programming languages, including Prolog. Unification is also used for type inference and type checking in programming languages such as ML.

Associative–Commutative (*AC*) Unification. The above algorithm for simple unification assumed no semantics for the function symbols appearing in equations. That is why simple equations such as $x = t$, where x appears in t, as well as $f(u_1, \ldots, u_i) = g(v_1, \ldots, v_j)$, where f , g are distinct symbols, cannot be solved. If we assume some properties of function symbols and constants, then some of these equations may be solvable. For example, since for any x we have $x = x + 0$ over the integers, any substitution for x is a solution to this equation over the integers. Similarly, $x * y = x + 1$ has a solution over natural numbers: $\{ x \leftarrow 1, y \leftarrow 2 \}$, even though the top function symbols of the terms on the two sides of the equation are different.

Unification algorithms have been developed for solving equations in which some of the function symbols have specific interpretations whereas other symbols may be uninterpreted. Procedures have been proposed for

solving the general E-unification problem in which equations are solved in the presence of *interpreted symbols*, whose semantics are given by an equational theory generated by a set E of equations.

Below, we briefly review a particular but very useful case of solving equations over algebraic structures when some of the function symbols have the associative–commutative (AC) properties. The use of an associative–commutative unification/matching algorithm was key in McCune's EQP theorem prover settling the Robbins algebra conjecture (13).

Consider a finite set of equations

$$\{s_1 = t_1, \ldots, s_k = t_k\}$$

in which some of the function symbols are known to be AC. Except for the decomposition step, all transformation steps discussed above for the simple unification problem are still applicable.

For a commutative function symbol f, an equation of the form $f(u_1, u_2) = f(v_1, v_2)$ has two possible most general solutions—one in which it is attempted to make u_1, u_2 equal to v_1, v_2, respectively, and the other in which it is attempted to make u_1, u_2 equal to v_2, v_1, respectively. In general, there are many solutions possible. Each of these possibilities can lead to a different most general unifier. So in the presence of commutative function symbols, there can be exponentially many most general unifiers of a set of equations. In fact, it is easy to construct examples of equations with commutative function symbols for which the number of most general solutions is exponential in the size of the input.

If f is associative as well, then the problem gets even more interesting. As a simple example, consider

$$x + y + z = u + u + u$$

where $+$ is an AC function symbol. There are many most general solutions of the above equation. The problem is related to the partitioning problem. In one most general solution x, y, z, and u all have the same substitution, say w ; in another, the substitution for u can be $u_1 + u_2$, whereas x gets u_1, y gets $u_2 + u_2$, and z gets $u_1 + u_1 + u_2$; of course, there are many other possibilities as well.

As the reader must have observed, for an AC symbol f, its occurrences appearing as arguments to f (i.e., an argument of f has f itself as the outermost symbol) can be *flattened*. An AC f can be viewed as an n-ary function symbol instead of a binary function symbol.

Consider an equation of the form $f(u_1, \ldots, u_i) = f(v_1, \ldots, v_j)$ where f is AC and no u_i or v_j has f as its outermost symbol. This equation cannot be decomposed as before, since the order of arguments is irrelevant; also notice that i need not be equal to j . Many different decomposition may be possible.

As shown in Ref. 18, such decompositions can be done by building a *decision tree* that records all possible different choices made. For every nonvariable argument u_k, it must be determined whether u_k will be unified with (made equal to) some nonvariable argument v_l with the same outermost symbol, or will be part of a solution for some variable v_l . Such choices can be enumerated systematically with some decision paths leading to a solution, whereas others may not give any solution. Similarly, for every variable u_k, it must be determined whether it will unify with another variable and/or what its top level symbol will be in a solution. Every possible choice must be pursued for generating a *complete* set of unifiers (from which every unifier can be generated by instantiating variables in some element in the set) unless it can be determined that a particular choice will not lead to a solution.

Partial solutions are built this way until equations of the form $f(u_1, \ldots, u_i) = f(v_1, \ldots, v_j)$ in which every argument is either a variable or a constant are generated. These equations are transformed to linear diophantine equations, for which nonzero nonnegative solutions are sought (18). Since constants stand for nonvariable subterms, a solution for a variable x should not include a constant that stands for a subterm in which x occurs.

The decision tree is so constructed that a complete set of *AC* unifiers of s and t is the union of complete sets of *AC* unifiers of the unification problem corresponding to the leaf node resulting from each path. The termination of the algorithm is obvious. Further, there is considerable flexibility and the possibility of using heuristics to speed up the computation as well as to discard *a priori* paths leading to leaf nodes not resulting in any solutions.

Computing a complete basis of nonnegative solutions of simultaneous linear diophantine equations can be done in exponentially many steps. *AC* unifiers can be constructed by considering every possible subset of such a basis that assigns a nonzero value to every variable. In the worst case, double-exponentially many steps must be performed. Since there are exponentially many leaf nodes in a decision tree in the worst case, the complexity of the algorithm has an upper bound of double-exponentially many steps [i.e., there is a polynomial $p(n)$, where n is the input size, such that the number of steps is $O(2^{2^{p(n)}})$]. This also gives a double-exponential bound on the size of a complete set of *AC* unifiers. In fact, there exist simple equations (generalizations of the equation describing the partitioning problem given above) for which this bound on the number of most general *AC* unifiers, as well as the number of steps for computing them, is optimal.

To illustrate the main steps of the above algorithm, consider an equation $s = t$, where

$$s = h(*(+(x, a), +(y, a), +(z, a)), x)$$
$$t = h(*(+(w, w, w), z, z), x)$$

and $+$ and $*$ are the only *AC* function symbols. Since h is assumed to be uninterpreted, the decomposition step applies and we get the equation $*(+(x, a), +(y, a), +(z, a)) = *(+(w, w, w), z, z)$ as well as $x = x$. The second equation is trivial (i.e., it is always true no matter what substitution is made) and is discarded by the deletion step.

Consider now

$$*(+(x, a), +(y, a), +(z, a)) = *(+(w, w, w), z, z) \qquad (24)$$

A decision tree can be built based on what arguments of $*$ on both sides are made equal. One possibility is to make $+(x, a) = +(w, w, w)$. Under this assumption, the above equation simplifies to

$$*(+(y, a), +(z, a)) = *(z, z)$$

The latter equation does not have any solution, as any possible solution would have to satisfy the equation $z = +(z, a)$, which has no solution.

Similarly, making $+(y, a) = +(w, w, w)$ also does not lead to any solution.

The next possibility is to make $+(z, a) = +(w, w, w)$, which simplifies Eq. (24) to

$$*(+(x, a), +(y, a)) = *(z, z)$$

From this equation, we have $z = +(x, a) = +(y, a)$, giving a solution $\{ x \leftarrow y \}$. Substituting for z in $+(z, a) = +(w, w, w)$ gives $+(y, a, a) = +(w, w, w)$. This path thus leads to the following set of equations:

$$\{z = +(x, a),\ \ z = +(y, a),\ \ +(y, a, a) = +(w, w, w)\}$$

The equation

$$+(y, a, a) = +(w, w, w)$$

has two most general solutions:

(1) $\{ y \leftarrow w \}$, which also makes $\{ w \leftarrow a \}$, thus producing $\{ x = y = w \leftarrow a \}$
(2) $\{ w \leftarrow +(v_1, a) \}$, which also makes $\{ x = y \leftarrow +(v_1, v_1, v_1, a) \}$

For this example, there are two most general unifiers:

$$\{w \leftarrow a,\ x \leftarrow a,\ y \leftarrow a,\ z \leftarrow +(a, a)\},$$
$$\{w \leftarrow +(v_1, a),\ x \leftarrow +(v_1, v_1, v_1, a),$$
$$y \leftarrow +(v_1, v_1, v_1, a),\ z \leftarrow +(v_1, v_1, v_1, a, a)\}$$

An algorithm for computing a complete set of *AC* unifiers is discussed in detail in Ref. 18, where its complexity analysis is also given. For function symbols that, in addition to being *AC*, have properties such as identity and idempotency, unification algorithms are discussed in Ref. 19, where their complexity is also given.

Other Aspects of Equation Solving. The discussion thus far has focused on solving equations over *first-order* terms, that is, equations in which variables range over terms, and function symbols cannot be variables. However, methods have been developed for solving equations over *higher-order* terms, that is, terms that are built with two types of variables: variables that range over terms, also called *first-order* variables, and variables that range over functions and functionals, called *higher-order* variables, An allowable substitution for a first-order variable is a term, whereas a function expression (also called a λ-expression) is substituted for a higher-order variable.

For illustration, consider a very simple equation in which f and x are variables, and 0 is a constant symbol; in contrast to x, $f\sim$ is a function variable:

$$f(x) = 0$$

This equation has many most general solutions, including the following two: $f = (\lambda v.0)$ to stand for the constant function 0, together with $f\sim = (\lambda u.u)$ to stand for the identity function; and $x = 0$.

Higher-order unification and equation solving have been useful in many applications, including program synthesis, program transformation, mechanization of proofs by induction (particularly for generation of intermediate lemmas), generic and higher-order programming, and mechanization of different logics. Theorem provers such as HOL, Isabelle, and NuPRL have been designed that use higher-order unification and matching as primitive inference steps. A logic-based programming language, λ-Prolog, has been designed for facilitating some of these applications. For more details about different approaches for solving equations over higher-order terms as well as their applications, the reader may consult Ref. 20.

Narrowing is a particular approach for solving equations; a by-product of the narrowing method is that it can also be used for solving unification problems with respect to a set of equations from which a canonical rewrite system can be generated.

Assume that E is a finite set of equations, from which a finite canonical rewrite system R can be generated. A term s is said to *narrow* to a term s' with respect to R if there is a rewrite rule $l \rightarrow r$ in R, and a nonvariable

subterm s/p at position p in s, such that s/p and l unify by the most general unifier σ, and $s' = \sigma(s[p \leftarrow r])$. To check whether $s = t$ can be solved [i.e., a substitution σ for variables in s and t can be found such that $\sigma(s)$ and $\sigma(t)$ are equivalent with respect to E], both s and t are narrowed using R so that narrowing sequences from s and t converge. In that case, a substitution solving $s = t$ can be generated from the narrowing sequence.

Equation solving using narrowing is discussed in Ref. 21; see also Ref. 22, where basic narrowing is used to derive complexity results on equation solving. Narrowing can be viewed as a generalization of pseudodivision of a polynomial by a polynomial; pseudodivision is discussed in a later subsection on the characteristic-set approach for polynomial equation solving.

Polynomial Equation Solving

So far, we have discussed equational reasoning and equation solving in a general and abstract framework. In this section, we focus on equation solving in a concrete setting. All function symbols appearing in equations are interpreted; that is, the meaning of the symbols is known. This additional information is exploited in developing algorithms for solving equations.

Consider what we learn in high school about solving a system of linear equations. Functions $+$, $-$, and multiplication by a constant, as well as numbers, have the usual meaning. We learn methods for determining whether a system of linear equations has a solution or not. For solving equations, Gauss's method involves selecting a variable to *eliminate*, determining its value (in terms of other variables), eliminating the variable from the equations, and so on. With a little more effort, it is also possible to determine whether a system of equations has a single solution or infinitely many solutions. In the latter case, it is possible to study the structure of the solution space by classifying the variables into independent and dependent subsets, and specifying the solutions in terms of independent variables.

In this section, we discuss how nonlinear polynomial equations can be solved in a similar fashion, though not as easily. Below we briefly review three different approaches for symbolically solving polynomial equations—*resultants, characteristic sets*, and *Gröbner bases*. The last two approaches are related to each other, as well as to the equational inference approach based on completion discussed in the second section. The treatment of resultants is the most detailed in this section, since the material on multivariate resultants is not easily accessible. In contrast, there are books written on the Gröbner-basis and characteristic-set approaches (23 24,25).

Nonlinear polynomials are used to model many physical phenomena in engineering applications. Often there is a need to solve nonlinear polynomial equations, of if solutions do not have to be explicitly computed, it is necessary to study the nature of solutions. Many engineering problems can be easily formulated using polynomial with the help of extra variables. It then becomes necessary to eliminate some of those extra variables. Examples include implicitization of curves and surfaces, geometric reasoning, formula derivation, invariants with respect to transformation groups, robotics, and kinematics. To get an idea about the use of polynomials for modeling in different application domains, the reader may consult books by Morgan (26,27) Kapur and Mundy 28, and Donald et al. 29.

Resultant Methods. Resultant means the *result of elimination*. It is also the projection of intersection. Let us start with a simple example. Given two univariate polynomials $f(x), g(x) \in Q\ [x]$ of degrees m and n respectively, where Q; is the field of rational numbers—that is,

$$f(x) = f_n x^n + f_{n-1} x^{n-1} + \ldots f_1 x + f_0$$

and

$$g(x) = g_m x^m + g_{m-1} x^{m-1} + \ldots + g_1 x + g_0$$

—do f and g have common roots over the complex numbers, the algebraically closed extension of Q; ? Equivalently, do $\{ f(x) = 0, g(x) = 0 \}$ have a common solution? If the coefficients of f and g, instead of being rational numbers, are themselves polynomials in parameters, one is then interested in finding conditions, if any, on the parameters so that a common solution exists.

The above problem can be generalized to the elimination of many variables.

Resultant methods were developed in the eighteenth century by Newton, Euler, and Bezout, in the nineteenth century by Sylvester and Cayley, and the early parts of the twentieth century by Dixon and Macaulay. Recently, these methods have generated considerable interest because of many applications using computers. Sparse resultant methods have been discussed in Ref. 30. In Refs. 31 and 32, we have extended and generalized Dixon's construction for simultaneously eliminating many variables.

Most of the earlier work on resultants can be viewed as an attempt to extend simple linear algebra techniques developed for linear equations to nonlinear polynomial equations. A number of books on the theory of equations were written that are now out of print. Some very interesting sections were omitted by the authors in revised editions of some of these books because abstract methods in algebraic geometry and invariant theory gained dominance, and constructive methods began to be viewed as too concrete to provide any useful structural insights. A good example is the 1970 edition of van der Waerden's book on algebra (33), which does not include a beautiful chapter on elimination theory that appeared as the first chapter in the second volume of its 1940 edition. For an excellent discussion of the history of constructive methods in algebra, the reader is referred to an article by Abhyankar (34).

Resultant techniques can be broadly classified into two categories: (1) dialytic methods, in which a square system of equations is generated by multiplying polynomials by terms so that the number of equations equals the number of distinct terms appearing, and (2) differential methods, in which suitable linear combinations of polynomials are constructed, again resulting in a square system. Methods of Euler, Sylvester, Macaulay, and (more recently) sparse resultants fall in the first category, whereas methods of Bezout, Cayley, and Dixon and their generalization, along with other hybrid methods, fall into the second category.

Euler and Sylvester's Univariate Resultants. In 1764, Euler proposed a method for determining whether two univariate polynomial equations in x have a common solution. Sylvester popularized this method by giving it a matrix form, and since then, it has been widely known as Sylvester's method. Most computer algebra systems implement this method for eliminating a variable from two polynomials. It is based on the observation that for polynomials $f(x)$, $g(x)$ to have a common root, it is necessary and sufficient that there exist factors ϕ, ψ respectively of f, g such that $\deg(\phi) < n$ and $\deg(\psi) < m$ and

$$\frac{f}{\phi} = \frac{g}{\psi}$$

which is equivalent to $f * \psi - g * \phi = 0$. Since the above relation is true for arbitrary $f(x)$, $g(x)$, the coefficient of each power of x in the left side $f * \psi - g * \phi$ must be identically 0. This gives rise to $m + n$ equations in $m +$

n unknowns, which are the coefficients of terms in ϕ, ψ. This linear system gives rise to the Sylvester matrix:

$$R = \begin{pmatrix} f_0 & 0 & \dots & 0 & g_0 & 0 & \dots & 0 \\ f_1 & f_0 & \dots & 0 & g_1 & g_0 & \dots & \vdots \\ f_2 & f_1 & \dots & & g_2 & g_1 & \dots & \vdots \\ \vdots & \vdots & & & \vdots & \vdots & & \\ & & & & g_m & g_{m-1} & \dots & 0 \\ & & & & 0 & g_m & \dots & 0 \\ f_{n-1} & f_{n-2} & \dots & f_{n-m-1} & \vdots & \vdots & \dots & g_0 \\ f_n & f_{n-1} & \dots & f_{n-m} & \vdots & & & g_1 \\ 0 & f_n & \dots & f_{n-m+1} & \vdots & \vdots & \dots & g_2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \dots & \vdots \\ & & & & \vdots & \vdots & & \vdots \\ 0 & 0 & & f_n & 0 & 0 & \dots & g_m \end{pmatrix}$$

The existence of a nonzero solution implies that the determinant of the matrix R is zero.

The Sylvester resultant can be used successively to eliminate several variables, one at a time. One soon discovers that the method suffers from an explosive growth in the degrees of the polynomials generated in the successive elimination steps. If one starts out with n or more polynomials in $Q;[x_1, x_2, \ldots, x_n]$, whose degrees are bounded by d, polynomials of degree double-exponential in n (i.e., $d^2\ n$) can get generated in the successive elimination process. The technique is impractical for eliminating more than three variables.

Macaulay's Multivariate Resultants. Macaulay generalized the resultant construction for eliminating one variable to eliminating several variables simultaneously from a system of *homogeneous* polynomial equations (35).

In a homogeneous polynomial, every term is of the same degree. A term or power product of the variables $x_1, x_2, \ldots, x_n$ is $x^{\alpha 1}{}_1\, x^{\alpha}{}_2, \ldots, x^{\alpha}{}_{\mathrm{n}}$ with $\alpha_j \geq 0$, and its degree is $\alpha_1 + \alpha_2 + \ldots + \alpha_n$, denoted by $\deg(t)$, where $t = x\ \alpha_1\ x\ \alpha_{22} \ldots x\ \alpha^{\mathrm{n}}{}_n$.

Macaulay's construction is also a generalization of the determinant of a system of homogeneous linear equations. For solving nonhomogeneous polynomial equations, the polynomials must be homogenized first. This can be easily done by introducing an extra variable; every term in the polynomial is multiplied by the appropriate power of the extra variable to make the degree of every term equal to the degree of the highest term in the polynomial being homogenized. Methods based on Macaulay's matrix give out projective zeros of the homogenized system of equations, and they can include zeros at infinity.

The key idea is to show which power products are sufficient to be used as multipliers for the polynomials so as to produce a square system of linear equations in as many unknowns as the equations. The construction below discusses that.

Let $f_1, f_2, \ldots, f_n$ be n *homogeneous* polynomials in $x_1, x_2, \ldots, x_n$. Let $d_i = \deg(f_i)$, $1 \leq i \leq n$, and $d_{\mathrm{M}} = 1 + \Sigma_1{}^{\mathrm{n}}\ (d_i - 1)$. Let T denote the set of all terms of degree d_{M} in the n variables $x_1, x_2, \ldots, x_n$:

$$T = \{x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n} | \alpha_1 + \alpha_2 + \ldots + \alpha_n = d_M\}$$

and let

$$|T| = \binom{d_M + n - 1}{n - 1}$$

The polynomial f_1 is viewed as introducing the variable x_1 ; similarly, f_2 is viewed as introducing x_2, and so on. For f_1, the power products used to multiply f_1 to generate equations are all terms of degree $d_{\mathrm{M}} - d_1$, where d_1 is the degree of f_1 . For f_2, the power products used to multiply f_2 to generate equations are all terms of degree $d_{\mathrm{M}} - d_2$ that are not multiples of $x_1\ \mathrm{d}^1$; power products that are multiples of $x_1\ \mathrm{d}_1$ are considered to be taken care of while generating equations from f_1 . That is why the polynomial f_1, for instance, is viewed as introducing the variable x_1 .) Similarly, for f_3, the power products used to multiply f_3 to generate equations are all terms of degree $d_{\mathrm{M}} - d_3$ that are not multiples of $x_1\ \mathrm{d}^1$ or $x_2\ \mathrm{d}^2$, and so on. The order in which polynomials are considered for selecting multipliers results in different systems of linear equations.

Macaulay showed that the above construction results in $|T|$ equations in which power products in x_i 's of degree d_M (these are the terms in T) are unknowns, thus resulting in a square matrix A . This matrix is quite sparse, most entries being zero; nonzero entries are coefficients of the terms in the polynomials.

Let det(A) denote the determinants of A, which is a polynomial in the coefficients of the f_i 's. It is easy to see that det(A) contains the resultant, denoted as R, as a factor. This polynomial det(A) is homogeneous in the coefficients of each f_i ; for instance, its degree in the coefficients of f_n is $d_1\ d_2\ \ldots\ d_{n-1}$.

Macaulay discussed two ways to extract a resultant from the determinants of matrices A . The resultant R can be computed by taking the gcd of all possible determinants of different matrices A that can be constructed by ordering f_i's in different ways (i.e., viewing them as introducing different variables). However, this is quite an expensive way of computing a resultant.

Macaulay also constructed the following formula relating R and det(A) :

$$R\det(B) = \det(A)$$

where det(B) is the determinant of a submatrix B of A; B is obtained from A by deleting those columns labeled by terms not divisible by any $n - 1$ of the power products $\{\ x\ 1\mathrm{d}_1, x\ 2\mathrm{d}_2, \ldots, x_{\mathrm{nd}}\ n\ \}$, and deleting those rows that contain at least one nonzero entry in the deleted columns. See Ref. 35 for more details.

u -Resultants The above construction is helpful for determining whether a system of polynomials has a common projective zero as well as for identifying a condition on parameters leading to a common projective zero. If the goal is to extract common projective zeros of a nonhomogeneous system $S = \{f_1(x_1, x_2, \ldots, x_n), \ldots, f_n(x_1, x_2, \ldots, x_n)\}$ of n polynomials in n variables, this can be done using the construction of u-resultants discussed in Ref. 33.

Homogenize the polynomials using a new homogenizing variable x_0 . Let R_u denote the Macaulay resultant of the $n + 1$ homogeneous polynomials ${}^hf_1, {}^hf_2, \ldots, {}^hf_n, {}^hf_u$ in $n + 1$ variables $x_0, x_1, \ldots, x_n$ where ${}^hf_i$ is a homogenization of f_i, f_u is the linear form

$$f_u = x_0u_0 + x_1u_1 + \cdots + x_nu_n$$

and $u_0, u_1, \ldots, u_n$ are *new unknowns*. R_u is a polynomial in $u_0, u_1, \ldots, u_n$, homogeneous in u_i 's of degree $B = \Pi^{\mathrm{n}}{}_{\mathrm{i}=1}\ d_i$. R_u is known as the u-resultant of $\mathscr{F}$. It can be shown that R_u factors into *linear factors* over $\mathbb{C}$; that

is,

$$R_u = \prod_{j=1}^{B} (u_0\alpha_{0,j} + u_1\alpha_{1,j} + \cdots + u_n\alpha_{n,j})$$

and if $u_0\alpha_{0,j} + u_1\alpha_{1,j} + \ldots + u_{n\alpha n,j}$ is a factor of R_u, then $(\alpha_{0,j}, \alpha_{1,j}, \ldots, \alpha_{n,j})$ is a common zero of ${}^hf_1, {}^hf_2, \ldots, {}^hf_n$. The converse can also be proved: if $(\beta_{0,j}, \beta_{1,j}, \ldots, \beta_{n,j})$ is a common zero of ${}^hf_1, {}^hf_2, \ldots, {}^hf_n$, then $u_0\beta_{0,j} + u_1\beta_{1,j} + \ldots + u_n\beta_{n,j}$ divides R_u. This gives an algorithm for finding all the common zeros of ${}^hf_1, {}^hf_2, \ldots, {}^hf_n$.

Recall that B is precisely the Bezout bound on the number of common zeros of n polynomials of degree $d_1, \ldots, d_n$ when zeros at infinity are included and the multiplicity of common zeros is also taken into consideration.

Nongeneric Polynomial Systems. The above methods based on Macaulay's formulation do not always work, however. For a specific polynomial system in which the coefficients of terms are specialized, the matrix A may be singular or the matrix B may be singular. If $\mathcal{F}$ has infinitely many common projective zeros, then its u-resultant R_u is identically zero, since for every zero, R_u has a factor. Even if we assume that the f_i's have only finitely many affine common zeros, that is not sufficient, since the u-resultant vanishes whenever there are infinitely many common zeros of the homogenized polynomials ${}^hf_i$'s—finitely many affine zeros, but infinitely many zeros at infinity. Often, one or both conditions arise.

Grigoriev and Chistov 36 and Canny 37 suggested a perturbation of the above algorithm that will give all the affine zeros of the original system (as long as they are finite in number) even in the presence of infinitely many zeros at infinity. Let $g_i = {}^hf_i + \lambda x^{\mathrm{d}}{}_i$ for $i = 1, \ldots, n$, and $g_u = (u_0 + \lambda)x_0 + u_1x_1 + \ldots + u_nx_n$, where λ is a new unknown. Let $R_u(\lambda, u_0, \ldots, u_n)$ be the Macaulay resultant of $g_1, g_2, \ldots, g_n$ and g_u, regarded as homogeneous polynomials in $x_0, x_1, \ldots, x_n$. The polynomial $R_u(\lambda, u_0, \ldots, u_n)$ is called the *generalized characteristic polynomial*. Suppose R_u is considered as a polynomial in λ whose coefficients are polynomials in $u_0, u_1, \ldots, u_n$:

$$R_u(\lambda, u_0, \ldots, u_n) = \lambda^{\delta} + R_{\delta-1}\lambda^{\delta-1} + \cdots + R_k\lambda^k$$

where $k \geq 0$ and the R_i's are polynomials in $u_0, u_1, \ldots, u_n$. Then the trailing coefficient R_k of R_u has all the information about the affine zeros of the original polynomial system. If $k = 0$, then R_0 is the same as the u-resultant of the original polynomial system when there are finitely many projective zeros. However, if there are infinitely many zeros at infinity, then $k > 0$. As in the case of the u-resultant, R_k can be factored, and the affine common zeros can be extracted from these factors; R_k may also include extraneous factors.

This perturbation technique is inefficient in practice because of the extra variable λ introduced. As shown in Table 1 later, the method is unable to compute a result in practice even on small examples (it typically runs out of memory). We discuss another linear algebra technique in a subsequent subsection, called the *rank submatrix* construction, for computing resultants in the context of Dixon resultant formulation. This construction can be used for singular Macaulay matrices also, as shown on a number of examples discussed in Table 1.

Sparse Resultants. As stated above, the Macaulay matrix is sparse (most entries are 0) since the size T of the matrix is larger than the number of terms in a polynomial f_i. Further, the number of affine roots of a polynomial system does not directly depend upon the degree of the polynomials, and this number decreases as certain terms are deleted from the polynomials. This observation has led to the development of *sparse elimination* theory and *sparse resultants*. The theoretical basis of this approach is the so-called *BKK bound* (30) on the number of zeros of a polynomial system in a toric variety (in which no variable can be 0); this bound depends only on the *support* of polynomials (the structure of terms appearing in them) irrespective of their degrees. The BKK bound is much tighter than the Bezout bound used in the Macaulay's formulation.

The main idea is to refine the subset of multipliers needed using *Newton polytopes*. Let $\hat{f}$ be the square matrix constructed by multiplying polynomials in $F\hat{}$ so that the number of multipliers is precisely the number

Table 1. Empirical Data

Example	Application	No. of Poly-nomials	Dixon			Macaulay			Sparse			Gröbner
			Matrix Size	Cnstr. Time	*RSC* Time	Matrix Size	*GCP* Time	*RSC* Time	Matrix Size	Cnstr. Time	*RSC* Time	Macaulay Time
1	Geom. reason.	3	13	0.18	342	55	*	4608	34	0.3	566	65836
2	Formula deriv.	5	14	0.36	76	330	*	*	86	9.8	4094	585
3	Formula deriv.	3	5	0.08	*	15	N/R	*	14	0.2	*	*
4	Geometry	3	4	0.08	1021	15	*	*	14	0.2	*	15429
5	Random	4	6	0.06	*	84	N/R	*	27	0.7	*	2207
6	Implicitization	3	10	0.1	0.58	45	3741	13.15	16	0.2	0.71	1
7	Implicitization	3	18	0.45	47	66	N/R	604	55	2	519	12530
8	Implicitization	3	18	9.33	126	153	*	94697	54	1.9	1369	70485
9	Random	4	7	0.21	220	56	*	9972	24	1.2	412	13508
10	Random	5	36	1.1	8130	1365	*	*	151	40	201558	*
11	Implicitization	3	11	0.06	3.01	36	N/R	32.58	33	0.7	31.04	57
12	Equilibrium	4	5	0.05	0.24	20	N/R	1.53	20	0.7	2.36	5
13	Vision	6	6	3.3	0.05	792	*	*	60	248	10.53	6
14	Comput. Biol.	3	8	0.067	0.27	28	11.18	0.68	16	0.2	0.33	1

of distinct power products in the resulting set of polynomials. Given a polynomial f_i, the exponents vector corresponding to every term in f_i defines a point in n -dimensional Euclidean space $\mathbb{R}^n$. The Newton polytope of f_i is the convex hull of the set of points corresponding to exponent vectors of all terms in f_i . The *Minkowski sum* of two Newton polytopes Q and S is the set of all vector sums, $q + s$, $q \in Q$, $s \in S$.

Let $Q_i \subset \mathbb{R}^n$ be the Newton polytope (wrt X) of the polynomial p_i in $\mathscr{F}$. Let $Q = Q_1 + \ldots + Q_{n+1} \subset \mathbb{R}^n$ be the Minkowski sum of Newton polytopes of all the polynomials in $\mathscr{F}$. Let $\mathscr{E}$ be the set of exponents (lattice points of $\mathbb{Z}^n$) in Q (obtained after applying a small perturbation to Q to move as many boundary lattice points as possible outside Q).

Construction of $F^{\wedge}$ is similar to the Macaulay formulation—each polynomial p_i in $\mathscr{F}$ is multiplied by certain terms to generate $F^{\wedge}$ with $|\mathscr{E}|$ equations in $|\mathscr{E}|$ unknowns (which are terms in $\mathscr{E}$), and its coefficient matrix is the *sparse resultant matrix*. Columns of the sparse resultant matrix are labeled by the terms in $\mathscr{E}$ in some order, and each row corresponds to a polynomial in $\mathscr{F}$ multiplied by a certain term. A *projection operator* (a nontrivial multiple of the resultant, as there can be extraneous factors) for $\mathscr{F}$ is simply the determinant of this matrix; see discussion on extraneous factors below.

In contrast to the size of the Macaulay matrix ($|T|$), the size of the sparse resultant matrix is $|\mathscr{E}|$, and it is typically smaller than $|T|$, especially for polynomial systems for which the BKK bound is tighter than the Bezout bound. Canny, Emiris, and others have developed algorithms to construct matrices, using greedy heuristics, that may result in smaller matrices, but in the worst case, the size can still be $|\mathscr{E}|$.

Much like Macaulay matrices, for specialization of coefficients, the sparse resultant matrix can be singular as well—even though this happens less frequently than in the case of Macaulay's formulation.

Theoretically as well as empirically, sparse resultants can be used wherever Macaulay resultants are needed (unless one is interested in projective zeros that are not affine). Further, sparse resultants are much more efficient to compute than Macaulay resultants.

So far, we have used the dialytic method for setting up the resultant matrix and computing a projection operator. To summarize, the main idea in this method is to identify enough power products that can be used as multipliers for polynomials to generate a square system with as many equations generated from the polynomials as the power products in the resulting equations.

In the next few subsections, we discuss a related but different approach for setting up the resultant matrix, based on the methods of Bezout and Cayley.

Bezout and Cayley's Method for Univariate Resultants. In 1764, around the same time as Euler, Bezout developed a method for computing the resultant that is quite different from Euler's method discussed in the previous sub-subsection. Instead of generating $m + n$ equations as in Euler's method, Bezout constructed n equations (assuming $n \geq m$) as follows:

(1) First multiply $g(x)$ by x^{n-m} to make the result a polynomial of degree n .
(2) From $f(x)$ and $g(x)x^{n-m}$, construct equations in which the degree of x is $n - 1$, by multiplying:

(1) $f(x)$ by g_m and $g(x)x^{n-m}$ by f_n and subtracting,
(2) $f(x)$ by $g_m x + g_{m-1}$, and $g(x)x^{n-m}$ by $f_n x + f_{n-1}$ and subtracting,
(3) $f(x)$ by $g_m x^2 + g_{m-1} x + g_{m-2}$, and $g(x)x^{n-m}$ by $f_n x^2 + f_{n-1} x + f_{n-2}$ and subtracting, and so on.

This construction yields m equations. An additional $n - m$ equations are obtained by multiplying $g(x)$ by $1, x, \ldots, x^{n-m-1}$, respectively. There are n equations and n unknowns—the power products, $1, x, \ldots, x^{n-1}$.

In contrast to Euler's construction, in which the coefficients of the terms in equations are the coefficients of the terms in f and g, the coefficients in this construction are sums of 2×2 determinants of the form $f_i g_j - f_i g_j$.

Cayley reformulated Bezout's method, and proposed viewing the resultant of $f(x)$ and $g(x)$ as follows: If we replace x by α in both $f(x)$ and $g(x)$, we get polynomials $f(\alpha)$ and $g(\alpha)$. The determinant

$$\Delta(x, \alpha) = \begin{vmatrix} f(x) & g(x) \\ f(\alpha) & g(\alpha) \end{vmatrix}$$

is a polynomial in x and α, and is obviously equal to zero if $x = \alpha$, meaning that $x - \alpha$ is a factor of $\Delta(x, \alpha)$. The polynomial

$$\delta(x, \alpha) = \frac{\Delta(x, \alpha)}{x - \alpha}$$

is a degree $n - 1$ polynomial in α and is symmetric in x and α . It vanishes at every common zero x_0 of $f(x)$ and $g(x)$, no matter what values α has. So, at $x = x_0$, the coefficient of every power product of α in $\delta(x, \alpha)$ is 0. This gives n equations which are polynomials in x, and the maximum degree of these polynomilas is $n - 1$. Any common zero of $f(x)$ and $g(x)$ is a solution of these polynomial equations, and they have a common solution if the determinant of their coefficients is 0. It is because of the above formulation of $\delta(x, \alpha)$ that we have called these techniques differential methods.

Extraneous Factor. There is a price to be paid in using this formulation instead of the Euler–Sylvester formulation. The result computed using Cayley's method has an additional extraneous factor of $f_n{}^{n-m}$. This factor arises because Cayley's formulation is set up assuming both polynomials are of the same degree.

Except for a system of generic polynomials, almost all multivariate elimination methods rarely compute the exact resultant. Instead, they produce a *projection operator*, which is a nontrivial nonzero multiple of the resultant. The resultant, on the other hand, is the principal generator of the ideal of the projection operators. Sometimes it is possible to predict these extraneous factors from the structure of a polynomial system, as in the case of two polynomials from which a single variable is eliminated. In general, however, it is a major challenge to determine the extraneous factors. For some results on this topic, the reader may consult Ref. 38.

Dixon's Formulation for Elimination of Two Variables. Dixon showed how to extend Cayley's formulation to three polynomials in two variables. Consider the following three generic bidegree polynomials, which have

all the power products of the type $x^i y^j$ where $0 \le i \le m$, $0 \le j \le n$:

$$f(x,y) = \sum_{i,j} a_{ij} x^i y^j, \qquad g(x,y) = \sum_{i,j} b_{ij} x^i y^j$$
$$h(x,y) = \sum_{i,j} c_{ij} x^i y^j$$

Just as in the single-variable case, Dixon 39 observed that the determinant

$$\Delta(x,y,\alpha,\beta) = \begin{vmatrix} f(x,y) & g(x,y) & h(x,y) \\ f(\alpha,y) & g(\alpha,y) & h(\alpha,y) \\ f(\alpha,\beta) & g(\alpha,\beta) & h(\alpha,\beta) \end{vmatrix}$$

vanishes when α is substituted for x or β is substituted for y, implying that $(x-\alpha)(y-\beta)$ is a factor of the above determinant. The expression

$$\delta(x,y,\alpha,\beta) = \frac{\Delta(x,y,\alpha,\beta)}{(x-\alpha)(y-\beta)}$$

is a polynomial of degree $2m-1$ in α, $n-1$ in β, $m-1$ in x, and $2n-1$ in y . Since the above determinant vanishes when we substitute $x = x_0, y = y_0$ where (x_0, y_0) is a common zero of $f(x,y), g(x,y), h(x,y)$ into the above matrix, $\delta(x, y, \alpha, \beta)$ must vanish no matter what α and β are. The coefficients of each power product $\alpha^i \beta^j$, $0 \le i \le 2m-1$, $0 \le j \le n-1$, have common zeros that include the common zeros of $f(x,y), g(x,y), h(x,y)$. This gives $2mn$ equations in power products of x, y, and the number of power products $x^i y^j$, $0 \le i \le m-1$, $0 \le j \le 2n-1$, is also $2mn$. Using a simple geometric argument, Dixon proved that the determinant is in fact the resultant up to a constant factor.

If the polynomials $f(x,y), g(x,y), h(x,y)$ are not bidegree, it can be the case that the resulting Dixon matrix is not square or, even if square, is singular.

Kapur, Saxena, and Yang's Formulation: Generalizing Dixon's Formulation. Cayley's construction for two polynomials generalizes for eliminating $n-1$ variables from a system of n nonhomogeneous polynomials $f_1, \ldots, f_n$. A matrix similar to the above can be set up by introducing new variables $\alpha_1, \ldots, \alpha_{n-1}$, and its determinant vanishes whenever $x_i = \alpha_i$, $1 \le i < n$, implying that $(x_1 - \alpha_1) \ldots (x_{n-1} - \alpha_{n-1})$ is a factor. The polynomial δ, henceforth called the *Dixon polynomial*, can be expressed directly as the determinant

$$\delta(X,\Gamma) = \begin{vmatrix} f_{1,1} & \cdots & f_{1,n} \\ f_{2,1} & \cdots & f_{2,n} \\ \vdots & & \vdots \\ f_{n-1,1} & \cdots & f_{n-1,n} \\ f_1(\alpha_1,\alpha_2,\ldots,\alpha_{n-1}) & \cdots & f_n(\alpha_1,\alpha_2,\ldots,\alpha_{n-1}) \end{vmatrix}$$

where $\Gamma = \{\alpha_1, \alpha_2, \ldots, \alpha_{n-1}\}$, where for $1 \leq j \leq n-1$ and $1 \leq i \leq n$ we have

$$f_{j,i} = \frac{f_i(\alpha_1, \ldots, \alpha_{j-1}, x_j, x_{j+1}, \ldots, x_n) - f_i(\alpha_1, \ldots, \alpha_{j-1}, \alpha_j, x_{j+1}, \ldots, x_n)}{x_j - \alpha_j}$$

and where $f_i(\alpha_1, \ldots, \alpha_k, x_{k+1}, \ldots, x_n)$ stands for uniformly replacing x_j by α_j for $1 \leq j \leq k$ in f_i.

Let

$$\begin{aligned} u &= u + 0 = u + (-(u) + -(-(u))) \\ &= (u + -(u)) + -(-(u)) = 0 + -(-(u)) \end{aligned}$$

$\mathcal{F}$ be the set of all coefficients (which are polynomials in X) of terms in δ, when viewed as a polynomial in Γ. Terms in Γ can be used to identify polynomials in

$$\begin{aligned} u &= u + 0 = u + (-(u) + -(-(u))) \\ &= (u + -(u)) + -(-(u)) = 0 + -(-(u)) \end{aligned}$$

$\mathcal{F}$ based on their coefficients. A matrix D is constructed from F in which rows are labeled by terms in Γ and columns are labeled by the terms in polynomials in

$$\begin{aligned} u &= u + 0 = u + (-(u) + -(-(u))) \\ &= (u + -(u)) + -(-(u)) = 0 + -(-(u)) \end{aligned}$$

$\mathcal{F}$. An entry $d_{i,j}$ is the coefficient of the jth term in the ith polynomial, where polynomials in

$$\begin{aligned} u &= u + 0 = u + (-(u) + -(-(u))) \\ &= (u + -(u)) + -(-(u)) = 0 + -(-(u)) \end{aligned}$$

$\mathcal{F}$ and terms appearing in the polynomials can be ordered in any manner. We have called D the *Dixon matrix*.

$n+1$ nonhomogeneous polynomials $f_0, f_1, \ldots, f_n$ in $x_1, \ldots, x_n$ are said to be *generic n-degree* if there exist nonnegative integers $m_1, \ldots, m_n$ such that each $f_j = \Sigma_{i_1}^{m_1} \mathrm{m}_1 = 0 \ldots \Sigma\, i_n \mathrm{m}_\mathrm{n} = 0\, a_{j,i1}, \ldots, \mathrm{i}_n\, x_1{}^{\mathrm{i}}{}_1 \ldots x_n\, \mathrm{i}^{\mathrm{n}}$ *for* $1 \leq j \leq n$, where $a_j, i_{i1}, \ldots, i_n$ is a distinct parameter. For generic n-degree polynomials, it can be shown that the determinant of the Dixon matrix so obtained is a resultant (up to a predetermined constant factor).

If a polynomial system is not generic n-degree, and/or the coefficients of different power products are related (specialized), then its Dixon matrix is (almost always) singular, much as in the case of Macaulay matrices. This phenomenon of resultant matrices being singular is observed in all the resultant methods when more than one variable is eliminated. In the next sub-subsection, we discuss a linear algebra construction, the *rank submatrix*, which has been found very useful for extracting projection operators from singular resultant matrices.

As shown in Table 1 below comparing different methods, this construction seems to outperform other techniques based on perturbation—for example, the generalized characteristic polynomial construction for singular Macaulay matrices discussed above. Further, empirical results suggest that the extraneous factors in projection operators computed by this method are fewer and of lower degrees.

Rank Submatrix (RSC) Construction. If a resultant matrix is singular (or rectangular), a projection operator can be extracted by computing the determinants of its largest nonsingular submatrices. This method is applicable to Macaulay matrices and sparse matrices as well as Dixon matrices. The general construction (31) is as follows:

(1) Set up the resultant matrix R of $\mathscr{F}$.
(2) Compute the rank of R and return the determinant of any *rank submatrix*, a maximal nonsingular submatrix of R. The determinant computation of a rank submatrix can be done along with the rank computation.

It was shown in Refs. 31 and 32 that

Theorem 5. If a Dixon matrix of a polynomial system f has a column that is linearly independent of the others, the determinant of any rank submatrix is a *nontrivial* multiple of the resultant of F.

The above theorem holds in general, including for nongeneric polynomial systems whose coefficients are specialized in any way. If the coefficients are assumed to be generic, then any rank submatrix is a projection operator, since one of the columns in the Dixon matrix is linearly independent of others.

Comparison of Different Resultant Methods. Interestingly, even though the Dixon formulation is classical, it does exploit the sparsity of the polynomial system, as illustrated by the following results about the size of the Dixon matrix (40).

Let $N(P)$ denote the Newton polytope of any polynomial P in F. A system F of polynomials is called *unmixed* if every polynomial in F has the same Newton polytope. Let $\pi_i(A)$ be the *projection* of an n-dimensional set A to $n - i$ dimensions obtained by substituting 0 for all the first i dimensions. Let $mvol(F)$ [which stands for $mvol(N(P), \ldots, N(P))$] be the n-fold *mixed volume* of the Newton polytopes of polynomials in F in the unmixed case, $mvol(F) = n!\ vol\ (N\,(P))$, where $vol(A)$ is the volume of an n-dimensional set A.

Theorem 6. The number of columns in the Dixon matrix of an unmixed set F of $n + 1$ polynomials in n variables is

$$O\left(vol\left(\sum_{i=1}^{n}\pi_i(N(P))\right)\right)$$

A multihomogeneous system of polynomials of type $(\ l_1, \ldots, l_r; d_1, \ldots, d_r\ )$ is defined to be an unmixed set of $\Sigma_{i=1}{}^{r}\ l_i + 1$ polynomials in $\Sigma_{i=1}{}^{r}\ l_i$ variables, where

(1) $r =$ number of partitions of variables,
(2) $l_i =$ number of variables in the ith partition, and
(3) $d_i =$ total degree of each polynomial in the variables of the ith partition.

The size of the Dixon matrix for such a system can be derived to be

$$O\left(\frac{\prod_{i=1}^{r}\left(\sum_{j=1}^{i} l_j + 1\right)^{l_i}}{n+1}\prod_{i=1}^{r}\frac{d_i{}^{l_i}}{l_i!}\right)$$

Table 2. Timings—Direct Determinant Expansion in Maple

Example	Dixon *RSC*	Macaulay *GCP*	Macaulay *RSC*	Sparce *RSC*
3	6.56	N/R	26.5	76.2
4	1.6	*	84	290
5	0.9	N/R	850	23.4

For asymptotically large number of partitions, the size of the Dixon matrix is

$$O\left(\frac{mvol(\mathcal{F})}{n+1}\right)$$

which is proportional to the mixed volume of F for a fixed n .

The size of the Dixon matrix is thus much smaller than the size of Macaulay matrix, since it depends upon the BKK bound instead of the Bezout bound. It also turns out to be much smaller than the size of the sparse resultant matrix for unmixed systems:

$$O\left(vol\left(\sum_{i=0}^{n}\mathcal{N}(P)\right)\right)$$

Since the projections are of successively lower dimension than the polytopes themselves, the Dixon matrix is smaller. Specifically, the size of the Dixon matrix of multihomogeneous polynomials is of smaller order than their n -fold mixed volume, whereas the size of the sparse resultant matrix is larger than the n -fold mixed volume by an exponential multiplicative factor.

Table 1 gives some empirical data comparing different methods on a suite of examples taken from different application domains including geometric reasoning, geometric formula derivation, implicitization problems in solid and geometric modeling, computer vision, chemical equilibrium, and computational biology, as well as some randomly generated examples. Macaulay *GCP* stands for Macaulay's method augmented with the generalized characteristic polynomial construction (perturbation method) for handling singular matrices. Macaulay *RSC*, Dixon *RSC*, and Sparse *RSC* stand for, respectively, Macaulay's method, Dixon's method, and the sparse resultant method augmented with rank submatrix computation for handling singular matrices. All timings are in seconds on a 64 Mbit Sun SPARC10. An asterisk in a time column means either that the resultant could not be computed even after running for more than a day or the program ran out of memory. N/R in the *GCP* column means there exists a polynomial ordering for which the Macaulay and the denominator matrix are nonsingular and therefore *GCP* computation is not required. Determinant computations are performed using multivariate sparse interpolation. Examples 3, 4, and 5 consist of generic polynomials with numerous parameters and dense resultants. Interpolation methods are not appropriate for such examples, and timings using straightforward determinant expansion in Maple are in Table 2. Further details are given in Ref. 41.

We included timings for computing resultant using Gröbner basis construction (discussed in a later subsection) with block ordering (variables in one block and parameters in another) using the Macaulay system. Gröbner basis computations were done in a finite field with a much smaller characteristic (31991) than in the resultant computations. Gröbner basis results produce exact resultants, in contrast to the other methods, where the results can include extraneous factors.

As can be seen from the tables, all the examples were completed using the Dixon *RSC* method. Sparse *RSC* also solved all the examples, but always took much longer than Dixon *RSC*. Macaulay *RSC* could not complete two problems and took much longer than Sparse *RSC* (for most problems) and Dixon *RSC* (for all problems).

Characteristic-Set Approach. The second approach for solving polynomial equations is to use Ritt's *characteristic-set* construction 42. This approach has been recently extended and popularized by Wu Wentsun. Despite its success in geometry theorem proving, the characteristic-set method does not seem to have gotten as much attention as it should. For many problems, characteristic-set construction is quite efficient and powerful, in contrast to both resultants and the Gröbner basis method discussed in the next section.

Given a system S of polynomials, the characteristic-set algorithm transforms S into a *triangular form* S', much in the spirit of Gauss's elimination method, with the objective that the zero set of S' (the variety of S') is "roughly equivalent" to the zero set of S . (This is precisely defined later in this subsection.) From the triangular form, the common solutions can then be extracted by solving the polynomial in the lowest variable first, back-substituting the solutions one by one, then solving the polynomial in the next variable, and so on. A total ordering on variables is assumed. Multivariate polynomials are treated as univariate polynomials in their highest variable.

Similarly to elimination steps in linear algebra, the primitive operation used in the transformation is that of *pseudodivision* of a polynomial by another polynomial. It proceeds by considering polynomials in the lowest variables. (It is also possible to devise an algorithm that considers the variables in descending order.) For each variable x_i, there may be several polynomials of different degrees with x_i as the highest variable. GCD-like computations (dividing polynomials by each other) are performed first to obtain the lowest-degree polynomial, say h_i, in x_i . If h_i is linear in x_i, then it can be used to eliminate x_i from the rest of the polynomials. Otherwise, polynomials in which the degree of x_i is higher are pseudodivided by h_i to give polynomials that are of lower degree in x_i . The smallest remainder is then used to pseudodivide other polynomials, until no remainder is generated. This process is repeated until for each variable, a minimal-degree polynomial is generated such that every polynomial generated thus far pseudodivides to 0. The set of these minimal-degree polynomials for each variable constitutes a characteristic set.

If the number of equations in S is less than the number of variables (which is typically the case when elimination or projection needs to be computed), then the variable set $\{ x_1, \ldots, x_n \}$ is typically classified into two subsets: *independent* variables (also called *parameters*) and *dependent* variables. A total ordering on the variables is chosen so that the dependent variables are all higher in the ordering than the independent variables. For elimination, the variables to be eliminated must be classified as dependent variables. The construction of a characteristic set can be employed with a goal to generate a polynomial free of variables being eliminated. We denote the independent variables by $u_1, \ldots, u_k$ and dependent variables by $y_1, \ldots, y_i$, and the total ordering is, $u_1 < \ldots < u_k < y_1 < \ldots < y_i$, where $k + l = n$.

To check whether another equation, say $f = 0$, follows from S [that is, whether f vanishes on (most of) the common zeros of S], f is pseudodivided using a characteristic set S' of S . If the remainder of the pseudodivision is 0, then $f = 0$ is said to follow from S under the condition that the *initials* (the leading coefficients) of polynomials in S' are not zero. This use of characteristic-set construction is extensively discussed in Refs. 24,43, and 44 for geometry theorem proving. Wu 45 also discusses how the characteristic set method can be used to study the zero set of polynomial equations. These uses of the characteristic-set method are discussed in detail in our introductory paper (46). In Ref. 47, a method for constructing a family of irreducible characteristic sets equivalent to a system of polynomials is discussed; unlike Wu's method discussed in a later subsection, this method does not use factorization over extension fields.

The following sub-subsections discuss the characteristic set construction in detail.

Preliminaries. Assuming an ordering $y_1 < y_2 < \ldots < y_{i-1} < y_i < \ldots < y_n$, the *highest variable* of a polynomial p is y_i if $p \notin Q;[u_1, \ldots, u_k, y_1, \ldots, y_i]$ and $p \notin Q;[u_1, \ldots, u_k, y_1, \ldots, y_{i-1}]$; that is, y_i appears in p and

every other variable in p is $< y_i$. The *class* of p is then said to be i . A polynomial p is $\geq$ another polynomial q if and only if

(1) the highest variable of p, say y_i, is $>$ the highest variable of q, say y_j (i.e., the class of p is higher than the class of q), or
(2) the highest variable of p and q is the same, say y_i, and the degree of p in y_i is $\geq$ the degree of q in y_i .

A polynomial p is *reduced with respect to* another polynomial q if (1) the highest variable y_i of p is $<$ the highest variable of q, say y_j (i.e., $p < q$), or (2) $y_i \geq y_j$ and the degree of y_j in q is $<$ the degree of y_j in p .

A list C of polynomials ($p_1, \ldots, p_m$) is called a *chain* if either (1) $m = 1$ and $p_1 \neq 0$, or (2) $m > 1$ and the class of p_1 is > 0, and for $j > i$, p_j is of higher class than p_i and reduced with respect to p_i ; we thus have $p_1 < p_2 < \ldots < pm$. [A chain is the same as an *ascending set* defined by Wu 44.] A chain is a triangular form.

Pseudodivision. Consider two multivariate polynomials p and q, viewed as polynomials in the main variable x, with coefficients that are polynomials in the other variables. Let I_q be the initial of q . The polynomial p can be pseudodivided by q if the degree of q is less than or equal to the degree of p . Let $e =$ degree $(p) -$ degree$(q) + 1$. Then

$$I_q^e p = s\, q + r$$

where s and r are polynomials, and the degree of r in x is lower than the degree of q . The polynomial r is called the *remainder* (or *pseudoremainder*) of p obtained by dividing by q . It is easy to see that the common zeros of p and q are also zeros of the remainder r, and that r is in the ideal of p and q .

For example, if $p = xy^2 + y + (x + 1)$ and $q = (x^2 + 1)y + (x + 1)$, then p cannot be divided by q but can be pseudodivided as follows:

$$(x^2 + 1)^2(xy^2 + y + x + 1) = ((x^3 + x)y + (1 - x))((x^2 + 1)y + (x + 1)) + (x^5 + x^4 + 2x^3 + 3x^2 + x)$$

with the polynomial $x^5 + x^4 + 2x^3 + 3x^2 + x$ as the pseudoremainder.

If p is not reduced with respect to q, then p reduces to r using q by *pseudodividing* p by q, giving r as the remainder of the result of pseudodivision.

Characteristic Set Algorithm.

Definition 7. Given a finite set Σ of polynomials in $u_1, \ldots, u_k, y_1, \ldots, y_l$ a characteristic set Φ of Σ is defined to be either

(1) $\{p_1\}$, where p_1 a polynomial in $u_1, \ldots, u_k$, or
(2) a chain $\langle p_1, \ldots, p_l \rangle$, where p_1 is a polynomial in $y_1, u_1, \ldots, u_k$ with initial I_1, p_2 is a polynomial in $y_2, y_1, u_1, \ldots, u_k$ with initial $I_2, \ldots, p_l$ is a polynomial in $y_l, \ldots, y_1, u_1, \ldots, u_k$ with initial I_l, such that

(1) any zero of Σ is a zero of Φ, and
(2) any zero of Φ that is not a zero of any of the initials I_i is a zero of Σ .

Then $\text{Zero}(\Sigma) \subseteq \text{Zero}(\Phi)$ as well as $\text{Zero}(\Phi/\Pi_{i=1}^{l} I_i) \subseteq \text{Zero}(\Sigma)$, where using Wu's notation, $\text{Zero}(\Phi/I)$ stands for $\text{Zero}(\Phi) - \text{Zero}(I)$. We also have

$$\text{Zero}(\Sigma) = \text{Zero}\left(\Phi / \prod_{i=1}^{l} I_i\right) \cup \bigcup_i \text{Zero}(\Sigma \cup \{I_i\})$$

Ritt 42 gave a method for computing a characteristic set from a finite basis Σ of an ideal. A characteristic set Φ is computed from Σ by successively adjoining Σ with remainder polynomials obtained by pseudodivision. Starting with $\Sigma_0 = \Sigma$, we extract a minimal chain C_i from the set Σ_i of polynomials generated so far as follows. Among the subset of polynomials with the lowest class (with the smallest highest variable, say x_j), include in C_i the polynomial of the lowest degree in x_j . This subset of polynomials is excluded for choosing other elements in C_i . Among the remaining polynomials, include in C_i a polynomial of the lowest degree in the next class (i.e., the next higher variable), and so on. So C_i is a chain consisting of the lowest-degree polynomials in each variable in Σ_i .

Compute nonzero remainders of polynomials in Σ_i with respect to the chain C_i . If this remainder set is nonempty, we adjoin it to Σ_i to obtain Σ_{i+1} . Repeat the above computation until we have Σ_j such that every polynomial in Σ_j pseudodivides to 0 with respect to its minimal chain. The set Σ_j is called a *saturation* of Σ, and that minimal chain C_j of Σ_j is a characteristic set of Σ_j as well as Σ . The above construction is guaranteed to terminate since in every step, the minimal chain of Σ_i is $>$ the minimal chain of Σ_{i+1}, and the ordering on chains is well founded.

The above algorithm can be viewed as augmenting Σ with additional polynomials from the ideal generated by Σ, much like the completion procedures discussed in the section on equational inference, until a set Δ is generated such that

(1) $\Sigma \subseteq \Delta$,
(2) Σ and Δ generate the same ideal, and
(3) a minimal chain Φ of Δ pseudodivides every polynomial in Δ to 0.

There can be many ways to compute such a Δ . For detailed descriptions of some of the algorithms, the reader may consult. Ref. 24 42,43,44,46.

Definition 8. A characteristic set $\Phi = \{p_1, \ldots, p_l\}$ is *irreducible* over $Q;[u_1, \ldots, u_k, y_1, \ldots, y_l,]$ if for $i =$ 1 to l, p_i cannot be factored over Q_{i-1} where $Q_0 = Q;(u_1, \ldots, u_k)$, the field of rational functions expressed as ratios of polynomials in $u_1, \ldots, u_k$ with rational coefficients, and $Q_j = Q_{j-1}(\alpha_j)$ is an algebraic extension of Q_{j-1}, obtained by adjoining a root α_j of $p_j = 0$ to Q_{j-1} that is $p_j(\alpha_j) = 0$ in Q_j for $1 \leq j < i$.

If a characteristic set Φ of Σ is irreducible, then $\text{Zero}(\Sigma) = \text{Zero}(\Phi)$, since the initials of Φ do not have a common zero with Φ . Ritt defined irreducible characteristic sets.

Not only can different orderings on dependent variables result in different characteristic sets, but one ordering can generate a reducible characteristic set whereas another one gives an irreducible characteristic set. For example, consider $\Sigma_1 = \{(x^2 - 2x + 1) = 0, (x - 1)z - 1 = 0\}$. Under the ordering $x < z$, Σ_1 is a characteristic set; it is however reducible, since $x^2 - 2x + 1$ can be factored. The two polynomials do not have a common zero, and under the ordering $z < x$, the characteristic set Σ_2 of Σ_1 includes only 1.

In the process of computing a characteristic set from Σ, if an element of the coefficient field (a rational number if $l = n$) is generated as a remainder, this implies that Σ does not have a solution, or is *inconsistent*. For instance, the above Σ_1 is indeed inconsistent, since a characteristic set of Σ_1 includes 1 if $z < x$ is used.

If Σ does not have a solution, either

- a characteristic set Φ of Σ includes a constant [in general, an element $Q(u_1, \ldots, u_k)$], or
- Φ is reducible, and each of the irreducible characteristic sets includes constants [respectively, elements of $Q(u_1, \ldots, u_k)$].

Theorem 9. Given a finite set Σ *of polynomials, if (i)* its characteristic set Φ is irreducible *(ii)* Φ does not include a constant, and *(iii)* the initials of the polynomials in Φ do not have a common zero with Φ then Σ has a common zero.

Ritt was apparently interested in associating characteristic sets only with *prime* ideals. A prime ideal is an ideal with the property that if an element h of the ideal can be factored as $h = h_1 h_2$, then either h_1 or h_2 must be in the ideal. A characteristic set C of a prime ideal PI is necessarily irreducible. It has the desirable property that a polynomial p pseudodivides to 0 using C if and only if p is in PI . As discussed in the next sub-subsection, in contrast, for any ideal, prime or not, a polynomial reduces by its Gröbner basis to 0 if and only if the polynomial is in the ideal.

For ideals in general, it is not the case that every polynomial in an ideal pseudodivides to 0 using its characteristic set. For example, consider the characteristic set Σ_1 above with the ordering $x < z$; even though 1 is in its ideal, 1 cannot be pseudodivided at all using Σ_1 . It is also not the case that if a polynomial pseudodivides to 0 by a characteristic set, then it is in the ideal of the characteristic set, since for pseudodivision a polynomial can be multiplied by initials.

Elimination Using the Characteristic-Set Method. For elimination (projection) of variables from a polynomial system Σ, variables being eliminated are made greater than other variables in the ordering, and a characteristic set is computed from Σ using this ordering, with the understanding that the characteristic set will include a polynomial for each eliminated variable, as well as a polynomial in the independent parameters (i.e., the other variables not being eliminated).

As soon as a polynomial in the independent parameters is generated during the construction of a characteristic set, it is a candidate for a projection operator; the conditions under which this polynomial is zero ensure a common zero of the original system Σ . As the characteristic set computation proceeds, lower-degree simpler polynomials serving as projection operators may be generated. Thus the characteristic set so computed may include a lower-degree polynomial in the parameters, with fewer extraneous factors, along with the resultant. Just as with resultant-based approaches for elimination, the characteristic-set method does not generate the exact resultant. In contrast, as we shall see in the next section, the Gröbner basis method can be used to compute the exact eliminant (resultant).

Proving Conjectures from a System of Equations. A direct way to check whether an equation $c = 0$ follows (under certain conditions) from a system f of equations is to

- compute a characteristic set $\Phi = \{p_1, \ldots, p_l\}$ from f , and
- check whether c pseudodivides to 0 with respect to Φ .

If c has a zero remainder with respect to Φ, then the equation $c = 0$ follows from Φ under the condition that none of the initials used to multiply c is 0. In this sense, $c = 0$ "almost" follows from the equations corresponding to the polynomial system S . The algebraic relation between the conjecture c and the polynomials in Φ can be expressed as

$$I_1^{i_1} \cdots I_l^{i_l}\, c = q_1 p_1 + \cdots + q_l p_l$$

where I_j is the initial of $p_j, j = 1, \ldots, l$. It is because of this relation that c is not in the ideal generated by Φ or S .

To check whether $c = 0$ exactly follows from the equations corresponding to S (i.e., the zero set of c includes the zero set of S), it must be checked whether Φ is irreducible. If so, then $c = 0$ exactly follows from the equations; otherwise, a family of irreducible characteristic sets must be generated from S, and it must be checked that for each such irreducible characteristic set, c pseudodivides to 0. This is discussed in the next sub-subsection.

The above approach is used by Wu for geometry theorem proving (43,44). A geometry problem is algebraized by translating (unordered) geometric relations into polynomial equations. A characteristic set is computed from the hypotheses of a geometry problem. For plane Euclidean geometry, most hypotheses can be formulated as linear polynomials in dependent variables, so a characteristic set can be computed efficiently. A conjecture is pseudodivided by the characteristic set to check whether the remainder is 0. If the remainder is 0, the conjecture is said to be generically valid from the hypotheses; that is, it is valid under the assumption that the initials of the polynomials in the characteristic set are nonzero. The initials correspond to the degenerate cases of the hypotheses. If the remainder is not 0, then it must be checked whether the characteristic set is reducible or not. If it is irreducible, then the conjecture can be declared to be not generically valid. Otherwise, the zero set of the hypotheses must be decomposed, and then the conjecture is checked for validity on each of the components or some of the components. This method has turned out to be quite effective in proving many geometry theorems, including many nontrivial theorems such as the butterfly theorem, Morley's theorem, and Pappus's theorem. Decomposition is necessary when case analysis must be performed to prove a theorem, for example theorems involving exterior angles and interior angles, or incircles and outcircles. An interested reader may consult Ref. 24 for many such examples.

A refutational way to check whether $c = 0$ follows from f is to compute a characteristic set of $S \cup \{cz - 1 = 0\}$, where z is a new variable. This method is used in Ref. 48 for proving plane geometry theorems.

Decomposing a Zero Set into Irreducible Zero Sets. The zero set of a polynomial system Σ can be also computed exactly using irreducible characteristic sets. The zero set of Σ can be presented as a union of zero sets of a family of irreducible characteristic sets. Below, this construction is outlined.

A reducible characteristic set can be decomposed using factorization over algebraic extensions of $Q;(u_1, \ldots, u_k)$. In the case that any of the polynomials in a characteristic set can be factored, there is a branch for each irreducible factor, as the zeros of p_j are the union of the zeros of its irreducible factors. Suppose a characteristic set $\Phi = \{p_1, \ldots, p_l\}$ from Σ is computed such that for $i > 0$, $p_1, \ldots, p_i$ cannot be factored over $Q_0, \ldots, Q_{i-1}$, respectively, but p_{i+1} can be factored over Q_i . It can be assumed that

$$gp_{i+1} = p_{i+1}^{\mathbf{1}} \cdots p_{i+1}^{j}$$

where g is in $Q;[u_1, \ldots, u_k, y_1, \ldots, y_i]$ and where $p_{i+1}{}^1 \ldots p^j{}_{i+1} \in Q$; $[u_1, \ldots, u_k, y_1, \ldots, y_i, y_{i+1}]$ and these polynomials are reduced with respect to $p_1, \ldots, p_i$. Wu 43 proved that

$$\begin{aligned}&\mathrm{Zero}(\Sigma)\\ &= \mathrm{Zero}(\Sigma^{11}) \cup \cdots \cup \mathrm{Zero}(\Sigma^{ij}) \cup \mathrm{Zero}(\Sigma^{21}) \cup \cdots \cup \mathrm{Zero}(\Sigma^{2i})\end{aligned}$$

where $\Sigma^{1h} = \Sigma \cup \{p^h{}_{i+1}\}$, $1 \le h \le j$, and $\Sigma^{2h} = \Sigma \cup \{I_h\}$, where I_h is the initial of p_h, $1 \le h \le l$.

Characteristic sets are instead computed from new polynomial sets to give a system of characteristic sets. (To make full use of the intermediate computations already performed, Σ above can be replaced by a saturation Δ of Σ used to compute the reducible characteristic set Φ.) Whenever a characteristic set includes a polynomial that can factored, this splitting is repeated. The final result of this decomposition is a system of irreducible

characteristic sets:

$$\text{Zero}(\Sigma) = \bigcup_i \text{Zero}(\Phi_i / J_i)$$

where Φ_i is an irreducible characteristic set and J_i is the product of the initials of all of the polynomials in Φ_i .

In Ref. 47, another method for decomposing the zero set of a set of polynomials into the zero set of irreducible characteristic sets is discussed. This method does not use factorization over extension fields. Instead, the method computes characteristic sets à la Ritt using Duval's D5 algorithm for inverting a polynomial with respect to an ideal. The inversion algorithm splits a polynomial in case it cannot be inverted. The result of this method is a family of characteristic sets in which the initial of every polynomial is invertible, implying that each characteristic set is irreducible.

Complexity and Implementation. Computing characteristic sets can, in general, be quite expensive. During pseudodivision, coefficients of terms can grow considerably. Techniques from subresultant and GCD computations can be used for controlling the size of the coefficients by removing some common factors *a priori*. In the worst case, the degree of a characteristic set of an ideal is bounded from both below and above by an exponential function in the number of variables. Other results on the complexity of computing characteristic sets are given in Ref. 49.

We are not aware of any commercial computer algebra system having an implementation of the characteristic-set method. We implemented the method in the *GeoMeter* system (50,51), a programming environment for geometric modeling and algebraic reasoning. Chou developed an efficient implementation of the characteristic-set algorithm with many heuristics and compact representation of polynomials (24). His implementation has been used to prove hundreds of geometry theorems. This seems to suggest that despite the worst case complexity of computing characteristic sets being double-exponential, characteristic sets can be computed efficiently for many practical applications.

Gröbner Basis Computations. In this subsection, we discuss another method for solving polynomial equations using a Gröbner basis algorithm proposed by Buchberger 52. A Gröbner basis is a special basis of a polynomial ideal with the following properties:

(1) every polynomial in the ideal simplifies to 0 using its Gröbner basis, and
(2) every polynomial has a unique normal form (canonical form) using a Gröbner basis.

A Gröbner basis of an ideal is thus a canonical (confluent and terminating) system of simplification by the polynomials in the ideal.

A Gröbner basis algorithm can be used for elimination, as well as for generating triangular forms from which common solutions of a polynomial system can be extracted. A Gröbner basis of a polynomial ideal can also be used for analyzing many structural properties of the ideal as well as its associated zero set (variety); see Refs. 23 and 25 for details.

Variables in polynomials are totally ordered. But unlike the case of characteristic set construction, polynomials are viewed as multivariate rather than as univariate polynomials in their highest variables. A polynomial is used as a simplification (rewrite) rule in which its highest monomial is used to replace the remaining part of the polynomial. Simplification of a polynomial by another polynomial is thus defined differently from pseudodivision.

The discussion below assumes the coefficient field to be Q; the approach however carries over when coefficients come from any other field. Extensions of the approach have been worked out when the coefficients are from a Euclidean domain (53).

Reduction Using a Polynomial. Recall that a term or power product of the variables $x_1, x_2, \ldots, x_n$ is $x_1\ \alpha_1\ x_2\ \alpha_2\ \ldots\ x\ \alpha_{nn}$ with $\alpha_j \geq 0$ and its degree is $\alpha_1 + \alpha_2 + \ldots + \alpha_n$, denoted by deg($t$), where $t = x^{\alpha 1}{}_1\ x\ \alpha_{22}\ \ldots\ x_n\ \alpha_n$. Assume an ordering $x_1 < x_2 < \ldots < x_n$.

Total orderings (denoted by $>$) on terms are defined as those satisfying the following properties:

(1) *Compatibility with multiplication*. If t, t_1, t_2 are terms, then $t_1 > t_2 \Rightarrow t\ t_1 > t\ t_2$.
(2) *Termination.* There can be no strictly decreasing infinite sequence of terms such as

$$t_1 > t_2 > t_3 > \cdots$$

Term orderings satisfying property 2 are called *admissible* term orderings. Two commonly used term orderings are

(1) The *lexicographic order*, $\succ$l, in which terms are ordered as in a dictionary; that is, for terms $t_1 = x\alpha_{1_1}\ x\alpha_{22}\ \ldots\ x\alpha_{n_n}$ and $t_2 = x_1\beta_1\ x^{\beta_2}{}_2\ \ldots\ x\beta^n{}_n$, $t_1 \succ$l t_2 iff $\exists i \leq n$ such that $\alpha_j = \beta_j$ for $i < j \leq n$ and $\alpha_i > \beta_i$.
(2) The *degree order*, $\succ$d, in which terms are compared first by their degrees, and equal-degree terms are compared lexicographically; that is,

$$t_1 \succ_d t_2 \text{ iff} \Big(\deg(t_1 > \deg(t_2) \text{ or } \Big(\deg(t_1 = \deg(t_2) \text{ and } t_1 \succ_l t_2 \Big) \Big)$$

Given an admissible term order $>$, for every polynomial f in Q[$x_1, x_2, \ldots, x_n$], the largest term (under $>$) in f that has a nonzero coefficient is called the *head term* of f , denoted by head (f). Let ldcf(f) denote the *leading coefficient* of f , i.e., the coefficient of head (f) in f . Every polynomial f can be written as

$$f = \text{ldcf}(f)\ \text{head}(f) + \hat{f}, \qquad \text{where} \quad \text{head}(f) \succ \text{head}(\hat{f})$$

We write tail(f) for $\hat{f}$. For example, if $f(x, y) = x^3 - y^2$, then head(f) = x^3 and tail(f) = $-y^2$ under the total degree ordering $\succ$d; under the purely lexicographic ordering with $y \succ_i x$, we have head (f) = y^2 and tail(f) = x^3 .

A polynomial f (i.e., the equation $f = 0$) is viewed as a rewrite rule

$$\text{ldcf}(f)\ \text{head}(f) \rightarrow -\text{tail}(f)$$

Let f and g be two polynomials; suppose g has a term t with a nonzero coefficient that is a multiple of head(f) ; that is,

$$g = at + \hat{g}, \qquad \text{where} \quad a \in \mathbb{Q} \qquad \text{and} \qquad t = t'\ \text{head}(f)$$

for some term t'. Then g is said to be *reducible* with respect to f, written as a *reduction by* $\rightarrow f$,

$$g \xrightarrow{f} h$$

where

$$h = g - \hat{a}t'f = -\hat{a}t' \operatorname{tail}(f) + \hat{g} \quad \text{and} \quad \hat{a} \operatorname{ldcf}(f) = a$$

The polynomial g is said to be *reducible* with respect to a set (or basis) of polynomials $F = \{f_1, f_2, \ldots, f_r\}$ if it is reducible with respect to one or more polynomials in f; otherwise, we say that g is reduced or g is a *normal form* with respect to F.

Given a polynomial g and a basis $F = \{f_1, f_2, \ldots, f_r\}$, through a *finite sequence of reductions*

$$g = g_1 \xrightarrow{F} g_2 \xrightarrow{F} g_3 \ldots \xrightarrow{F} g_s$$

such that g_s cannot be reduced further, a normal form g_s of g with respect to F can be computed. Because of the admissible ordering on terms used to choose head terms of polynomials, any sequence of reductions must terminate. Further, for every g_i in the above reduction sequence, $g_i - g \in (f_1, f_2, \ldots, f_r)$.

For example, let $F = \{f_1, f_2, f_3\}$, where

$$\begin{aligned} f_1 &= x_1^2 x_2 - 2x_2 x_3 + 1 \\ f_2 &= x_1 x_2^2 - x_3^2 + 2x_1 \\ f_3 &= x_2^2 x_3 - x_1^2 + 5 \end{aligned}$$

and

$$g = 3x_1^2 x_2^2 + x_3^1 - 1$$

Under $\succ$d, we have head $(f_1) = x^2{}_1\, x_2$, head $(f_2) = x_1\, x_2{}^2$, head $(f_3) = x_2{}^2\, x_3$, and g is reducible with respect to f. One possible reduction sequence is

$$\begin{aligned} g &= g_1 \xrightarrow{f_1} g_2 \\ &= 6x_2^2 x_3 - 3x_2 + x_1^3 - 1 \xrightarrow{f_3} g_3 = 6x_1^2 - 3x_2 + x_1^3 - 31 \end{aligned}$$

and g_3 is a normal form with respect to F. It is possible to reduce g in a another way that leads to a different normal form. For example,

$$g = g_1 \xrightarrow{f_2} g_2' = 3x_1 x_3^2 + x_1^3 - 6x_1^2 - 1$$

and the normal form g'_2 is different from g_3.

Gröbner Basis Algorithm.

Definition 10. A finite set of polynomials $G \in Q; [x_1, x_2, \ldots, x_n]$ is called a Gröbner basis for the idea (G) it generates if and only if every polynomial in $Q;x_1, x_2, \ldots, x_n]$ has a unique normal form with respect to G .

In other words, the reduction relation defined by a Gröbner basis is canonical (confluent and terminating). Buchberger (52,54) showed that every ideal in $Q; [x_1, x_2, \ldots, x_n]$ has a Gröbner basis. He also designed an algorithm to construct a Gröbner basis for any ideal I in $Q;x_1, x_2, \ldots, x_n]$ starting from an arbitrary basis for I .

Much like the superpositions and critical pairs discussed for first-order equations in an earlier section on equational inference, head terms of polynomials can be analyzed to determine whether a given basis is a Gröbner basis. For the above example, the reason for two different normal forms (g_3 and g'_2) for g with respect to f is that the monomial $3x^2{}_1\, x^2{}_2$ in g can be reduced by two different polynomials in the basis F in different ways; so head(g) was a common multiple of head(f_1) and head (f_2) .

Buchberger proposed a completion procedure to compute a Gröbner basis by augmenting the basis f by the polynomial $g_3 - g'_2$ [the augmented basis still generates the same ideal since $g_3 - g'_2 \in (f\ )$], much like the completion procedure for equations discussed in the first section. The polynomial $g_3 - g_2{}'$ corresponds to the equation between two different normal forms of a critical pair. Much like a critical pair, an s-polynomial of two polynomials f_1, f_2 is defined as follows. Let

$$m = \mathrm{lcm}(\mathrm{head}(f_1), \mathrm{head}(f_2)) = m_1 \quad \mathrm{head}(f_1) = m_2 \quad \mathrm{head}(f_2)$$

where m_1, m_2 are terms; m plays the role of a superposition. Define

$$s\text{-poly}(f_1, f_2) = m_2\, \mathrm{ldcf}(f_2)\, f_1 - m_1\, \mathrm{ldcf}(f_1)\, f_2$$

Given a basis F for an ideal I and an admissible term ordering $>$, the following algorithm returns a Gröbner basis for I for the term ordering $>$:

$$f(x, y) = x^3 - y^2, \qquad g(x, y) = 2x^2 + (y - 1)^2 - 2$$

In the above, $NF_{\mathrm{G}}(f)$ stands for *any* normal form of f with respect to the basis G . Unlike completion for equational theories, the above procedure always terminates, since by Dickson's lemma there are only finitely many noncomparable terms that can serve as the leading terms of polynomials in a Gröbner basis. For proofs of termination and correctness of the algorithm, the reader is referred to Refs. 23 and 25. The above algorithm does not use any heuristics or optimizations. Most Gröbner basis implementations use several modifications to Buchberger's algorithm in order to speed up the computations.

Examples. Consider the ideal I generated by

$$G_1 = \{y^2 + 2x^2 - 2y - 1, x^3 + 2x^2 - 2y - 1\}$$

(f defines a cusp and g defines an ellipse). Then

$$G_2 = \{g_1(x) = x^6 + 4x^5 + 4x^4 - 6x^3 - 4x^2 + 1$$
$$g_2(x, y) = y + \frac{1}{2}(-x^3 - 2x^2 + 1)\}$$

is a Gröbner basis for I under the degree ordering with $y > x$,

$$G_3 = \{h_1(y) = y^6 - 6y^5 + 17y^4 + 4y^3 - 9y^2 - 6y - 1$$
$$h_2(x, y) = x + \tfrac{1}{4}(2y^5 - 13y^4 + 40y^3 - 10y^2 - 18y - 5)\}$$

is a Gröbner basis for I under the lexicographic ordering with $y \succ x$, and

$G := F;$
$B := \{\langle f_i, f_j\rangle | f_i, f_j \in F\};$
while B is nonempty do
$h := NF_G(s - \text{poly}(f_i, f_j));$ %for some $\langle f_i, f_j\rangle \in B$
if $h \neq 0$ then
$G := G \cup \{h\};$
$B := B \cup \{\langle h, g\rangle | g \in G\};$
$fi;$
$B := B - 0\{\langle f_i, f_j\rangle\};$
od;

is a Gröbner basis for I under the lexicographic ordering with $x \sim x$.

These examples illustrate the fact that, in general, an ideal has different Gröbner bases for different term orderings. For the same term ordering, a *reduced* Gröbner bases is *unique* for an ideal; for every g in a reduced Gröbner basis G, $N\$\$_{G'}(g) = g$ where $G' = G - \{g\}$; i.e., each polynomial in G is reduced with respect to all the other polynomials in G .

Finding Common Solutions: Lexicographic Gröbner Bases. In the Gröbner basis G_2 for the above example, there is a polynomial $g_1(x)$ that depends only on x and one that depends on both x and y (in general, there can be several polynomials in a Gröbner basis that depend on x, y).

Given a reduced Gröbner basis $G = \{g_1, \ldots, g_k\}$ and an admissible term ordering $>$, G can be partitioned based on the variables appearing in the polynomials. If G includes a single polynomial in x_1, a finite set of polynomials in x_1, x_2, a finite set of polynomials in x_1, x_2, x_3, and so on, then variables are said to be *separated* in the Gröbner basis G . For example, variables are separated in Gröbner bases G_2 and G_3, whereas they are not separated in G_1 . From a basis in which variables are separated, a triangular form can be extracted by picking the least-degree polynomial for every variable in the set of polynomials introducing that variable. (It is possible that some of the variables get skipped in a separated basis.) It was observed by Trinks that such a separation of variables exists in Gröbner bases computed using lexicographic term orderings (23).

The triangular form extracted from a Gröbner basis of I in which variables are separated can be used to compute all the common zeros of I . We first find all the roots of the univariate polynomial introducing x_1 . These give the x_1-coordinates of the common zeros of the ideal I . For each such root α, we can find the common roots of $g_2(\alpha, x_2)$, the lowest-degree polynomial in x_2 that may have both x_1, x_2 ; this gives the x_2-coordinates of the corresponding common zeros of I . In this way, all the coordinates of all the common zeros can be computed. The product of the degrees of the polynomials in triangular form used to compute these coordinates also gives the total number of common zeros (including their multiplicities) of a zero-dimensional ideal I, where an ideal is zero-dimensional if and only if Zero(I) is finite.

If a variable gets skipped in a triangular form (meaning that there is no polynomial in a Gröbner basis introducing it), this implies that I is not zero-dimensional, and that I has infinitely many common zeros. As

the reader might have guessed, a Gröbner basis in which variables are separated can be used to determine the dimension of an ideal.

In principle, any system of polynomial equations can be solved using a lexicographic Gröbner basis for the ideal generated by the given polynomials. However, Gröbner bases, particularly lexicographic Gröbner bases, are hard to compute. For zero-dimensional ideals, a basis conversion has been proposed that can be used to convert a Gröbner basis computed using one admissible ordering to a Gröbner basis with respect to another admissible ordering (25). In particular, a Gröbner basis with respect to a lexicographic term ordering can be computed from a Gröbner basis with respect to a total degree ordering, which is easier to compute.

If a set of polynomials does not have a common zero (i.e., its ideal is the whole ring), then it is easy to see that a Gröbner basis of such a set of polynomials includes 1 no matter what term ordering is used. Gröbner basis computations can thus be used to check for the consistency of a system of nonlinear polynomial equations.

Theorem 11. A set of polynomials in $Q;[x_1, \ldots, x_n]$ has no common zero in C if and only if their reduced Gröbner basis with respect to any admissible term ordering is $\{1\}$.

Elimination. A Gröbner basis algorithm can also be used to eliminate variables as well as to compute the exact resultant. Variables to be eliminated are made higher than the other variables in the term ordering, just as in characteristic set computation. A Gröbner basis G of a set S of polynomials is then computed using the lexicographic term ordering from S . If there are $k + 1$ polynomials from which k variables have to be eliminated, then the smallest polynomial g in the Gröbner basis G is the exact resultant, as it can be shown that this polynomial g is the unique generator of the elimination ideal (contraction) of I in the subring of polynomials in the variables that are not being eliminated.

In Table 1 above, methods for computing multivariate resultants are contrasted with the Gröbner basis method for computing resultants on a variety of problems from different application domains. Even though the timings for the Gröbner basis approach do not compare well with resultant methods, the Gröbner basis method has an edge over the resultant methods in that it computes the resultant exactly. In contrast, resultant methods produce extraneous factors; identifying extraneous factors can require considerable effort.

The results reported in Table 1 for the Gröbner basis method were obtained using *block* ordering instead of lexicographic ordering. In a block ordering, variables are partitioned into blocks, and blocks are lexicographically ordered. Terms are compared considering variables block by block. Starting with the biggest block, the degree of a term in the variables in that block is compared against the degree of another term in these variables. Only if these degrees are the same is the next block considered, and so on. The lexicographic ordering and total degree ordering are particular cases of block orderings; each variable constitutes a block in the former, and all variables together constitute a single block in the latter. For elimination, variables being eliminated together as a block are made lexicographically bigger than the parameters (variables not being eliminated) considered together as another block. Gröbner basis computation is typically faster using a block ordering than using a lexicographic ordering, but slower than using a total degree ordering.

Theorem Proving Using Gröbner Basis Computations. A refutational approach to theorem proving has been developed exploiting the property that a Gröbner basis algorithm can be used to check whether a set of polynomial equations is inconsistent. In Ref. 55, we discussed a refutational method for geometry theorem proving. A geometry theorem proving problem (that does not involve an order relation) is formulated as the problem of checking inconsistency of a set of polynomial equations. This approach can also be used to discover missing degenerate cases as well as missing hypotheses in an incompletely stated geometry theorem. Many examples proved using Geometer, including nontrivial problems such as the butterfly theorem and Pappus's theorem, are discussed in Ref. 55.

An approach based on Gröbner basis computations has also been proposed for first-order theorem proving and implemented in our theorem prover *RRL*. For propositional calculus, formulae are translated into polynomial equations over the Boolean ring generated by propositional variables. Deciding whether a formula is a theorem is done by checking whether the corresponding polynomial equations have a solution over $\{0, 1\}$. This idea is then generalized to work on first-order rings and first-order polynomials. Details can be found in Ref. 16.

Complexity Issues and Implementation. In general, Gröbner bases are hard to compute. It was shown by Mayr and Meyer (56) that the problem of testing for ideal membership is *exponential space complete*. Their construction shows that for ideals given by bases of the form $(m_{1,1} - m_{1,2}, m_{2,1} - m_{2,2}, \ldots, m_{k,1} - m_{k,2})$ where $m_{i,j}$ is a monomial of degree at most d, Gröbner basis computation will encounter polynomials of degree as high as $O(d\ 2^{n})$, double-exponential in n, the number of variables. This is an inherent difficulty and cannot be avoided if one expects to handle all possible ideals. While double-exponential degree explosions are not observed in all problems of interest, high-degree polynomials are frequently encountered in practice.

The second problem comes from the extremely large size of the coefficients of polynomials that are generated during Gröbner basis computations. While intermediate expression swell is a common problem in computer algebra, it seems to be particularly acute in this context.

Despite these difficulties, highly nontrivial Gröbner bases computations have been performed. If the coefficients belong to a finite field (typically Z_p, where p is a word-sized prime), much larger computations are possible. Macaulay, CoCoA, and Singular are specialized computer algebra systems built for performing large computations in algebraic geometry and commutative algebra. Most general computer algebra systems (such as Maple, Macsyma, Mathematica, Reduce) provide the basic Gröbner basis functions. An implementation of the Gröbner basis algorithm also exists in GeoMeter (50,51), a programming environment for geometric modeling and algebraic reasoning. This implementation has been used for proving nontrivial plane geometry theorems.

Acknowledgment

The author would like to thank his colleagues and coauthors of articles on which most of the material in this paper is based—Lakshman Y. N., P. Narendran, T. Saxena, G. Sivakumar, M. Subramaniam, L. Yang, and H. Zhang. The author also thanks S. Lee for editorial comments.

This work was supported in part by NSF grants CCR-9622860, CCR-9712366, CCR-9712396, CCR-9996150, CCR-9996144, and CDA-9503064.

Footnotes

1. Using a sequence of positions of arguments, each subterm in a term can be uniquely identified by its position. For instance, in $(u + -(u)) + -(-(u))$, the position Λ (the empty sequence) identifies the whole term; 1 identifies the subterm $u + -(u)$, the first argument of the top level symbol $+$; 1.1 and 1.2 identify, respectively, u and $-(u)$, the first and second arguments of $+$ in $u + -(u)$; similarly, 2 identifies $-(-(u))$, the second argument of the top level symbol $+$, 2.1 identifies $-(u)$, and 2.1.1 identifies the subterm u which is the argument of $-$ in the subterm $-(u)$, the argument of $-$ in $-(-(u))$.
2. For certain equations such as $x + y = y + x$, the commutativity law, it is not even possible to transform an equation into a terminating rule, no matter which side is considered more complex. Such equations are handled semantically as discussed in a later subsection.
3. This case can be handled by one of the heuristics discussed earlier, including unfailing completion.
4. There are other ways to define rewriting modulo a set of equational axioms. For details, a reader may consult 9. The above definition matches the implementation of *AC* rewriting in our theorem prover Rewrite Rule Laboratory (*RRL*).
5. H. Robbins in 1930 conjectured that three equations defining commutativity and associativity of a binary function symbol $+$, and a third equation $-(-(x + y) + -(x + -(y))) = x$, with a unary function symbol $-$, are a basis for the variety of Boolean algebras.

BIBLIOGRAPHY

1. D. Kapur H. Zhang, An overview of Rewrite Rule Laboratory (RRL), *J. Comput. Math. Appl.*, **29** (2): 91–114, 1995.
2. D. Kapur, M. Subramaniam, Using an induction prover for verifying arithmetic circuits, *J. Softw. Tools Technol. Transfer*, 1999, to appear.
3. D. E. Knuth, P. B. Bendix, Simple word problems in universal algebras, in J. Leech (ed.), *Computational Problems in Abstract Algebras*, Oxford, England: Pergamon, 1970, pp. 263–297.
4. N. Dershowitz, Termination of rewriting, *J. Symb. Comput.*, Vol. 3, 1987, pp. 69–115.
5. L. Bachmair, *Canonical Equational Proofs*, Basel: Birkhaeuser, 1991.
6. D. Kapur, G. Sivakumar, Architecture and experiments with *RRL*, a Rewrite Rule Laboratory, in J. V. Gultag, D. Kapur, and D. R. Musser (eds.), *Proceedings of the NSF Workshop on the Rewrite Rule Laboratory*, Tech. No. GE-84GEN008 Schenectady, NY: General Electric Corporate Research and Development, 1984, pp. 33–56.
7. L. Bachmair, N. Dershowitz, D. A. Plaisted, Completion without failure, in H. Ait-Kaci and M. Nivat (eds.), *Resolution of Equations in Algebraic Structures*, Vol. 2, Boston, MA: Cambridge Press, 1989, pp. 1–30.
8. D. Kapur P. Narendran, A finite Thue system with decidable word problem and without finite equivalent canonical system, *Theor. Comput. Sci.*, **35**: 337–344, 1985.
9. J.-P. Jouannaud, H. Kirchner, Completion of a set of rules modulo a set of equations, *SIAM J. Comput.* **15**: 349–391, 1997.
10. G. E. Peterson, M. E. Stickel, Complete set of reductions for some equational theories, *J. ACM*, **28**: 223–264, 1981.
11. D. Kapur, G. Sivakumar, Proving associative–commutative termination using RPO-compatible orderings, to appear in *Invited Pap., Proc. 1st Order Theorem Proving*, 1999.
12. D. Kapur H. Zhang, A case study of the completion procedure. Proving ring commutativity problems, in J.L. Lassez and G. Plotkin (eds.), *Computational Logic: Essays in Honor of Alan Robinson*, Cambridge, MA: MIT Press, 1991, pp. 360–394.
13. W. McCune, Solution of the Robbins problem, *J. Autom. Reasoning*, **19** (3): 263–276, 1997.
14. R. S. Boyer, J. S. Moore, *A Computational Logic Handbook*, Orlando, FL: Academic Press, 1988, pp. 162–181.
15. H. Zhang, D. Kapur, M. S. Krishnamoorthy, A mechanizable induction principle for equational specifications *Proc. 9th Int. Conf. Autom. Deduction (CADE-9)*, Argonne, IL, 1988, pp. 162–181.
16. D. Kapur, P. Narendran, An equational approach to theorem proving in first-order predicate calculus, *Proc. 7th Int. Jt. Conf. Artif. Intell. (IJCAI-85)*, 1985, pp. 1146–1153.
17. M. S. Paterson M. Wegman, Linear unification, *J. Comput. Syst. Sci.*, **16**: 158–167, 1978.
18. D. Kapur P. Narendran, Complexity of associative–commutative unification check and related problems. *J. Autom. Reasoning* **9** (2): 261–288, 1992.
19. D. Kapur, P. Narendran, Double-exponential complexity of computing a complete set of AC-unifiers, *Proc. Logic Comput. Sci. (LICS)*, Santa Cruz, CA, 1992, pp. 11–21.
20. C. Prehofer, Solving higher order equations: From logic to programming, Tech. Rep. 19508, Munich, Technische Universität, 1995.
21. M. Hanus, The integration of functions into logic programming: >From theory to practice, *J. Logic Program.*, **19–20**: 583–628, 1994.
22. D. Kapur, P. Narendran, F. Otto, On ground confluence of term rewriting systems, *Inf. Comput.*, Vol. 86, San Diego, CA: Academic Press, May 1990, pp. 14–31.
23. T. Becker V. Weispfenning H. Kredel, *Gröbner Bases: A Computational Approach to Commutative Algebra*, Berlin: Springer-Verlag, 1993.
24. S.-C. Chou, *Mechanical Geometry Theorem Proving*, Dordrecht, The Netherlands: Reidel Publ., 1988.
25. D. Cox, J. Little, D. O'Shea, *Ideals, Varieties, and Algorithms*, Berlin: Springer-Verlag, 1992.
26. C. Hoffman, *Geometric and Solid Modeling: An Introduction*, San Mateo, CA: Morgan Kaufmann, 1989.
27. A. P. Morgan, *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems*, Englewood Cliffs, NJ: Prentice-Hall, 1987.
28. D. Kapur, J. L. Mundy (eds.), *Geometric Reasoning*, Cambridge, MA: MIT Press, 1989.
29. B. Donald, D. Kapur, J. L. Mundy (eds.), *Symbolic and Numeric Methods in Artificial Intelligence*, London: Academic Press, 1992.

30. I. M. Gelfand, M. M. Kapranov, A. V. Zelevinsky, *Discriminants, Resultants and Multidimensional Determinants*, Boston: Birkhaeuser, 1994.
31. D. Kapur, T. Saxena, L. Yang, Algebraic and geometric reasoning using Dixon resultants, *Proc. Int. Symp. Symb. Algebraic Comput. (ISSAC-94)*, Oxford, England, pp. 99–107, 1994.
32. D. Kapur T. Saxena, Comparison of various multivariate resultant formulations, *Proc. Int. Symp. Symb. Algebraic Comput. (ISSAC-95)*, Montreal, 1995, pp. 187–194.
33. B. L. van der Waerden, *Algebra*, Vols. 1 and 2, New York: Frederick Ungar Publ. Co., 1950, 1970.
34. S. S. Abhyankar, Historical ramblings in algebraic geometry and related algebra, *Am. Math. Mon.*, **83** (6): 409–448, 1976.
35. F. S. Macaulay, *The Algebraic Theory of Modular Systems*, Cambridge Tracts in Math. Math. Phy. Vol. 19, 1916.
36. D. Y. Grigoryev and A. L. Chistov, Sub-exponential time solving of systems of algebraic equations, LOMI Preprints E-9-83 and E-10-83, Leningrad, 1983.
37. J. Canny, Generalized characteristic polynomials, *J. Symb. Comput.*, **9**: 241–250, 1990.
38. D. Kapur, T. Saxena, Extraneous factors in the Dixon resultant formulation, *Proc. Int. Symp. Symb. Algebraic Comput. (ISSAC-97)*, Maui, HI, 1997, pp. 141–148.
39. A. L. Dixon, The eliminant of three quantics in two independent variables, *Proc. London Math. Soc.*, **6**: 468–478, 1908.
40. D. Kapur, T. Saxena, Sparsity considerations in Dixon resultants, *Proc. ACM Symp. Theory Comput. (STOC)*, Philadelphia, pp. 184–191, 1996.
41. T. Saxena, Efficient Variable Elimination Using Resultants, Ph.D. Thesis, Department of Computer Science, State University of New York, Albany, NY, 1996.
42. J. F. Ritt, *Differential Algebra*, New York: AMS Colloquium Publications, 1950.
43. W. Wu, On the decision problem and the mechanization of theorem proving in elementary geometry, in W. W. Bledsoe and D. W. Loveland (eds.), *Theorem Proving: After 25 Years, Contemporary Mathematics*, Vol. 29, Providence, RI: American Mathematical Society, 1984, pp. 213–234.
44. W. Wu, Basic principles of mechanical theorem proving in geometries, *J. Autom. Reasoning*, **2**: 221–252, 1986.
45. W. Wu, On zeros of algebraic equations—an application of Ritt's principle, *Kexue Tongbao*, **31** (1): 1–5, 1986.
46. D. Kapur, Y. N. Lakshman, Elimination methods: An introduction, in B. Donald, D. Kapur, and J. Mundy (eds.), *Symbolic and Numerical Computation for Artificial Intelligence*, San Diego, CA: Academic Press, 1992, pp. 45–89.
47. D. Kapur, *Algorithmic Elimination Methods*, Tutorial Notes for *ISSAC-95*, Montreal, 1995.
48. D. Kapur, H. Wan, Refutational proofs of geometry theorems via characteristic set computation, *Proc. Int. Symp. Symb. Algebraic Comput. (ISSAC-90)*, Japan, 1990, pp. 277–284.
49. G. Gallo, B. Mishra, *Efficient Algorithms and Bounds for Ritt–Wu Characteristic Sets*, Tech. Rep. No. 478, New York: Department of Computer Science, New York University, 1989.
50. D. Cyrluk, R. Harris, D. Kapur, GEOMETER: A theorem prover for algebraic geometry, *Proc. 9th Int. Conf. Autom. Deduction (CADE-9)*, Argonne, IL, 1988.
51. C. I. Connolly, *et al.* GeoMeter: A system for modeling and algebraic manipulation, *Proc. DARPA Workshop Image Understanding*, pp. 797–804, 1989.
52. B. Buchberger, Gröbner bases: An algorithmic method in polynomial ideal theory, in N.K. Bose (ed.), *Multidimensional Systems Theory*, Dordrecht, The Netherlands: Reidel Publ., 1985, pp. 184–232.
53. A. Kandri-Rody, D. Kapur, An algorithm for computing the Gröbner basis of a polynomial ideal over an Euclidean ring, *J. Symb. Comput.*, **6**: 37–57, 1988.
54. B. Buchberger, Applications of Gröbner bases in non-linear computational geometry, in D. Kapur and J. Mundy (eds.), *Geo-metric Reasoning*, Cambridge, MA: MIT Press, 1989, pp. 415– 447.
55. D. Kapur, A refutational approach to theorem proving in geometry, *Artif. Intell. J.*, **37** (1–3): 61–93, 1988.
56. E. Mayr, A. Meyer, The complexity of word problem for commutative semigroups and polynomial ideals, *Adv. Math.*, **46**: 305–329, 1982.

DEEPAK KAPUR
University of New Mexico

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Fourier Analysis
Standard Article

Xin Li[1]
[1]University of Central Florida, Orlando, FL

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2416
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (223K)

Abstract

The sections in this article are

Fourier Series

Signal Sampling

Numerical Computation

Wavelets Approach

Conclusions

| | |

Copyright © 1999-2008 All Rights Reserved.

FOURIER ANALYSIS

Fourier analysis is a collection of related techniques for representing general functions as linear combinations of simple functions or functions with certain special properties. In the classical theory, these simple functions (called *basis functions*) are sinusoids (sine or cosine functions). The modern theory uses many other functions as the basis functions. Every basis function carries certain characteristics that can be used to describe the functions of interest; it plays the role of a building block for the complicated structures of the functions we want to study. The choice of a particular set of basis functions reflects how much we know and what we want to find out about the functions we want to analyze. In this article, we will restrict our discussion (except for the last section about wavelets) to the classical theory of Fourier analysis. For a broader point of view, see FOURIER TRANSFORM.

In applications, Fourier analysis is used either simply as an efficient computational algorithm or as a tool for analyzing the properties of the signals, functions of time, or space variables at hand. (In this article, we will use the terms signal and function interchangeably.)

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

A very important component of modern technology is the processing of signals of various forms in order to extract the most significant characteristics carried in the signals. In practice, most of the signals in their raw format, are given as functions of the time or space variables, so we also call the domain of the signal the *time* (or *space*) *domain*. This time- or space-domain representation of a signal is not always the best for most applications. In many cases, the most distinguished information is hidden in the *frequency content* or *frequency spectrum* of a signal. Fourier analysis is used to accomplish the representation of signals in the frequency domain.

Fourier analysis allows us to calculate the "weights" (amplitudes) of the different frequency sinusoids which make up the signal. Given a signal, we can view the process of analyzing the signal by Fourier analysis as one of transforming the original signal into another form that reveals its properties (in the frequency domain) that cannot be directly seen in the original form of the signal.

The most useful tools in Fourier analysis are the following three types of transforms: Fourier series, discrete Fourier transform, and (continuous) Fourier transform. With each transform there is associated an inverse transform that recovers (in a sense to be discussed later) the original signal from the transformed one. The process of calculating a transform is also referred to as Fourier spectral analysis; the process of recovering the original function from its transform by using the inverse transform is called Fourier synthesis.

The wide use of Fourier analysis in engineering must be credited to the existence of the fast Fourier transform (FFT), a fast computer implementation of the discrete Fourier transform. Areas where Fourier analysis (via FFT) has been successfully applied, include applied mechanics, biomedical engineering, computer vision,numerical methods, signal and image processing, and sonics and accoustics.

Fourier analysis is closely related to the sampling of signals. In order to analyze signals using a computer, a continuous time signal must be sampled (at either equally or unequally spaced time intervals). Instead, they are given by a set of sample values. The resulting discrete-time signal is called the *sampled version* of the original continuous-time signal. There are two types of sampling: uniform sampling and nonuniform sampling. We only discuss the uniform sampling in this article. How often must a signal be sampled in order that all the frequencies present should be detected? This is discussed with sampling theorems.

Recently, a new set of tools under the generic name *wavelets analysis* has found various applications. Wavelets analysis can be view as an enhancement of the classical Fourier analysis. In wavelets analysis, the basis functions are not sinusoids but functions with zero average and other additional properties. These basis functions are localized in both time and frequency domains.

FOURIER SERIES

History

The history of Fourier analysis can be dated back at least to the year 1747 when Jean Le Rond d'Alembert (1717–1783) derived the "wave equation" which governs the vibration of a string. Other mathematicians involved in the study of Fourier analysis include Leonard Euler (1707–1783), Daniel Bernoulli (1700–1782), and Joseph Louis Lagrange (1736–1813). An enormous and important step was made by Jean Baptiste Joseph Fourier (1768–1830) when he took up the study of heat conduction. He used sines and cosines in his study of the flow of heat. He submitted a basic paper on heat conduction to the Academy of Sciences of Paris in 1807 in which he announced his belief in the possibility of representing every function $f(x)$ on the interval (a, b) by a trigonometric series of the form (with $P = b - a$)

$$\frac{1}{2}A_0 + \sum_{n=1}^{\infty}\left(A_n \cos\frac{2\pi nx}{P} + B_n \sin\frac{2\pi nx}{P}\right) \tag{1}$$

where

$$A_n = \frac{2}{P}\int_a^b f(x)\cos\frac{2\pi nx}{P}\,dx \qquad (n = 0, 1, 2, \ldots) \tag{2}$$

$$B_n = \frac{2}{P}\int_a^b f(x)\sin\frac{2\pi nx}{P}\,dx \qquad (n = 1, 2, 3, \ldots) \tag{3}$$

Because of its lack of rigor, the paper was rejected by a committee consisting of Lagrange, Laplace, and Legendre. Fourier then revised the paper and resubmitted it in 1811. The paper was judged again by the three aforementioned mathematicians as well as others. Showing great insight, Academy awarded Fourier the Grand Prize of the Academy despite the defects in his reasoning. This 1811 paper was not published in its original form in the *Mémoires* of the Academy until 1824 when Fourier became the secretary of the Academy. (It is worthwhile to point out that there were good reasons that Fourier's theorem was criticized by his contemporaries: At that time, the modern concepts of function and limit were not available.)

As a result of Fourier's work, the sequences $\{A_n\}_{n=0}^{\infty}$ and $\{B_n\}_{n=0}^{\infty}$ defined by Eqs. (2) and (3) are now universally known as the (real) Fourier coefficients of $f(x)$ (though these formulae were known to Euler and Lagrange before Fourier). The term, $A_1 \cos(2\pi \times /P) + \beta_1 \sin(2\pi \times /P)$, is called the principal (spectral) component of the expansion; and the number $\omega_0 = 1/P$ is called the principal (or fundamental) frequency.

Since Fourier coefficients are defined by integrals, the function f must be integrable. In searching for a more general concept of integration (so as to include more functions in Fourier analysis), Bernhardt Riemann (1826–1866) introduced the definition of integral now associated with his name, the Riemann integral. Later, Henri Lebesgue (1875–1941) constructed an even more general integral, the Lebesgue integral.

Because changing the values of a function at finitely many points will not change the value of its integral, we will not distinguish two functions if they are the same except at finitely many points.

The Complex Form of Fourier Series

Given a function $f(x)$ on (a, b), to calculate its Fourier series of the form shown in Eq. (1) we have to use two equations [Eqs. (2) and (3)] to obtain the coefficients A_n and B_n. This is why we sometimes want to use an alternative form of Fourier

series, the complex form. To rewrite Eq. (1), we use Euler's identity (as usual, with $j = \sqrt{-1}$)

$$e^{j\phi} = \cos\phi + j\sin\phi$$

Then the trigonometric series in Eq. (1) can be put in a formally equivalent form,

$$\sum_{n=-\infty}^{\infty} c_n e^{j2\pi nx/P} \tag{4}$$

in which, on writing $B_0 = 0$, we have

$$c_n = \frac{1}{2}(A_n + B_n), \quad c_{-n} = \frac{1}{2}(A_n - B_n), \qquad n = 0, 1, 2, \ldots$$

From Eqs. (2) and (3), we can derive

$$c_n = \frac{1}{P}\int_a^b f(x)e^{-j2\pi nx/P}\,dx, \qquad n = 0, \pm 1, \pm 2, \ldots \tag{5}$$

The numbers $\{c_n\}_{n=-\infty}^{\infty}$ are called the "complex" Fourier coefficients of $f(x)$. The two series in Eqs. (1) and (4) are referred to as the real and complex Fourier series of $f(x)$, respectively.

The Orthogonality Relations

Before we explore Fourier series further, it is important to point out the facts that provided the heuristic basis for the formulae in Eqs. (2), (3), and (5) for the Fourier coefficients. These facts, which can be proved by simple and straightforward calculations, are expressed in the following orthogonality relations. In the real form, we have

$$\frac{1}{P}\int_a^b \cos\frac{2\pi mx}{P}\cos\frac{2\pi nx}{P}\,dx = \begin{cases} 0 & \text{for } m \neq n \\ \frac{1}{2} & \text{for } m = n \neq 0 \\ 1 & \text{for } m = n = 0 \end{cases} \tag{6}$$

$$\frac{1}{P}\int_a^b \sin\frac{2\pi mx}{P}\sin\frac{2\pi nx}{P}\,dx = \begin{cases} 0 & \text{for } m \neq n \\ \frac{1}{2} & \text{for } m = n \neq 0 \\ 1 & \text{for } m = n = 0 \end{cases} \tag{7}$$

$$\frac{1}{P}\int_a^b \sin\frac{2\pi mx}{P}\cos\frac{2\pi nx}{P}\,dx = 0 \tag{8}$$

and in the complex form, we have

$$\frac{1}{P}\int_a^b e^{j2\pi mx/P}\,e^{-j2\pi nx/P}\,dx = \begin{cases} 0 & \text{for } m \neq n \\ 1 & \text{for } m = n \end{cases} \tag{9}$$

where m and n are integers, and the interval of integration $[a, b]$ can be replaced by any other interval of length P.

Note that to express the orthogonality among trigonometric functions, we need three identities, namely, Eqs. (6), (7), and (8); but to do the same among exponential functions, we need only one identity, Eq. (9). In general, it is more convenient to compute the complex Fourier series first and then change it to the "real" form in sine and cosine functions. From the definition, we can easily verify that if $f(x)$ is real-valued, then its complex Fourier series can always be put into a real-valued trigonometric series. We will illustrate this in our examples.

Examples of Fourier Series

Example 1. Find the Fourier series of $f(x) = \pi - x$ on interval $(0, 2\pi)$.

SOLUTION. We use Eq. (5) to find the complex Fourier coefficients first. For $n = 0$, we have $c_0 = (1/2\pi)\int_0^{2\pi}(\pi - x)\,dx = 0$. For $n \neq 0$, using integration by parts, we have

$$\begin{aligned} c_n &= \frac{1}{2\pi}\int_0^{2\pi}(\pi - x)e^{-jnx}\,dx \\ &= \frac{1}{2\pi}\left\{-\frac{1}{jn}e^{-jnx}(\pi - x)\Big|_0^{2\pi} - \frac{1}{jn}\int_0^{2\pi} e^{-jnx}\,dx\right\} \\ &= \frac{1}{2\pi}\left\{\frac{2\pi}{jn} + \frac{1}{(jn)^2}e^{-jnx}\Big|_0^{2\pi}\right\} = -\frac{j}{n} \end{aligned}$$

Hence, the complex Fourier series of $f(x)$ on $(0, 2\pi)$ is given by

$$\sum_{n=-\infty}^{\infty}{}' -\frac{j}{n}e^{jnx} \tag{10}$$

where the prime on the sum is used to indicate that the $n = 0$ term is omitted.

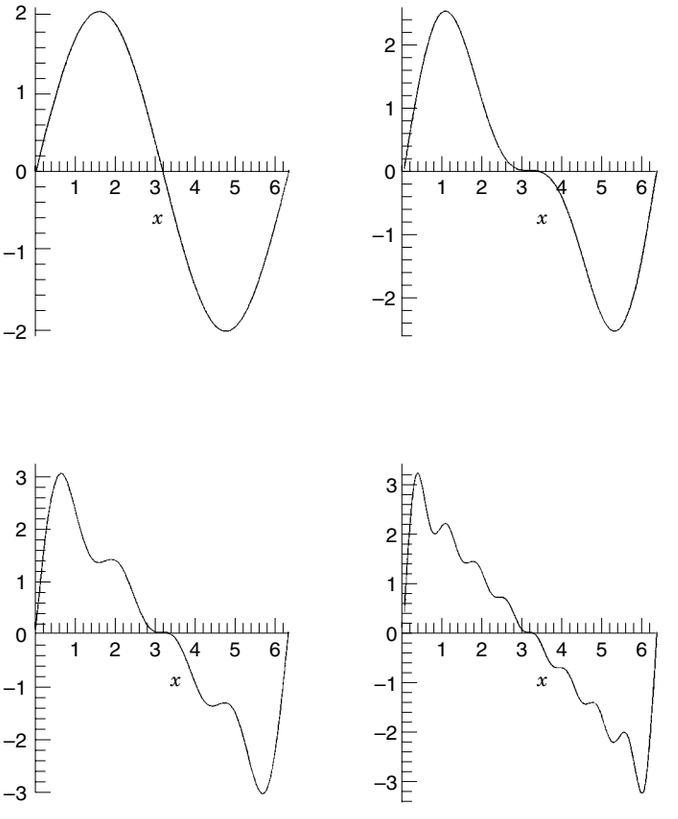


Figure 1. $S_1(x)$, $S_2(x)$, $S_4(x)$, and $S_8(x)$.

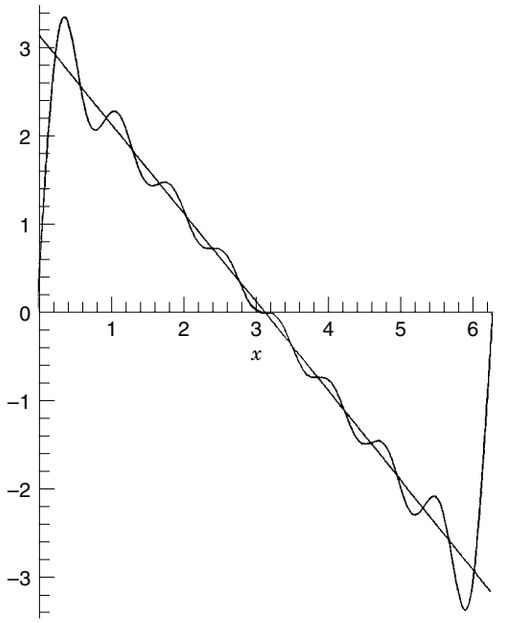


Figure 2. $f(x) = \pi - x$ and $S_8(x)$ on $(0, 2\pi)$.

Grouping $-(j/n)e^{jnx}$ and $-(j/-n)e^{-jnx}$, we obtain $(2/n)\sin nx$, so we can write Eq. (10) in the real form:

$$\sum_{n=1}^{\infty} \frac{2}{n} \sin nx$$

In Fig. 1, we show the graphs of the partial sums:

$$S_m(x) = \sum_{n=1}^{m} \frac{2}{n} \sin nx$$

for $m = 1, 2, 4$, and 8. In Fig. 2, we show both $f(x) = \pi - x$ and $S_8(x)$ on the interval $(0, 2\pi)$.

Notice that the graph of $S_8(x)$ is a wavy approximation to the original function $f(x) = \pi - x$ on $(0, 2\pi)$. Outside of the interval, the graph of $S_8(x)$ is approximating the *periodic extension* (*with period* 2π) $f_p(x)$ of $f(x)$ (see Fig. 3).

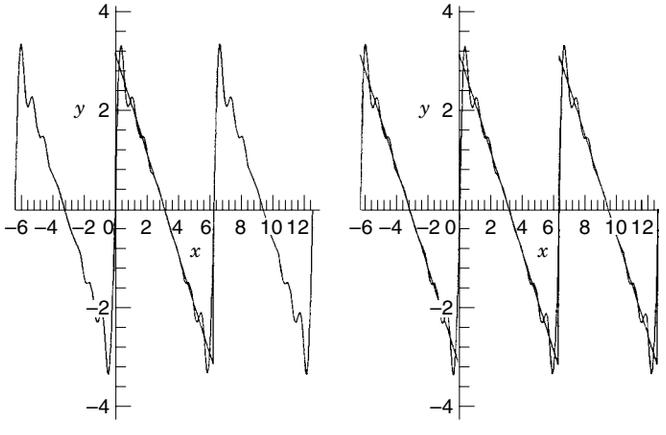


Figure 3. $f(x)$ and $S_8(x)$ on $(-2\pi, 4\pi)$, $f_p(x)$ and $S_8(x)$ on $(-2\pi, 4\pi)$.

Example 2. Find the Fourier series of $f(x)$ defined by

$$f(x) = \begin{cases} 0, & -1 < x < 0 \\ \dfrac{1}{2}, & x = 0 \\ 1, & 0 < x < 1 \end{cases}$$

on the interval $(-1, 1)$.

SOLUTION. The complex Fourier series of $f(x)$ is given by

$$\frac{1}{2} + \sum_{n=-\infty}^{\infty}{}' \frac{j}{2\pi n}[(-1)^n - 1]e^{j\pi nx} \tag{11}$$

We can write Eq. (11) in the following real form:

$$\frac{1}{2} + \sum_{n=1}^{\infty} \frac{1}{\pi n}[1 - (-1)^n] \sin \pi nx$$
$$= \frac{1}{2} + \sum_{n=1}^{\infty} \frac{2}{\pi(2n-1)} \sin \pi(2n-1)x$$

Let

$$S_m(x) = \frac{1}{2} + \sum_{n=1}^{m} \frac{2}{\pi(2n-1)} \sin \pi(2n-1)x$$

denote the partial sum. In Fig. 4, we show the graphs of $S_m(x)$ for $m = 12$, 24, and 36, along with the graph of $f(x)$ and $S_{36}(x)$.

Convergence

Does the Fourier series of a function $f(x)$ converge to $f(x)$? Fourier's assertion that the answer is yes was initially greeted with a great amount of disbelief as we mentioned earlier. In fact, the answer depends on what sense of convergence is understood. Fourier was right, and the answer is always yes provided that things are interpreted suitably. Pointwise convergence is one of the many choices; it is also the first one considered in the study of Fourier series. Because of this, there is a lot of pointwise convergence theorems, although most of them are sufficient conditions. Dirichlet was the first mathematician who carefully studied the validity of pointwise convergence of the Fourier series. We will state two such results that more or less cover most application problems from physics and engineering.

We say that the function $f(x)$ is piecewise continuous on (a, b) if (1) f is continuous on (a, b) except perhaps at finitely many *exceptional* points and (2) at each x_* of the exceptional points, both one-sided limits

$$\lim_{x \to x_*^-} f(x) =: f(x_*-) \text{ and } \lim_{x \to x_*^+} f(x) =: f(x_*+)$$

exist. [At the endpoints, a and b, we assume both $\lim_{x \to a^+} f(x) =: f(a+)$ and $\lim_{x \to b^-} f(x) =: f(b-)$ exist.]

Next, we say that function $f(x)$ is piecewise smooth on (a, b) if both $f(x)$ and its first derivative $f'(x)$ are piecewise continuous on (a, b).

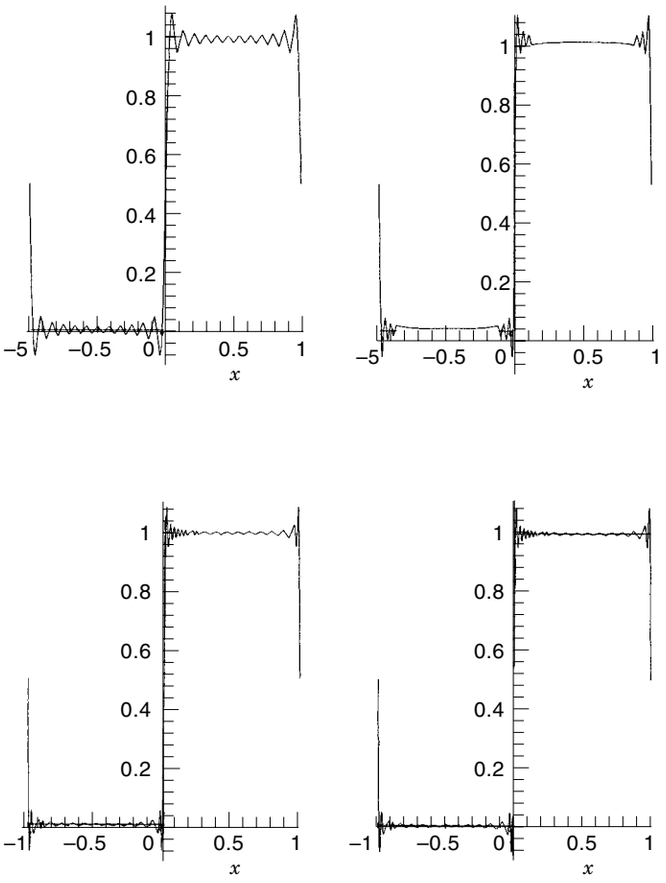


Figure 4. $S_{12}(x)$, $S_{24}(x)$, $S_{36}(x)$, and $f(x)$ and $S_{36}(x)$.

Dirichlet Theorem. If $f(x)$ is piecewise smooth on (a, b), and $S_m(x)$ denotes the mth partial sums of the Fourier series of $f(x)$, then

$$\lim_{m\to\infty} S_m(x) = \begin{cases} \frac{1}{2}[f(x-) + f(x+)], & \text{for } a < x < b \\ \frac{1}{2}[f(a+) + f(b-)], & \text{for } x = a \text{ or } b \end{cases}$$

In particular, $\lim_{m\to\infty} S_m(x) = f(x)$ for every $x \in (a, b)$ at which f is continuous.

The functions in Examples 1 and 2 are both piecewise smooth functions. For example, in Example 1 the Fourier series converges to $f(x)$ at every point in $(0, 2\pi)$ and to 0 at the endpoints 0 and 2π.

Dirichlet's Theorem can be extended to more functions. We say that the function $f(x)$ is of bounded variation on (a, b) if the sums of the form

$$|f(x_1) - f(a)| + |f(x_2) - f(x_1)| + \cdots + |f(b) - f(x_k)|$$

are bounded for all k and for all choices of $(a<)x_1 < x_2 < \cdots < x_k(<b)$.

It is known that a function is of bounded variation if and only if it can be written as the difference of two monotonic (either increasing or decreasing) functions (see Ref. 1).

Dirichlet–Jordan Theorem. If $f(x)$ is of bounded variation on interval (a, b), and if $S_m(x)$ denotes the mth partial sum of its Fourier series, then

$$\lim_{m\to\infty} S_m(x) = \begin{cases} \frac{1}{2}[f(x-) + f(x+)], & \text{for } a < x < b \\ \frac{1}{2}[f(a+) + f(b-)], & \text{for } x = a \text{ or } b \end{cases}$$

It is not hard to show that a piecewise smooth function can be represented as the difference of two increasing functions; so a piecewise smooth function is of bounded variation. Therefore, the Dirichlet–Jordan Theorem is more general than the Dirichlet Theorem.

Limitations of Pointwise Convergence

Although it is undeniably of great intrinsic interest to know that a certain function admits a pointwise representation by its Fourier series, it must be pointed out without delay that, in many situations, simple pointwise convergence is not the appropriate thing to look at.

It has been known since 1876 that the Fourier series of a continuous function may diverge at infinitely many points, and the Fourier series of a integrable function may diverge at all points. For almost a century, whether the Fourier series of a general continuous function is guaranteed to converge at least at some points remained in doubt. An affirmative answer was obtained by L. Carleson in 1966 with a deep theorem asserting that the Fourier series of every square-integrable function must converge to the function at "almost every" point. See the article by Hunt in Ref. 2. Therefore, we have to restrict functions to certain special types in order to achieve the pointwise convergence. Although the theorems presented above are sufficient for many purposes, they do not give the whole picture. See Zygmund's book (1) for more details.

Other Types of Convergence. Here we briefly mention some other types of convergence that may be used when studying the Fourier series. First, there is *uniform convergence,* which is stronger than pointwise convergence. Next, there is *pth power mean convergence,* according to which the series $\sum_{n=-\infty}^{\infty} c_n e^{j2\pi nx/P}$ converges to $f(x)$ on the interval (a, b) (with $P = b - a$) if

$$\lim_{m\to\infty} \int_a^b \left| f(x) - \sum_{n=-m}^{m} c_n e^{j2\pi nx/P} \right|^p dx = 0$$

The case when $p = 2$ is especially simple and useful. Finally, there is *distributional convergence* which is defined as follows: The series $\sum_{n=-\infty}^{\infty} c_n e^{j2\pi nx/P}$ is said to distributionally converge to $f(x)$ on the interval (a, b) $(P = b - a)$ if

$$\lim_{m\to\infty} \int_a^b u(x) \sum_{n=-m}^{m} c_n e^{j2\pi nx/P}\, dx = \int_z^b u(x) f(x)\, dx$$

for every infinitely differentiable periodic function $u(x)$ with period P.

Fourier Sine Series and Fourier Cosine Series

A function $f(x)$ on $(-L, L)$ is said to be an even function if $f(-x) = f(x)$ for $x \in (-L, L)$; it is said to be an odd function if $f(-x) = -f(x)$ for $x \in (-L, L)$. The graph of an even function

is symmetric with respect to the y axis in the xy plane, while the graph of an odd function is symmetric with respect to the origin. Note that $\cos(\pi nx/L)$ is an even function in x ($n = 0, 1, 2, \ldots$), and $\sin(\pi nx/L)$ is an odd function in x ($n = 1,2,3, \ldots$). By exploring the symmetry, we can show that if $f(x)$ is an even function on $(-L, L)$, then its real Fourier series on $(-L, L)$ contains no sine terms; and if $f(x)$ is an odd function, then its real Fourier series is a series of sines only. Of course, a trigonometric series containing no sines must be even, if the series ever converges. Similarly, a trigonometric series containing only sines must be odd.

When a function $f(x)$ is defined on $(0, L)$, we can extend it to a larger interval $(-L, L)$. Among the infinitely many possible extensions, we consider the following two. The first one is to extend the function so that it is an even function on $(-L, L)$ (we still use $f(x)$ to denote the extension):

$$f(x) = \begin{cases} f(x), & 0 < x < L \\ f(-x), & -L < x < 0 \end{cases}$$

The other way is to extend it as an odd function on $(-L, L)$:

$$f(x) = \begin{cases} f(x), & 0 < x < L \\ -f(-x), & -L < x < 0 \end{cases}$$

Note that in the above definitions of the extensions of $f(x)$ we did not give any definition for the extension at $x = 0$. Because of an earlier remark, changing one value of a function will not change its Fourier series; so at $x = 0$ we can define $f(x)$ in any way we want. For example, in view of the convergence theorems, we may define $f(0) = f(0+)$ in the even extension, and $f(0) = 0$ in the odd extension.

Now we can form the Fourier series of $f(x)$ on the larger interval $(-L, L)$ after the extension. When we use the even extension, the Fourier series of $f(x)$ on $(-L, L)$ is given by

$$\frac{1}{2}A_0 + \sum_{n=1}^{\infty} A_n \cos\frac{\pi nx}{L} \tag{12}$$

with

$$\begin{aligned} A_n &= \frac{1}{L}\int_{-L}^{L} f(x)\cos\frac{\pi nx}{L}\,dx \\ &= \frac{1}{L}\left[\int_{-L}^{0} f(x)\cos\frac{\pi nx}{L}\,dx + \int_{0}^{L} f(x)\cos\frac{\pi nx}{L}\,dx\right] \end{aligned}$$

The last two integrals are equal since $f(x)\cos(\pi nx/L)$ is even. So the formula for the real Fourier coefficient A_n can be simplified to

$$A_n = \frac{2}{L}\int_{0}^{L} f(x)\cos\frac{\pi nx}{L}\,dx \tag{13}$$

Notice that this formula uses only the function $f(x)$ on its original domain $(0, L)$. This motivates the following definition of Fourier cosine series.

Definition. Let $f(x)$ be a function defined on $(0, L)$. Then its Fourier cosine series on $(0, L)$ is given by Eq. (12), where A_n are defined by Eq. (13) for $n = 0, 1, 2, \ldots$.

Similarly, we can define the Fourier sine series by using the odd extension of a function defined on $(0, L)$.

Definition. Let $f(x)$ be a function defined on $(0, L)$. Then its Fourier sine series on $(0, L)$ is given by

$$\sum_{n=1}^{\infty} B_n \sin\frac{\pi nx}{L}\,dx$$

where B_n's are defined by

$$B_n = \frac{2}{L}\int_{0}^{L} f(x)\sin\frac{\pi nx}{L}\,dx$$

for $n = 0, 1, 2, \ldots$.

In Fig. 5, we show the graph of function $f(x) = \pi - x$ on $(0, \pi)$, the even extension of $f(x)$ on $(-\pi, \pi)$, and the odd extension of $f(x)$ on $(-\pi, \pi)$; in Fig. 6, we sketch some partial sums of the Fourier cosine and Fourier sines series of the even and the odd extensions, respectively. It is clear from graphs that on the interval $(0, \pi)$, the Fourier cosine series provides much better approximation to $f(x)$ than the Fourier sine series. This is because the even extension is continuous while the odd extension has a jump at $x = 0$.

Gibb's Phenomenon

In Fig. 4, notice the sharp peaks near 0, the point of discontinuity of $f(x)$. That this is not an isolated case was first explained in 1899 (in a letter to *Nature*) by Josiah Gibbs in re-

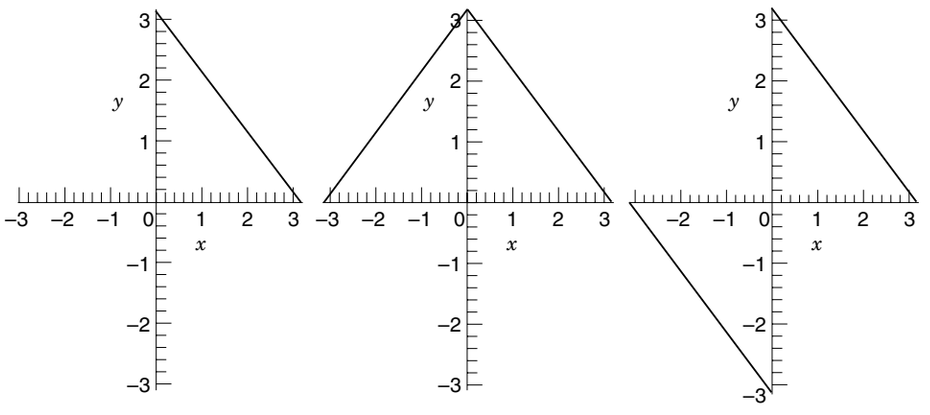


Figure 5. Graphs of $f(x)$ on $(0, \pi)$, and its even and odd extensions on $(-\pi, \pi)$.

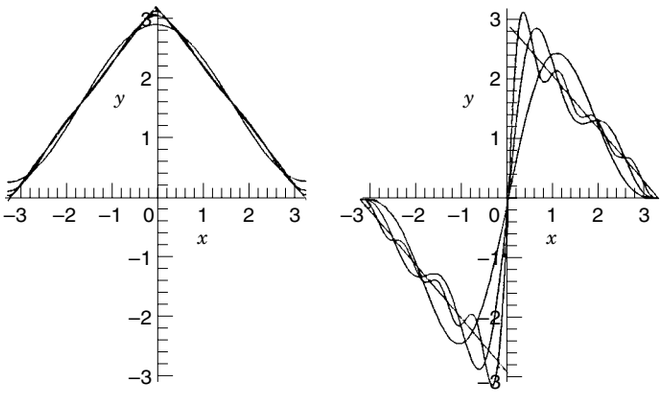


Figure 6. The even extension of $f(x)$ and the Fourier cosine series on $(-\pi, \pi)$; the odd extension and the Fourier sine series on $(-\pi, \pi)$.

sponse to a question of the American physicist Albert Michelson, who observed such a phenomenon in his experiments on superposition of harmonics. It can be proved that Gibbs's phenomenon occurs at x_* whenever it is a discontinuous point of a piecewise smooth function $f(x)$. In Fig. 7, we show a close-up of what is happening near the right side of 0 for the function $f(x)$ in Example 2, together with the partial sums of its Fourier series S_{12}, S_{24}, and S_{36}. Note that the amount of overshot is almost unchanged.

The Discrete Fourier Transform

As a motivation for the definition of the discrete Fourier transform, let us consider the computation of the complex Fourier coefficients,

$$c_n = \frac{1}{P}\int_a^b f(x)e^{-j2\pi nx/P}\,dx \tag{14}$$

when we only know the values of $f(x)$ at evenly spaced points in (a, b), say $x_k = a + k\delta$ for $k = 0, 1, \ldots, N-1$ with $\delta = P/N$. We have to approximate the integral in Eq. (14). To do this, we use a left-endpoint Riemann sum:

$$\begin{aligned} c_n &\approx \frac{1}{P}\sum_{k=0}^{N-1} f(x_k)e^{-j2\pi na/P}\,e^{-j2\pi nk\delta/P}\,\frac{P}{N} \\ &= \frac{1}{Ne^{j2\pi na/P}}\sum_{k=0}^{N-1} f(x_k)e^{-j2\pi nk/N} \end{aligned} \tag{15}$$

The final summation in Eq. (15) motivates the following definition.

Definition. Let $\{h_k\}_{k=0}^{N-1}$ be a set of complex numbers. The discrete Fourier transform of $\{h_k\}_{k=0}^{N-1}$ is denoted by $\{H_n\}$ and is given by

$$H_n = \sum_{k=0}^{N-1} h_k e^{-j2\pi nk/N}$$

for $n = 0, \pm 1, \pm 2, \ldots$.

Although H_n is defined for all integers n, there are only at most N distinct values since $H_{n+N} = H_n$. So, we can just use $\{H_n\}_{n=0}^{N-1}$. Therefore, the discrete Fourier transform maps a set of N numbers ($\{h_k\}_{k=0}^{N-1}$) into a set of N numbers ($\{H_n\}_{n=0}^{N-1}$).

Using the terminology of the discrete Fourier transform, we can say that Eq. (15) gives c_k, the kth complex Fourier coefficient of the function $f(x)$, as approximately $(Ne^{j2\pi na/P})^{-1}F_k$, with $\{F_k\}$ being the discrete Fourier transform of $\{f(x_k)\}_{k=0}^{N-1}$. Of course, in order to get satisfactory approximation of c_k, we have to choose a large value for N. Based on numerical observations (see, for example, Ref. 3), it seems that we need to make $N \geq 8|k|$ to ensure some degree of good approximation of c_k.

As in the case of Fourier series, the orthogonality property plays a very important role in the discrete Fourier transform. We now have

$$\sum_{k=0}^{N-1} e^{j2\pi mk/N}\,e^{-j2\pi nk/N} = \begin{cases} 0, & \text{if } m \neq n \\ N, & \text{if } m = n \end{cases}$$

With the orthogonality property, we derive the following inversion formula for the discrete Fourier transform: If $\{H_n\}_{n=0}^{N-1}$ is the discrete Fourier transform of $\{h_k\}_{k=0}^{N-1}$, then

$$h_k = \frac{1}{N}\sum_{n=0}^{N-1} H_n e^{j2\pi nk/N}, \qquad k = 0, 1, \ldots, N-1 \tag{16}$$

Equation (16) defines the discrete *inverse* Fourier transform of $\{H_n\}_{n=0}^{N-1}$. Note that there are only two differences in the definitions of the discrete Fourier transform and the discrete inverse Fourier transform: (1) opposite signs in the exponential and (2) presence or absence of a factor $1/N$. This means that an algorithm for calculating the discrete Fourier transforms can also calculate the discrete inverse Fourier transforms with minor changes.

In the literature of Fourier analysis, there are alternative ways to define the discrete Fourier transforms. One variation is in the factor $1/N$; some authors use it in the definition of the discrete Fourier transforms instead of putting it in the discrete inverse Fourier transforms like we do here. Another

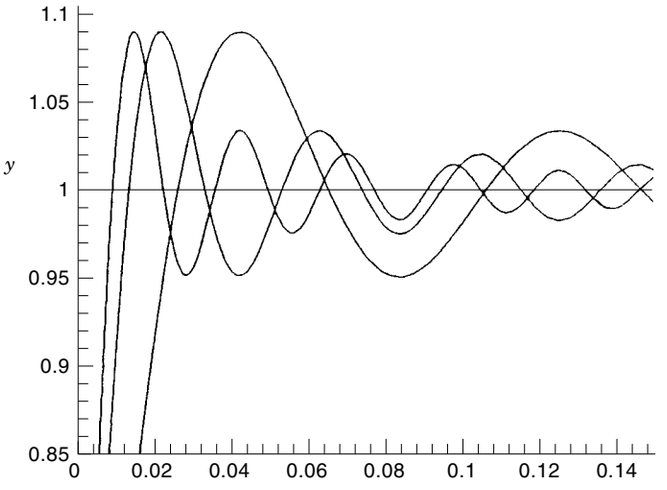


Figure 7. Gibbs' phenomenon.

difference is the ranges of the indices; there are good reasons in favor of (or against) the use of indices running either between 0 and $N-1$ or from $-N/2$ to $N/2$. With little modification, anything that can be done for one version of discrete Fourier transform can also be done for other versions. For this reason, we now state an alternate definition.

Discrete Fourier Transform (Alternate Version). (1) Let N be an even positive integer and let $\{h_k\}_{k=-N/2+1}^{N/2}$ be a set of complex numbers. Then its discrete Fourier transform is given by

$$H_n = \sum_{k=-N/2+1}^{N/2} h_k e^{-j2\pi nk/N}, \qquad k = -\frac{N}{2}+1, \ldots, \frac{N}{2}$$

(2) If N is an odd positive integer and $\{h_k\}_{k=-N/2+1}^{N/2}$ is a set of N complex numbers, then its discrete Fourier transform is given by

$$H_n = \sum_{k=-(N-1)/2}^{(N-1)/2} h_k e^{-j2\pi nk/N}, \qquad k = -\frac{N-1}{2}, \ldots, \frac{N-1}{2}$$

Continuous Fourier Transform

Now, we briefly discuss the last of the three types of transforms in Fourier analysis: the continuous Fourier transform. It is also referred to simply as the Fourier transform. It applies to functions of a continuous variable that runs on the whole real line $(-\infty, \infty)$. Given a signal $f(x)$, the *Fourier transform* $F(\xi)$ of $f(x)$ is defined by

$$F(\xi) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi\xi x}\, dx, \qquad -\infty < \xi < \infty$$

where ξ is usually called the *frequency variable.* Due to the presence of the complex exponential $e^{-j2\pi\xi x}$ in the integrand of the above integral, the values of $F(\xi)$ may be complex. So, to specify $F(\xi)$, it is necessary to display both the magnitude and the angle of $F(\xi)$. From the definition, like the Fourier coefficients, Fourier transform of a function $f(x)$ is defined only if the above integral makes sense. A function $f(x)$ is said to be absolutely integrable if

$$\int_{-\infty}^{\infty} |f(x)|\, dx < \infty \tag{17}$$

Let L^1 denote the set of all absolutely integrable functions. [That is, L^1 denotes the set of all Lebesgue integrable functions defined on $(-\infty, \infty)$.] It is a well-known fact in the theory of Lebesgue integration that if $|f(x)|$ is Riemann integrable on $(-\infty, \infty)$, then $f(x)$ is in L^1. For the readers not familiar with the Lebesgue integration, it is safe to interpret all integrals as Riemann integrals since most signals in practice are Riemann integrable.

Equation (17) guarantees that the Fourier transform $F(\xi)$ of $f(x)$ is well-defined. Actually, in this case, $F(\xi)$ is a (uniformly) continuous function of ξ in $(-\infty, \infty)$. But, a continuous function on $(-\infty, \infty)$ is not necessarily in L^1. For example, the constant function $f(x) = 1$, $-\infty < x < \infty$, has infinite area under its graph over $(-\infty, \infty)$. Thus, even if $f(x)$ is in L^1 so that $F(\xi)$ is uniformly continuous, still we cannot assert that $F(\xi)$ is in L^1. The following result shows the magic when we know that $F(x)$ is in L^1.

The Inversion Theorem. Let $f(x)$ be in L^1 and let $F(\xi)$ denote its Fourier transform. Assume that $F(\xi)$ is in L^1. Then

$$f(x) = \int_{-\infty}^{\infty} F(\xi) e^{j2\pi\xi x}\, d\xi, \tag{18}$$

for almost every x in $(-\infty, \infty)$.

This theorem tells us that under the suitable conditions, we can recover a function from its Fourier transform. This is why the integral in Eq. (18) is called the *inverse Fourier transform* of $F(\xi)$. Notice the positive sign in the exponential.

When we treat Fourier transforms as operations on functions, it is more convenient to use the following notation to indicate that $F(\xi)$ is the Fourier transform of $f(x)$:

$$\mathscr{F}(f) = F \text{ or } f(x) \xrightarrow{\mathscr{F}} F(\xi)$$

Similarly, the Fourier inverse transform is denoted by $\mathscr{F}^{-1}$.

Examples

Given a function, it is not always easy to find its Fourier transform explicitly. For simple functions, tables of Fourier transform formulas are available in most books on Fourier transforms. Symbolic mathematical packages all have Fourier transform routines. Here we look at two important cases where it is possible to find the Fourier transform explicitly.

Example 1 We verify that $\mathrm{F}(\xi) = \sin \pi\xi/(\pi\xi)$ when

$$f(x) = \begin{cases} 1, & \text{for } |x| \le 1/2 \\ 0, & \text{for } |x| > 1/2 \end{cases}$$

Indeed, for $\xi \neq 0$, we have

$$\begin{aligned} F(\xi) &= \int_{-\infty}^{\infty} f(x) e^{-jz\pi\xi x}\, dx \\ &= \int_{-1/2}^{1/2} e^{-jz\pi\xi x}\, dx \\ &= -\frac{e^{-jz\pi\xi x}}{jz\pi\xi}\bigg|_{x=-1/2}^{x=1/2} \\ &= \frac{\sin \pi\xi}{\pi\xi} \end{aligned}$$

Example 2. Let $f(x) = \mathrm{e}^{-\pi x^2}$. Then

$$F(\xi) = \int_{-\infty}^{\infty} e^{-\pi x^2} e^{-j2\pi\xi x}\, dx = \int_{-\infty}^{\infty} e^{-(\sqrt{\pi}x)^2 - 2(\sqrt{\pi}x)(j\sqrt{\pi}\xi)}\, dx$$

The integrand is equal to

$$e^{-(\sqrt{\pi}x + j\sqrt{\pi}\xi)^2 - \pi\xi^2}$$

By using Cauchy's Theorem to shift the path of integration from the real axis to the horizontal line $\mathrm{Im}(z) = \sqrt{\pi}\xi$ [with fixed ξ in $(-\infty, \infty)$], one can derive that

$$F(\xi) = e^{-\pi\xi^2}$$

Notice that in this example both the function and its Fourier transform are given by the same formula.

Some Important Results

We now discuss some most important results in Fourier transforms. Among them, the most significant one is related to a very useful operation in Fourier analysis, the convolution of two functions. If $f(x)$ and $g(x)$ are two functions on $(-\infty, \infty)$, their *convolution* is the function $f * g(x)$ defined by

$$f * g(x) = \int_{-\infty}^{\infty} f(y)g(x-y)\,dy$$

provided that the integral exists. Note that if $f * g$ is well-defined, then so is $g * f$, and $f * g(x) = g * f(x)$. There are various assumptions on $f(x)$ and $g(x)$ to ensure that the convolution $f * g(x)$ is defined for all x in $(-\infty, \infty)$. For example:

1. Assume that $f(x)$ is in L^1 and $g(x)$ is bounded (say $|g(x)| < C$ for all x). Then $f * g(x)$ is defined for all x since
$$\int_{-\infty}^{\infty} |f(y)g(x-y)|\,dy \le C\int_{-\infty}^{\infty} |f(x)| < \infty$$
2. Assume that $|f(x)|^2$ and $|g(x)|^2$ are in L^1. Then, using the Cauchy–Schwarz inequality, we have
$$\int_{-\infty}^{\infty} |f(y)g(x-y)|\,dy \le \sqrt{\int_{-\infty}^{\infty} |f(y)|^2\,dy}\sqrt{\int_{-\infty}^{\infty} |g(x-y)|^2\,dy} < \infty$$
We will use L^2 to denote the set of all functions $f(x)$ such that $|f(x)|2$ is in L^1.
3. Assume that both $f(x)$ and $g(x)$ are in L^1. Then it can be shown that $f * g(x)$ exists for "almost every" x, and $f * g(x)$ is itself in L^1 (see Ref. 4, Sec. 8.1).

The most important property of the Fourier transform is the following result.

Convolution Theorem. Suppose that $f(x)$ and $g(x)$ are in L^1, and $F(\xi)$ and $G(\xi)$ are their Fourier transforms, respectively. Then the Fourier transform of $f * g(x)$ is given by $F(\xi)G(\xi)$; that is,

$$\mathscr{F}(f*g)(\xi) = F(\xi)G(\xi) \quad \text{or} \quad f*g(x) \xrightarrow{\mathscr{F}} F(\xi)G(\xi)$$

The next result is closely related to sampling theory.

Poisson Summation. Let $f(x)$ be a continuous function in L^1 and $F = \mathscr{F}(f)$. If $\sum_{n=-\infty}^{\infty} f(x - 2nL)$ defines a continuous function on $(-L, L)$, and if $\sum_{-\infty}^{\infty} |F(n/2L))|$ converges, then

$$\sum_{-\infty}^{\infty} f(x-2nL) = \sum_{-\infty}^{\infty} \frac{1}{2L} F\left(\frac{n}{2L}\right) e^{jn\pi x/L}$$

Since the integral of the absolute-value squared of a function can be interpreted as its energy, the following identity, Parseval's identity, expresses the fact that a signal's energy is equal to its frequency energy.

Parseval's Identity. Let $f(x)$ be a function in $L^1 \cap L^2$, and let $F(\xi)$ be its Fourier transform. Then

$$\int_{-\infty}^{\infty} |f(x)|^2\,dx = \int_{-\infty}^{\infty} |F(x)|^2\,dx$$

SIGNAL SAMPLING

When we analyze signals using a computer, we are no longer working with continuous time signals, but rather with discrete time functions. This requires the sampling of the continuous signals. Let $f(x)$ be a signal. Let us assume that we sample at equally spaced time intervals, Δ, so that the sequence of sampled signal of $f(x)$ is

$$f_n = f(n\Delta), \qquad n = 0, \pm 1, \pm 2, \ldots$$

We call the number $1/\Delta$ the *sampling rate;* it is the number of samples recorded per second, if time is measured in seconds. Half of the sampling rate is a critical value called the *Nyquist critical frequency,* denoted by f_c; that is, $f_c = 1/(2\Delta)$. The importance of the Nyquist critical frequency can be seen in the following result.

Sampling Theorem. Suppose $f(x)$ is a continuous function in L^1 and $F(\xi) = 0$ for all $|\xi| > f_c$. Then $f(x)$ is completely determined by its values f_n at $n\Delta$, $n = 0, \pm 1, \pm 2, \ldots$. In fact,

$$f(x) = \Delta \sum_{n=-\infty}^{\infty} f_n \frac{\sin[2\pi f_c(x - n\Delta)]}{\pi(x - n\Delta)}$$

A function $f(x)$ whose Fourier transform $F(\xi)$ vanishes outside of a finite interval is said to be *bandwidth-limited.* Therefore, a bandwidth-limited signal whose frequencies are bounded in $[-f_c, f_c]$ can be fully recovered from its sampled values at $n\Delta$, $n = 0, \pm 1, \pm 2, \ldots$ if the sampling rate is twice its Nyquist critical frequency f_c, that is, $1/\Delta = 2f_c$.

NUMERICAL COMPUTATION

Fast Fourier Transform

The fast Fourier transform (FFT) is a family of methods for computing the discrete Fourier transform of a function with minimum computational effort. The FFT became well known after the publication of the article by Cooley and Tukey in 1965, although it had been used in various forms by others before this. Various forms of FFT are available as subroutines in almost every mathematical software package, such as Matlab, Mathematica, and Maple, to name a few. In this section, we provide some basics of the FFT so that the reader will be able to make the best use of it.

Recall that the discrete Fourier transform of N numbers $[h_k]_{k=0}^{N-1}$ is given by

$$H_n = \sum_{k=0}^{N-1} h_k e^{-j2\pi nk/N}, \qquad n = 0, 1, \ldots, N-1$$

Let $W_N = e^{-j2\pi/N}$. Then W_N is an Nth root of the unity; that is,

$$W_N^N = 1$$

Notice that if we compute the transformed points $[H_n]_{n=0}^{N-1}$ directly by their definitions, we need to use N multiplications for each H_n. So, the N numbers H_n, $n = 0, 1, \ldots, N-1$, would require N^2 multiplications. This can result in a great deal of computation when N is large. It turns out that the discrete Fourier transform of a data set of length N can be computed by using the FFT algorithm, which requires only $(N \log_2 N)/2$ multiplications. This is a significant decrease in the N^2 multiplications required in the direct evaluation of the transform. For example, if $N = 1024$, the direct evaluation requires $N^2 = 1{,}048{,}576$ multiplications. In contrast, the FFT algorithm requires $(1024\log_2 1024)/2 = 5120$ multiplications.

Suppose $N = 2^M$, where M is a positive integer. Let us split the sum for each n into even and odd parts:

$$\begin{aligned} H_n &= \sum_{k=0}^{N-1} h_k W_N^{nk} \\ &= \sum_{k=0}^{N/2-1} h_{2k} W_N^{n(2k)} + \sum_{k=0}^{N/2-1} h_{2k+1} W_N^{n(2k+1)} \\ &= \sum_{k=0}^{N/2-1} h_{2k} W_{N/2}^{nk} + W_N^n \sum_{k=0}^{N/2-1} h_{2k+1} W_{N/2}^{nk} \end{aligned}$$

Let us write

$$H_n = H_n^0 + W_N^n H_n^1 \tag{19}$$

with

$$H_n^0 = \sum_{k=0}^{N/2-1} h_{2k} W_{N/2}^{nk}$$

and

$$H_n^1 = \sum_{k=0}^{N/2-1} h_{2k+1} W_{N/2}^{nk}$$

Note that $[H_n^0]_{n=0}^{N/2-1}$ and $[H_n^1]_{n=0}^{N/2-1}$ are, respectively, the discrete Fourier transforms of the even components $[h_{2k}]_{k=0}^{N/2-1}$ and the odd components $[h_{2k+1}]_{k=0}^{N/2-1}$. Note also that $[H_n^0]_{n=0}^{N/2-1}$ and $[H_n^1]_{n=0}^{N/2-1}$ are of length $N/2$. So, we have

$$H_{n+N/2}^0 = H_n^0 \quad \text{and} \quad H_{n+N/2}^1 = H_n^1, \qquad n = 0, 1, \ldots, \frac{N}{2} - 1$$

This, together with the fact that $W_N^{n+N/2} = -W_N^n$ ($n = 1, 2, \ldots, N/2 - 1$), allows us to write the equation in Eq. (19) as

$$H_n = H_n^0 + W_N^n H_n^1 \quad \text{and} \quad H_{n+N/2} = H_n^0 - W_N^n H_n^1 \tag{20}$$

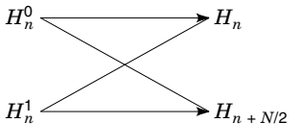


Figure 8. The butterfly diagram.

for $n = 1, 2, \ldots, N/2 - 1$. This pair of calculations is called *combined formulae* or *butterfly relations* since it can be visualized in the so-called *butterfly diagram* (see Fig. 8).

The splitting of $\{H_n\}$ into two half length (i.e., $N/2$) discrete Fourier transforms $\{H_n^0\}$ and $\{H_n^1\}$ can be applied now on a smaller scale. In other words, we define H_n^{00} and H_n^{01} to be the discrete Fourier transforms of the even and odd components of $\{h_{2k}\}_{k=0}^{N/2-1}$ and define H_n^{10} and H_n^{11} to be the discrete Fourier transforms of the even and odd components of $\{h_{2k+1}\}_{k=0}^{N/2-1}$, respectively. We then get

$$H_n^0 = H_n^{00} + W_{N/2}^n H_n^{01} \quad H_{n+N/4}^0 = H_n^{00} - W_{N/2}^n H_n^{01}$$

and

$$H_n^1 = H_n^{10} + W_{N/2}^n H_n^{11} \quad H_{n+N/4}^1 = H_n^{10} - W_{N/2}^n H_n^{11}$$

for $n = 0, 1, \ldots, N/4 - 1$.

If we continue with this process of halving the length of the discrete Fourier transforms, then after $M = \log_2 N$ steps we reach the point where we are performing the transforms on data of length 1, which is trivial since the discrete Fourier transform of a data set of length 1 is the identity transform. For each of the M steps, there are $N/2$ multiplications, and so there are about $MN = (N \log_2 N)/2$ multiplications needed for the FFT as we claimed at the beginning of this section.

Another important step in FFT is that the repeated processes of splitting the data set into the even and odd subsets of half length can be realized by bits reversal. That is, we order $\{h_k\}$ so that h_k is put at the k'th place, where $k' = a_p a_{p-1} \ldots a_1$ (base 2) if $k = a_1 a_2 \ldots a_p$ (base 2). After the reordering of $\{h_k\}$, we start to use the butterfly relations to combine the adjacent pairs to get 2-point transforms, then combine the 2-point transforms to get the 4-point transforms, and so on, until the final transform $\{H_n\}$ is formed from two $N/2$-point transforms. So, there are two major steps in the FFT algorithm: The first step sorts the data into bit-reversed order. This can be done without additional storage and involves at most $N/2$ swaps of the elements of a data set of length N. The second step calculates, in turn, transforms of length 2, 4, . . ., N. Figure 9 shows steps in the FFT algorithm for $N = 8$.

We have only discussed the case when N is a power of 2, the so-called radix 2 case. The algorithm first reorders the input data in bit-reversed order, then builds up the transform in $\log_2 N$ steps. This is referred to as the *decimation-in-time* FFT. It is also possible to go through the $\log_2 N$ steps of transforms and then rearrange the output into the bit-reversed order. This is the *decimation-in-time* FFT.

There are higher-dimensional generalizations of the FFT for transforming complex functions defined over a two- or higher-dimensional grid (see Ref. 5).

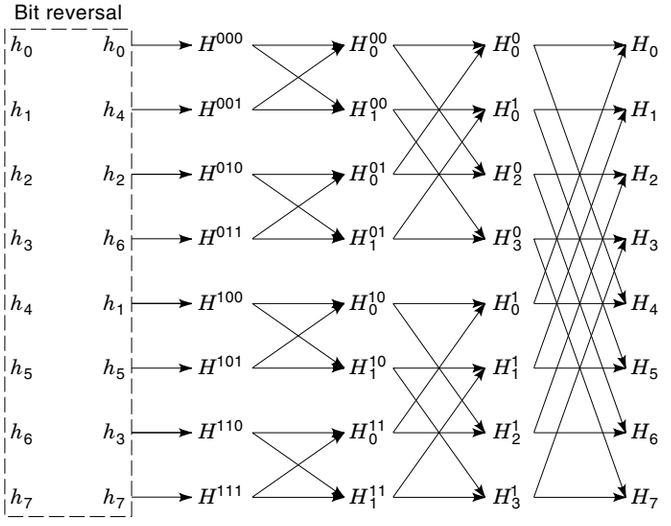


Figure 9. The decimation-in-time (radix 2) FFT for a data set of length $N = 8$.

Fast Sine and Cosine Transform

Discrete sine and cosine transforms can be derived from the Fourier sine and cosine series discussed earlier.

Discrete sine and cosine transforms. For a real sequence $\{h_k\}_{k=1}^{N-1}$ the discrete sine transform (DST), $\{H_n^S\}_{n=1}^{N-1}$, is given by

$$H_n^S = \sum_{k=1}^{N-1} h_k \sin \frac{\pi nk}{N}, \qquad n = 1, 2, \ldots, N-1$$

For a real sequence $\{h_k\}_{k=0}^{N-1}$ the discrete cosine transform (DCT), $\{H_n^C\}_{n=0}^{N-1}$, is given by

$$H_n^C = \sum_{k=0}^{N-1} h_k \cos \frac{\pi nk}{N}, \qquad n = 1, 2, \ldots, N-1$$

The form of DCT given above is only one of several commonly used DCTs. DSTs and DCTs are related to odd/even symmetry of DFTs, as with Fourier sine and cosine series. See Ref. 6. To compute the DSTs and DCTs, there exist the fast sine transform and fast cosine transform. These algorithms are usually implemented by using the FFT routine (see Refs. 3 and 5).

WAVELETS APPROACH

In Fourier analysis, every function is expanded into a series or an integral of sines and cosines which are themselves analytic functions. When approximating a function with a point of discontinuity, the partial sums of its Fourier series do not converge uniformly in any neighborhood of the point of discountinuity; they do a very poor job in approximating sharp spikes. (Recall that near an isolated point of discontinuity of a function of bounded variation, the Gibbs' phenomenon occurs.) For many years, scientists have searched for more appropriate functions than sines and cosines to represent functions with discontinuities. With the construction of smooth, compactly supported, orthogonal wavelet basis functions (now referred to as the Daubechies wavelets) by Ingrid Daubechies in 1988, wavelet analysis emerged as a powerful toolbox leading to new and varied applications in, for example, data compression, signal and image processing, nuclear engineering, geology, and such pure mathematics as solving differential equations.

A (mother) wavelet $\Phi(x)$ is a function in L^1 that has zero average:

$$\int_{-\infty}^{\infty} \Phi(x)\, dx = 0$$

Useful wavelets satisfy further conditions that we will not specify in this article since a more detailed discussion on wavelets is given in WAVELET TRANSFORMS. Here, we just try to give a very brief introduction of the wavelet theory and compare it with the classical Fourier analysis discussed in this article. In Fig. 10, we show two typical mother wavelets: Haar wavelet and Daubechies wavelet (DAUB6).

The Discrete Wavelet Transform

Unlike Fourier analysis, in which we represent functions by series of sines and cosines, in wavelet analysis we represent

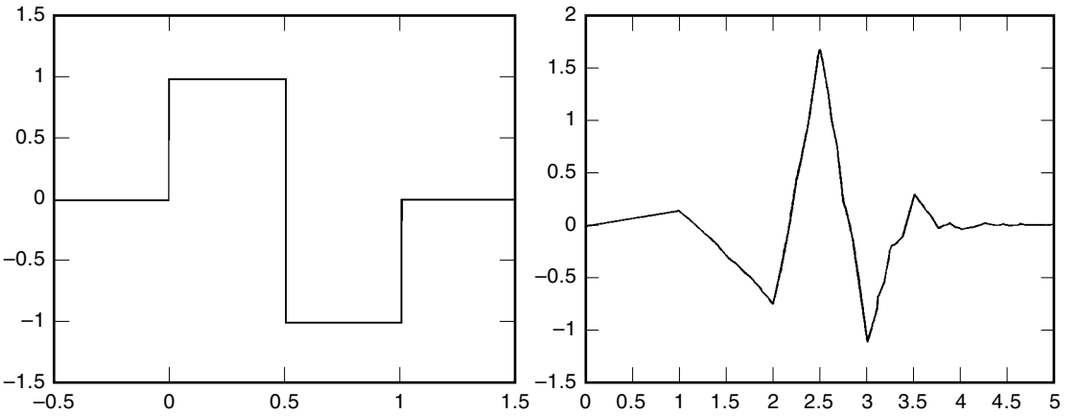


Figure 10. Haar wavelet and Daubechies wavelet (DAUB6).

functions by series of dilations and translations of a single function called the *mother wavelet* $\Phi(x)$:

$$\Phi_{s,l}(x) = 2^{-s/2}\Phi(2^{-s}x - l), \qquad s, l = 0, \pm 1, \pm 2, \ldots$$

So, we would like to write

$$f(x) = \sum_{s,l=-\infty}^{\infty} f_{s,l}\, \Phi_{s,l}(x)$$

See WAVELET TRANSFORMS or Ref. 7. The set of coefficients $\{f_{s,l}\}$ is called the discrete wavelet transform of $f(x)$ (with respect to mother wavelet $\Phi(x)$). Of course, as in the case of Fourier series, the above "equality" needs suitable interpretations. More important to applications is the fact that mother wavelets Φ can be chosen best adapted to the problems at hand. This is possible because there are many different mother wavelets with different properties available. There is even a fairly general scheme for generating various mother wavelets, a procedure called *multiresolution analysis.*

To compute the wavelet transform, we face the same complexity issue that we previously faced for the computation of the discrete Fourier transform. Fortunately, there exists a "fast" wavelet transform algorithm that requires only order n operations to transform an n-sample vector.

Wavelet Analysis Versus Fourier Analysis

We start with the similarities between these two. The discrete Fourier transform and discrete wavelet transform are both linear operations that can be carried out in "almost linear" time; that is, about $n \log_2 n$ or n operations (addition, multiplication) are needed to transform a sample vector of size n. Another similarity is that both basis functions, sines and cosines in Fourier analysis, and dilations and translations of a mother wavelet in wavelet analysis are localized in frequency, making spectral analysis possible. When used to represent smooth or stationary signals, both Fourier and wavelet methods perform almost equally well.

The most striking difference between these two kinds of transforms is that wavelet functions are localized in time (or space) domain, while Fourier sines and cosines are not. When analyzing nonstationary signals, it is often desirable to be able to acquire a correlation between the time and frequency domains of a signal. The Fourier analysis provides information about the frequency domain, but time localized information is essentially lost in the process. In contrast to the Fourier analysis, the wavelet transforms allow exceptional localization in time domain via translations of the mother wavelet, as well as in frequency (scale) domain via dilations.

CONCLUSIONS

Given a signal, Fourier analysis decomposes it into its frequency components. This decomposition can be used to represent the original signal if it is smooth (or piecewise smooth) and time-invariant (stationary). When a signal has lots of jumps (points of discontinuity), at each jump Gibbs' phenomenon may occur. In the case of analyzing transient or nonstationary signals, wavelet transforms provide much desired tools. The mathematical theory of wavelets has been more or less established in the last decade. The future of wavelets lies in the discovery of its applications in every discipline of engineering. See FREQUENCY DOMAIN CIRCUIT ANALYSIS, WAVELET TRANSFORMS.

BIBLIOGRAPHY

1. A. Zygmund, *Trigonometric Series,* Cambridge: Cambridge Univ. Press, 1968.
2. J. M. Ash (ed.), *Studies in Harmonic Analysis,* The Mathematical Association of America, 1976.
3. J. S. Walker, *Fast Fourier Transforms,* Second Edition, Boca Raton, FL: CRC Press, 1996.
4. G. B. Folland, *Fourier Analysis and its Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1985.
5. E. O. Brigham, *The Fast Fourier Transform and Its Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1988.
6. W. L. Briggs and V. E. Henson, *The DFT, An Owner's Manual for the Discrete Fourier Transform,* Philadelphia, PA: Society of Industrial and Applied Mathematics, 1995.
7. S. Mallat, *A Wavelet Tour of Signal Processing,* San Diego, CA: Academic, 1998.

XIN LI
University of Central Florida

FOURIER SERIES. See POWER SYSTEM HARMONICS.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Function Approximation
Standard Article

Mohamed N. Ahmed[1], Sameh M. Yamany[1], Aly A. Farag[1]
[1]University of Louisville, Louisville, KY

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2436
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (432K)

Abstract

The sections in this article are

Classical Interpolation Techniques

Neural Network as Universal Approximator

Hypersurface Reconstruction as an Ill-Posed Problem

Results

Applications

| | |

Copyright © 1999-2008 All Rights Reserved.

FUNCTION APPROXIMATION

This article presents a survey of techniques used for function approximation and free form surface reconstruction. A comparative study is performed between classical interpolation methods and two methods based on neural networks. We show that the neural networks approach provides good approximation and provides better results than classical techniques when used for reconstructing smoothly varying surfaces.

The interpolation problem, in its strict sense, may be stated as follows (1): Given a set of N different points: $\{x_i \in \Re^m | i = 1, 2, \ldots, N\}$ and a corresponding set of N real numbers: $\{d_i \in \Re^n | i = 1, 2, \ldots, N\}$ find a function F:

$$F : \Re^m \rightarrow \Re^n | F(x_i) = d_i, i = 1, 2, \ldots, N \tag{1}$$

where m and n are integers. The interpolation surface is constrained to pass through all the data points. The interpolation function can take the form:

$$F(x) = \sum_{i=1}^{N} w_i \phi(x, x_i) \tag{2}$$

where $\{\phi(x, x_i) | i = 1, 2, \ldots, N\}$ is a set of N arbitrary functions known as the radial basis functions. Inserting the interpolation conditions, we obtain the following set of simultaneous linear equations for the unknown coefficients (weights) of the expansion w_i:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdot & \cdot & \phi_{1N} \\ \phi_{21} & \phi_{22} & \cdot & \cdot & \phi_{2N} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \phi_{N1} & \phi_{N2} & \cdot & \cdot & \phi_{NN} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \cdot \\ \cdot \\ w_N \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \cdot \\ \cdot \\ d_N \end{bmatrix} \tag{3}$$

where

$$\phi_{ji} = \phi(x, x_i), j, i = 1, 2, ..N \tag{4}$$

Let the $N \times 1$ vectors $\boldsymbol{d}$ and $\boldsymbol{W}$ defined as

$$\boldsymbol{d} = [d_1, d_2, \ldots, d_N]^t \tag{5}$$

$$\boldsymbol{W} = [w_1, w_2, \ldots, w_N]^t \tag{6}$$

represent the desired response vector and the linear weight vector, respectively. Let

$$\Phi = \{\phi_{i,j}, i, j \in [1, N]\} \tag{7}$$

denote the $N \times N$ interpolation matrix. Hence, Eq. (3) can be rewritten as

$$\Phi \boldsymbol{W} = \boldsymbol{d} \tag{8}$$

Provided that the data points are all distinct, the interpolation matrix Φ is positive definite (1). Therefore, the weight vector $\boldsymbol{W}$ can be obtained by

$$\boldsymbol{W} = \Phi^{-1} \boldsymbol{d} \tag{9}$$

where Φ^{-1} is the inverse of the interpolation matrix Φ. Theoretically speaking, a solution to the system in Eq. (9) always exits. Practically speaking, however, the matrix Φ can be singular. In such cases, regularization theory can be used; where the matrix Φ is perturbed to $\Phi + \lambda \mathbf{I}$ to assure positive definiteness (1).

Based on the interpolation matrix Φ, different interpolation techniques are available (2–5). Some of these techniques will be reviewed in next section.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

CLASSICAL INTERPOLATION TECHNIQUES

Shepard's Interpolation

Shepard formulated an explicit function for interpolating scattered data (16). The value of the modeling function is calculated as a weighted average of data values according to their distance from the point at which the function is to be evaluated. Shepard's expression for globally modeling a surface is

$$S(x,y) = \begin{cases} \frac{\sum_{i=1}^{n} Z_i/r_i^2}{\sum_{i=1}^{n} 1/r_i^2} & r_i \neq 0 \\ z_i & r_i = 0 \end{cases} \tag{10}$$

where r_i is the standard distance metric:

$$r_i = [(x - x_i)^2 + (y - y_i)^2]^{1/2} \tag{11}$$

The above algorithm is global; local Shepard methods can be formed by evaluating S(x, y) from the weighted data values within a disk of radius R from (x, y). A function $\Psi(r)$ is defined that ensures the local behavior of the interpolating method by calculating a surface model for any $r <= R$, and which also weights the points at $r <= R/3$ more heavily, as follows:

$$\Psi(r) = \begin{cases} 1/r & 0 < r < R/3 \\ 27(r/R - 1)^2/4R & R/3 < r \leq R \\ 0 & R < r \end{cases} \tag{12}$$

Therefore, for points that are within R distance of (x, y), the surface is given by

$$S(x,y) = \begin{cases} \frac{\sum_{i=1}^{n} z_i \psi(r_i)^2}{\sum_{i=1}^{n} \psi(r_i)^2} & r_i \neq 0 \\ z_i & r_i = 0 \end{cases} \tag{13}$$

A usable number of points must fall within the local region of radius R for this method to be applicable. Shepard's method is simple to implement and can be localized, which is advantageous for the large cloud data sets (3).

Thin Plate Splines

The method of thin plate splines proposed by Shepard (2) and the multiquadric method proposed by Hardy (5) can both be classified as interpolating functions composed of a sum of radial basis functions. The basis functions are radially symmetric about the points at which the interpolating function is evaluated. Conceptually, the method is simple to understand in terms of a thin, deformable plate passing through the data points collected off the surface of the object. The thin plate spline radial basis functions are obtained from the solution of minimizing the energy of the thin plate constrained to pass through loads positioned at the cloud data set. The modeling surface is constructed from the radial basis functions $\beta_i(x, y)$ by expanding them in a series of $(n + 3)$ terms with c_i coefficients:

$$S(x,y) = \sum_{i=1}^{n} c_i \beta_i(x,y) \tag{14}$$

where the basis functions are given by

$$\beta_i(x,y) = r_i^2 ln(r_i) \tag{15}$$

The modeling surface function $S(x, y)$ has the form derived in Harder and Desmairis (11)

$$S(x,y) = a_0 + a_1 x + a_2 y + \sum_{i=1}^{n} c_i r_i^2 ln(r_i) \tag{16}$$

The coefficients are determined by substituting the discrete data set into and solving the resulting set of linear equations:

$$\sum_{i=1}^{n} c_i = 0 \tag{17}$$

$$\sum_{i=1}^{n} x_i c_i = 0 \tag{18}$$

$$\sum_{i=1}^{n} y_i c_i = 0 \tag{19}$$

$$f(x_i, y_i) = a_0 + a_1 x + a_2 y + \sum_{i=1}^{n} c_i r_i^2 ln(r_i) \tag{20}$$

Hardy's Multiquadric Interpolation. Hardy (5) proposed a method for interpolating scattered data that employs the summation of equations of quadratic surfaces that have unknown coefficients. The multiquadric basis functions are the hyperbolic quadrics

$$\phi_i = (r_i^2 + b^2)^{1/2} \tag{21}$$

where b is a constant and r_i is the standard Euclidean distance metric. The summation of a series of hyperbolic quadrics have been found to perform best compared with the other members of the multiquadrics family. The modeling surface is given by

$$S(x,y) = \sum_{i=1}^{n} c_i \phi_i = \sum_{i=1}^{n} c_i [(x - x_i)^2 + (y - y_i)^2 + b^2]^{1/2} \tag{22}$$

The cloud data set is substituted into Eq. (22), giving set of linear equations

$$z_i = \sum_{i=1}^{n} c_i [(x_j - x_i)^2 + (y_j - y_i)^2 + b^2]^{1/2} \quad j = 1, \ldots, n \tag{23}$$

NEURAL NETWORK AS UNIVERSAL APPROXIMATOR

Although classification is a very important form of neural computation, neural networks could be used to find an approximation of a multivariable function $F(x)$ (6). This could be approached through a supervised training of an input-output mapping from a data set. The learning proceeds as a sequence of iterative weight adjustments until a weight vector is found that satisfies certain criterion.

In a more formal approach, multilayer networks can be used to map $\Re^n$ into $\Re$ by using P examples of the function $F(x)$ to be approximated by performing nonlinear mapping with continuous neurons in the first layer, and then comput-

ing the linear combination by the single node of the output layer as follows:

$$y = \Gamma[VX] \tag{24}$$

$$O = W^t y \tag{25}$$

where V and W are the weight matrices for hidden and output layer respectively, and $\Gamma[\cdot]$ is a diagonal operator matrix consisting of nonlinear squashing functions $\phi(\cdot)$

$$\Gamma = \begin{bmatrix} \phi(\cdot) & 0 & 0 & \cdot & 0 \\ 0 & \phi(\cdot) & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \phi(\cdot) \end{bmatrix} \tag{26}$$

A function $\phi(\cdot): \Re \rightarrow [0, 1]$ is a squashing function if (1) it is nondecreasing, (2) $lim_{\lambda \rightarrow \infty}\ \phi(\lambda) = 1$, (3) $lim_{\lambda \rightarrow -\infty}\ \phi(\lambda) = 0$. Here we have used a bipolar squashing function of the form

$$\phi(x) = \frac{2}{1 + e^{-\lambda x}} - 1 \tag{27}$$

The studies of Funanashi (8), Hornik, Stinchcombe and White (7) prove that multilayer feedforward networks perform as a class of universal approximators. Although the concept of nonlinear mapping, followed by linear mapping, pervasively demonstrates the approximating potential of neural networks, the majority of the reported studies have dealt with second layer also providing the nonlinear mapping (6,7). The general network architecture performing the nested nonlinear scheme consists of a single hidden layer and a single output O such that

$$O = \Gamma(W\Gamma[VX]) \tag{28}$$

This standard class of neural networks architecture can approximate virtually any multivariable function of interest provided that sufficiently many hidden neurons are available.

Approximation Using Multilayer Networks

A 2-layer network was used for surface approximation. The x and y coordinates of the data points were the input to the network, while the function value $F(x, y)$ was the desired response d.

The learning algorithm applied was the error back-propagation learning technique. This technique calculates an error signal at the output layer and uses the signal to adjust network weights in the direction of the negative gradient descent of the network error E so that, for a network with I neurons in the input layer, J neurons in the hidden layer, and K neurons the output layer, the weight adjustment is as follows:

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}}, k = 1, 2, \ldots, K \quad j = 1, 2, \ldots J \tag{29}$$

$$\Delta v_{ji} = -\eta \frac{\partial E}{\partial v_{ji}}, j = 1, 2, \ldots, J \quad i = 1, 2, \ldots I \tag{30}$$

where

$$E = \frac{1}{2} \sum_{k=1}^{K} (d_k - O_k)^2 \tag{31}$$

The size J of the hidden layer is one of the most important considerations when solving actual problems using multilayer feedforward networks. The problem of the size choice is under intensive study, with no conclusive answers available thus far for many tasks. The exact analysis of the issue is rather difficult because of the complexity of the network mapping and the nondeterministic nature of many successfully completed training procedures (6). Here, we tested the network using different number of hidden neurons. The degree of accuracy reflected by mean square error was chosen to be 0.05. Results are provided later in this paper.

Approximation Using Functional Link Networks

Instead of carrying out a multistage transformation, as in multilayer networks, input/output mapping can also be achieved through an artificially augmented single-layer network. The concept of training an augmented and expanded network leads to the so-called functional link network as introduced by Pao (1989) (10). Functional link networks are single-layer neural networks that can handle linearly nonseparable tasks by using appropriately enhanced representation. This enhanced representation is obtained by augmenting the input by higher order terms that are generally nonlinear functions of the input.

The functional link network was used to approximate the surfaces by enhancing the 2-component input pattern (x, y) by 26 orthogonal components such as xy, $\sin(n\pi x)$, $\cos(n\pi y)$, *etc.* for $n = 1, 2, \ldots, m$. The output of the network can be expressed as follows:

$$\begin{aligned} F(x, y) = {} & xw_x + yw_y + xyw_{xy} \\ & + \sum_{i=1}^{m} \sin(i\pi x) w_{xi} + \cos(i\pi x) w_{xi} \\ & + \sum_{i=1}^{m} \sin(i\pi y) w_{yi} + \cos(i\pi y) w_{yi} \end{aligned} \tag{32}$$

The basic mathematical theory indicates that the functional expansion model should converge to a flat-net solution if a large enough number of additional independent terms are used.

HYPERSURFACE RECONSTRUCTION AS AN ILL-POSED PROBLEM

For the following reason, the strict interpolation procedure described here may be a poor strategy for training of function approximators (for certain classes) because of poor new data generalization: When the number of data points in the training set is much larger than the number of degrees of freedom of the underlying physical process, and we are constrained to have as many basis functions as data points, the problem is overdetermined. Consequently, the algorithm may end up fitting misleading variations because of noise in the input data, and thereby result in a degraded generalization performance (1).

The approximation problem belongs to a generic class of problems called *inverse problems.* An inverse problem may be *well posed* or *ill posed.* The term *well posed* has been used in applied mathematics since the time of Hadamard in the early 1900s. To explain what we mean by this term, assume that

we have a domain X and a range Y taken to be metric spaces, and that they are related by a fixed but unknown mapping F. The problem of reconstructing the mapping F is said to be well posed if three conditions are satisfied (1):

1. *Existence.* For every input vector $\boldsymbol{x} \in X$, there does exist an output $y = F(\mathbf{x})$, where $y \in Y$.
2. *Uniqueness.* For any pair of input vectors $\boldsymbol{x}, \boldsymbol{t} \in X$, we have $F(\boldsymbol{x}) = F(\boldsymbol{t})$ if, and only if, $\boldsymbol{x} = \boldsymbol{t}$.
3. *Continuity.* The mapping is continuous, that is for any $\epsilon > 0$ there exists $\delta = \delta(\epsilon)$ such that the condition $d_X(\boldsymbol{x}, \boldsymbol{t}) < \delta$ implies that $d_Y(F(\boldsymbol{x}), F(\boldsymbol{t})) < \epsilon$, where $d(\cdot, \cdot)$ is the distance between two arguments in their respective spaces.

If these conditions are not satisfied, the inverse problem is said to be *ill posed.* Function approximation is an ill posed inverse problem for the following reasons. First there is not as much information in the training data as we really need to reconstruct the input-output mapping uniquely, hence the uniqueness criterion is violated. Second, the presence of noise or imprecision in the input data adds uncertainty to the reconstructed input-output mapping in such a way that an output may be produced outside the range for a giving input inside the domain, in other words, when the continuity condition is violated.

Regularization Theory

Tikhonov (12) proposed a method called *regularization* for solving ill-posed problems. In the context of approximation problems, the basic idea of regularization is to stabilize the solution by means of some auxiliary nonnegative functional that embeds prior information, for example, smoothness constraints on the input-output mapping (that is, a solution to the approximation problem), and thereby make an ill-posed problem into a well-posed one (1,11).

According to Tikhonov's regularization theory, the function F is determined by minimizing a cost functional $\mathscr{E}(F)$, so called because it maps functions (in some suitable function space) to the real line. It involves two terms:

1. *Standard Error Term.* This first term, denoted by $\mathscr{E}_s(F)$, measures the standard error between the desired response d_i and the actual response y_i for training example $i = 1, 2, \ldots, N$. Specifically,

$$\begin{aligned} \mathscr{E}_s(F) &= \sum_{i=1} N(d_i - y_i)^2 \\ &= \sum_{i=1} N[d_i - F(\mathbf{x}_i)]^2 \end{aligned} \tag{33}$$

2. *Regularization Term.* This second term, denoted by $\mathscr{E}_c(F)$, depends on the geometric properties of the approximating function $F(\mathbf{x})$. Specifically, we write

$$\mathscr{E}_c(F) = \|\mathbf{P}F\|^2 \tag{34}$$

where $\mathbf{P}$ is a linear differential operator. Prior information about the form of the solution is embedded in the operator $\mathbf{P}$, which naturally makes the selection of $\mathbf{P}$ problem dependent. $\mathbf{P}$ is referred to as a stabilizer in the sense that it stabilizes the solution F, making it smooth and therefore continuous.

The analytical approach used for the situation described here draws a strong analogy between linear differential operators and matrices, thereby placing both types of models in the same conceptual framework. Thus the symbol $\|.\|$ denotes a norm imposed on the function space to which $\mathbf{P}F$ belongs. By a function space we mean a normed vector space of functions. Ordinarily, the function space used here is the L^2 space that consists of all real-valued functions $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathscr{R}^p$, for which $|f(\boldsymbol{x})|^2$ is Lebesgue integrable. The function $f(\boldsymbol{x})$ denotes the actual function that defines the underlying physical process responsible for the generation of the input-output pair. Strictly speaking, we require the function $f(\boldsymbol{x})$ to be a member of a reproducing kernel Hilbert space (RKHS) with a reproducing kernel in the form of the Dirac delta distribution (14). The simplest RKHS that satisfies the previously mentioned conditions is the space of rapidly decreasing, infinitely continuous differentiable functions, that is, the classical space S of rapidly decreasing test functions for the Shawrz theory of distributions, with finite $\mathbf{P}$-induced norm

$$H_p = \{f \in S : \|\mathbf{P}f\| < \infty\} \tag{35}$$

where the norm of $\mathbf{P}f$ is taken with respect to the range of $\mathbf{P}$, assumed to be another Hilbert space. The principal of regularization may now be stated as follows: Find the function F(x) that minimizes the cost functional $\mathscr{E}(F)$ defined by

$$\begin{aligned} \mathscr{E}(F) &= \mathscr{E}_s(F) + \lambda \mathscr{E}_c(F) \\ &= \sum_{i=1} N[d_i - F(\mathbf{x}_i)]^2 + \lambda \|\mathbf{P}F\|^2 \end{aligned} \tag{36}$$

where λ is a positive real number called regularization parameter.

Regularization Networks

Poggio et al., (13) suggested some form of prior information about the input-output mapping that would make the learning problem well posed so that the generalization to new data is feasible. They also suggested a network structure that they called *regularization network.* It consists of three layers. The first layer is composed of input (source) nodes whose number is equal to the dimension p of the input vector $\boldsymbol{x}$. The second layer is a hidden layer, composed of nonlinear units that are connected directly to all the nodes in the input layer. There is one hidden unit for each data point x_i, $i = 1, 2, \ldots, N$, where N is the number of training examples. The activation of the individual hidden units are defined by Green's function $G(x, x_i)$ given by (20)

$$G(x, x_i) = \exp\left[-\frac{1}{2\sigma_i^2} \sum_{k=1}^{p} (x_k - x_{i,k})^2\right] \tag{37}$$

This Green's function is recognized to be a multivariate Gaussian function. Correspondingly, the regularized solution takes on the following special form:

$$F(x) = \sum_{i=1}^{N} w_i \exp\left[-\frac{1}{2\sigma_i^2} \sum_{k=1}^{p} (x_k - x_{i,k})^2\right] \tag{38}$$

which consists of a linear superposition of multivariate Gaussian basis functions with center x_i (located at the data points) and widths σ_i (1).

The output layer consists of a single linear unit, and is fully connected to the hidden layer. The weights of the output layer are the unknown coefficients of the expansion, defined in terms of the Green's functions $G(x; x_i)$ and the regularization parameter λ by

$$w = (G + \lambda I)^{-1} d \tag{39}$$

This regularization network assumes that the Green's function $G(x; x_i)$ is positive definite for all i. Provided that this condition is satisfied, which it is in the case of the $G(x; x_i)$ having the Gaussian form, for example, then the solution produced by this network will be an optimal interpolant in the sense that it minimizes the functional $\mathscr{E}(F)$. Moreover, from the viewpoint of approximation theory, the regularization network has three desirable properties (17):

1. The regularization network is a universal approximator in that it can approximate arbitrarily well any multivariate continuous function on a compact subset $\mathscr{R}^p$, given a sufficiently large number of hidden units.
2. Since the approximation scheme derived from regularization theory is linear in the unknown coefficients, it follows that the regularization network has the best approximation property. This means that given an unknown nonlinear function f, there always exists a choice of coefficients that approximates f better than all other possible choices.
3. The solution computed by the regularization network is optimal. Optimality here means that the regularization minimizes a functional that measures how much the solution deviates from its true value as represented by the training data.

RESULTS

We now quantitatively compare the performance of the classic techniques shown in Section 1 with the neural network approaches using synthetic range data for four typical free-form surface patches suggested by Bradley and Vickers (3), and two surfaces suggested in this paper. The six test surfaces are:

1. Surface 1:

$$z = \sin(0.5x) + \cos(0.5y) \tag{40}$$

2. Surface 2:

$$z = \sin(x) + \cos(y) \tag{41}$$

3. Surface 3:

$$z = e^{-(x-5)^2+(y-5)^2/4} \tag{42}$$

4. Surface 4:

$$z = \tanh(x + y - 11) \tag{43}$$

5. Surface 5:

$$z = \text{Rectangular Box} \tag{44}$$

6. Surface 6:

$$z = \text{Pyramid} \tag{45}$$

The first two surfaces are bivariate sinusoidal functions, with Surface 2 having twice the spatial frequency of Surface 1. Many consumer items are composed of smoothly varying free-form patches similar to Surface 1, while Surface 2 is less common. Surface 3 has a peaked form, and Surface 4 is smooth with a sharp ridge diagonally across it. Surfaces 5 and 6 have

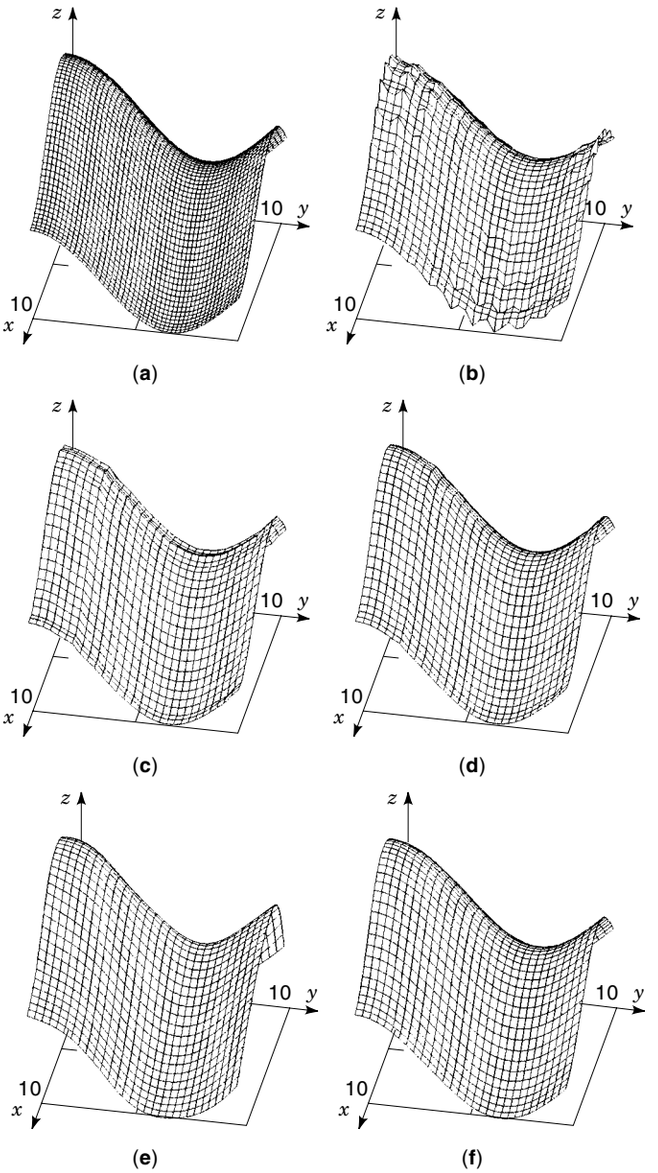


Figure 1. Comparison of reconstruction of Surface 1 using all methods: (a) original surface; (b) Shepard Interpolation; (c) Thin Plate B-spline; (d) Hardy's Multiquadric Interpolation; (e) MultiLayer Neural Network; and (f) Functional link Neural Network.

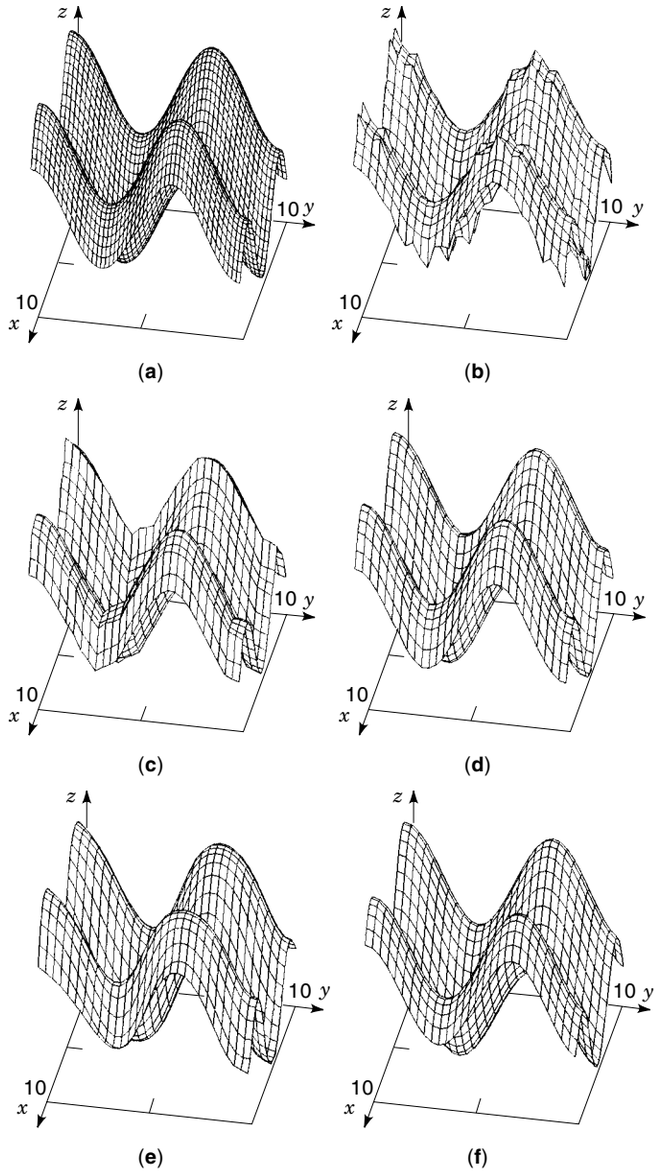


Figure 2. Comparison of reconstruction of Surface 2 using all methods: (a) original surface; (b) Shepard Interpolation; (c) Thin Plate B-spline; (d) Hardy's Multiquadric Interpolation; (e) MultiLayer Neural Network; and (f) Functional link Neural Network.

sharp edges and were included to test the modeling techniques on discontinuous surfaces. Data sets were generated, using the six surfaces, with each set consisting of 2500 points contained in a rectangular patch and with x and y varying from 0.0 to 10.0. Testing of these surface fitting methods has been done by local interpolation over 8×8 overlapping square regions. All methods were applied to a data set mesh of 900 points contained in the same rectangular patch as the original data set (9).

From the results, we can deduce that the Hardy's Multiquadratic Interpolation provides a large improvement over the Shepard's Interpolation and the Thin Plate Splines method for the first four surfaces, while performing the worst of the three methods for the sharp edge surfaces. The reconstructed surfaces, using all interpolation techniques, are shown in Figs. 1–6.

The interpolations using the neural networks perform exceptionally well on the first four surfaces, but because of the sharp edges in Surfaces 5 and 6, their performance was not as good. The networks seem to have difficulty with sharp transitions and discontinuities. A method for dealing with this difficulty could be to use a block training technique in which the neural network learns the surfaces in smaller patches. This should decrease learning time and make the training cycles less complicated, although it creates the need for local regions. During training, the weights are updated for each training point; the error for that training point and corresponding weight set is then calculated. When using a large number of training pairs, the weights are significantly changed from the beginning to the end of the training cycle. Therefore, the calculated error is not equivalent to the true error, which is based on this final weight set. This is because

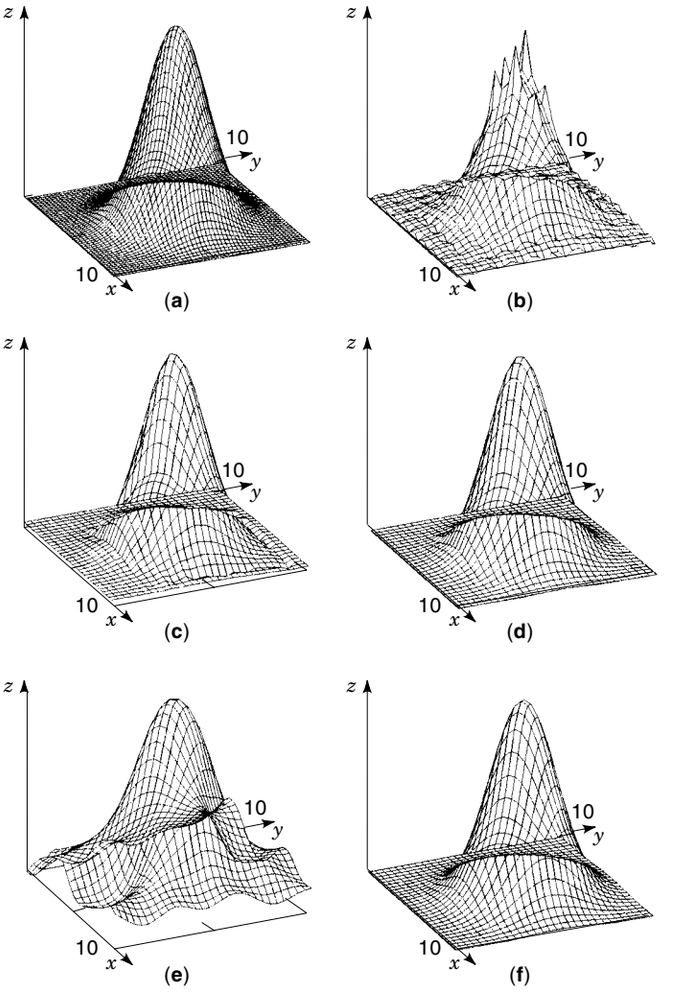


Figure 3. Comparison of reconstruction of Surface 3 using all methods: (a) original surface; (b) Shepard Interpolation; (c) Thin Plate B-spline; (d) Hardy's Multiquadric Interpolation; (e) MultiLayer Neural Network; and (f) Functional link Neural Network.

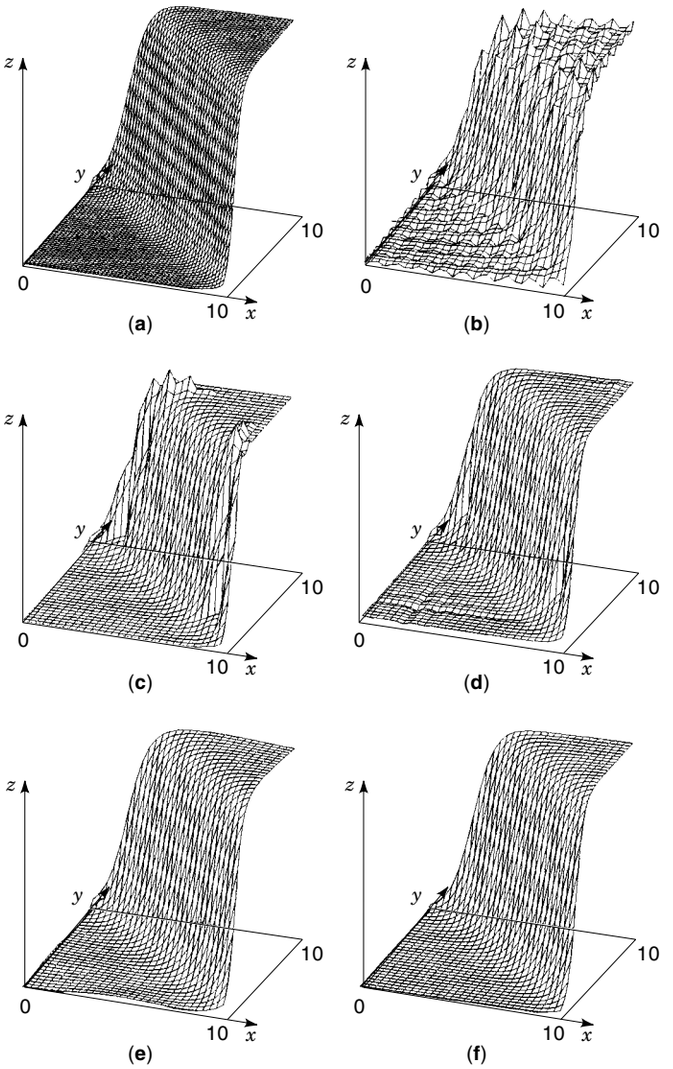


Figure 4. Comparison of reconstruction of Surface 4 using all methods: (a) original surface; (b) Shepard Interpolation; (c) Thin Plate B-spline; (d) Hardy's Multiquadric Interpolation; (e) MultiLayer Neural Network; and (f) Functional link Neural Network.

each error calculation has been made with a different weight set. A correction for this can be done by simply recalculating the true error using the final weight set at the end of each training cycle.

It is clear that the neural networks method produces smooth surfaces as compared with those produced by Hardy's Multiquadratic Interpolation without the need of constructing local regions. In general, the neural networks approach is far superior in terms of the ease of implementation. The results also show the potential of Functional Link Neural Network as an approximator; it is easier to implement than the MultiLayer Neural Network and faster to converge (9). However, neither of the two networks performed well on Surface 6, possibly because of the presence of two discontinuities.

A new approach for free-form surface modeling is to combine neural networks with classical techniques to create a new hybrid interpolation method. The surface can be divided into smoothly varying portions and areas of sharp transitions. The neural network could then be used to approximate these smoothly varying areas, while a classical technique like Hardy's Multiquadric Interpolation could be used on the areas of sharp transition.

APPLICATIONS

Recently, laser range finders (also known as 3-D laser scanners) have been employed to scan multiple views of an object. The scanner output is usually an unformatted file of large size (known as the "cloud of data") (3,31). In order to use the cloud of data for surface reconstruction, a registration technique is implemented that makes correspondence with the actual surface. Laser scanners are convenient in applications where the object is irregular but in general smooth. The corre-

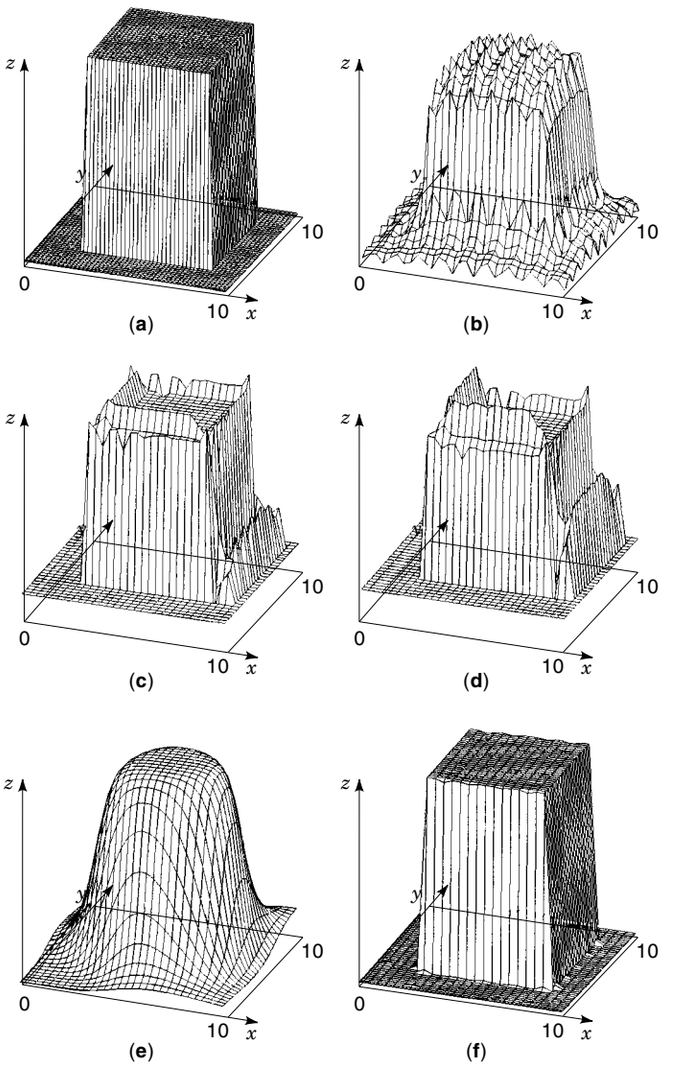


Figure 5. Comparison of reconstruction of Surface 5 using all methods: (a) original surface; (b) Shepard Interpolation; (c) Thin Plate B-spline; (d) Hardy's Multiquadric Interpolation; (e) MultiLayer Neural Network; and (f) Functional link Neural Network.

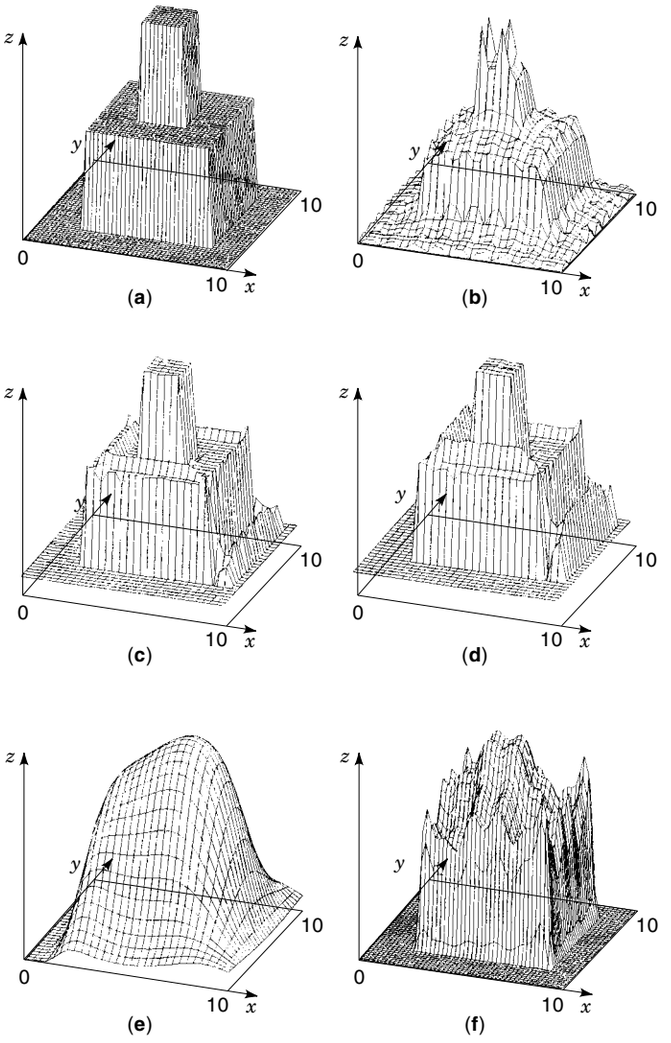


Figure 6. Comparison of reconstruction of Surface 6 using all methods: (a) original surface; (b) Shepard Interpolation; (c) Thin Plate B-spline; (d) Hardy's Multiquadric Interpolation; (e) MultiLayer Neural Network; and (f) Functional link Neural Network.

sponding surfaces of these objects are denoted by "free-form surfaces." As Besl (31) stated, a free-form surface is not constrained to be piecewise planar, piecewise quadratic, piecewise superquadratic, or cylindrical. However, the free-form surface is smooth in the sense that the surface normal is well defined and continuous everywhere (except at isolated cusps, vertices, and edges). Common examples of free-form surface shapes include human faces, cars, airplanes, boats, clay models, sculptures, and terrain.

Historically, free-form shape matching using 3-D data was done earliest by Faugeras and his group at INRIA (33), where they demonstrated effective matching with a Renault auto part (steering knuckle) in the early 1980s. This work popularized the use of quaternions for least squares registration of corresponding 3-D point sets in the computer vision community. The alternative use of the singular value decomposition (SVD) algorithm was not as widely known at that time. The primary limitation of this work was that it relied on the probable existence of reasonably large planar regions within a free-form shape.

There are a number of studies dealing with global shape matching or registration of free-form surfaces on limited classes of shapes. For example, there have been a number of studies on point sets with known correspondence (31–35). Bajcsy (34) is an example of studies on polyhedral models and piecewise models.

As we indicated previously, the output of the laser scanner (the cloud of data) is in the form of a sparse, unformatted file. The goal is to build a model of the physical surface using this data. Two issues need to be examined: (1) how to fit a surface using the data from a single view; and (2) how to merge the data from multiple views to build an overall 3-D model for the object. Bradley and Vickers (3), surveyed a number of algorithms for surface reconstruction using the cloud of data of one viewpoint. Results were shown on basic surfaces like sinusoids and exponentials. They also suggested an algorithm for surface modeling based on the following steps: (1) divide the cloud of data into meshes of smaller sizes; (2) fit a surface using a subset of data points on each mesh; and (3) merge the surfaces of the meshes together. This approach has been shown to be convenient for simple surfaces with little or no occlusion.

BIBLIOGRAPHY

1. S. Haykin, *Neural Networks: A Comprehensive Foundation,* Indianapolis, IN: Macmillan College, 1994.
2. D. Shepard, A two-dimensional interpolation function for irregularly spaced data, *Proc. ACM Nat. Conf.,* 1964, pp. 517–524.
3. C. Bradley and G. Vickers, Free-form surface reconstruction for machine vision rapid prototyping, *Opt. Eng.,* **32** (9): 2191–2200, September 1993.
4. B. Bhanu and C. C. Ho, CAD-based 3-D object representation for robot vision, *IEEE Comput.,* **20** (8): 19–36, 1987.
5. R. L. Hardy, Multiquadratic equations of topography and other irregular surfaces, interpolation using surface splines. *J. Geophys.,* **76**: 1971.
6. J. Zurada, *Introduction to Artificial Neural Systems,* St. Paul, MN: West, 1992.
7. K. Hornik and M. Stinchombe, Multilayer feedforward networks as universal approximators, *Neural Networks,* 359–366, 1989.
8. K. I. Funanashi, On the approximate realization of continuous mappings by neural networks, *Neural Networks,* 183–192, 1989.
9. M. N. Ahmed and A. A. Farag, Free form surface reconstruction using neural networks, *Proc. of ANNIE'94,* **1**: 51–56, 1994.
10. P. H. Pao, *Adaptive Pattern Recognition and Neural Networks,* New York: Addison Wesley, 1989.
11. R. Courant and D. Hilbert, *Methods and Mathematical Physics,* vols. 1 and 2, New York: Wiley, 1970.
12. A. N. Tikhonov, On solving incorrectly posed problems and method of regularization, *Doklady Akademii Nauk USSR,* **151**: 501–504, 1963.
13. T. Poggio and F. Girosi, Networks for approximation and learning, *Proc. IEEE,* **78**: 1481–1497, 1990.
14. T. Poggio, A theory of how the brain might work, *Cold Spring Harbor Symp. Quantitative Biol.,* **55**: 899–910, 1990.
15. T. Poggio and S. Edelman, A network that learns to recognize three-dimensional objects, *Nature,* (London) **343**: 263–266, 1990.

16. T. Poggio and F. Girosi, Regularization algorithms for learning that are equivalent to multilayer networks, *Science,* **247**: 978–982, 1990.
17. T. Poggio, V. Torre, and C. Koch, Computational vision and regularization theory, *Nature* (London) **317**: 314–319, 1985.
18. D. S. Broomhead and D. Lowe, Multivariable functional interpolation and adaptive networks, *Complex Systems,* **2**: 321–355, 1988.
19. D. S. Broomhead, D. Lowe, and A. R. Webb, A sum rule satisfied by optimized feedforward layered networks, *RSRE Memorandum No. 4341, Royal Signals and Radar Establishment,* Malvern, UK, 1989.
20. M. J. D. Powell, Radial basis functions for multivariate interpolation: a review, *IMA Conf. Algorithms Approximation Functions Data, RMCS,* Shivenham, UK, 1985, pp. 143–167.
21. M. J. D. Powel, Radial basis function approximations to polynomials, *Numerical Analysis 1987 Proc.,* Dundee, UK, 1988, pp. 223–241.
22. D. Lowe, Adaptive radial basis function nonlinearities, and the problem of generalization, *1st IEE Int. Conf. Artificial Neural Networks,* London, UK, 1989, pp. 171–175.
23. D. Lowe, On the iterative inversion of RBF networks: a statistical interpretation, *2nd IEE Int. Conf. Artificial Neural Networks,* Bournemouth, UK, 1991, pp. 29–33.
24. D. Lowe and A. R. Webb, Adaptive networks, dynamical systems, and the predictive analysis of time series, *1st IEE Int. Conf. Artificial Neural Networks,* London, UK, 1989, pp. 95–99.
25. F. Girosi and T. Poggio, Representative properties of networks: Kolmogorov's theorem is irrelevant, *Neural Comput.,* **1**: 465–469, 1989.
26. F. Girosi and T. Poggio, Networks and best approximation property, *Biol. Cybern.,* **63**: 169–176, 1990.
27. C. N. Dorny, *A Vector Space Approach to Models and Optimization,* New York: Wiley, 1975.
28. P. Simard and Y. LeCun, Reverse TDNN: An architecture for trajectory generation, *Advances Neural Inf. Process. Syst.* **4**: 579–588, 1992.
29. A. N. Tikhonov, On regularization of ill-posed problems, *Doklady Akadermii Nauk USSR,* **153**: 49–52, 1973.
30. A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems,* Washington, DC: Winston, 1977.
31. P. J. Besl, Free-form Surface Matching, in H. Freeman (ed.), *Machine Vision for Three-Dimensional Scenes,* San Diego: Academic Press, 1990, pp. 25–71.
32. P. Besl and D. McKay, A method for registration of 3-D shapes, *IEEE Trans. Pattern Anal. Mach. Intel.,* **PAMI-14**: 239–256, Feb. 1992.
33. O. D. Faugeras and M. Herbert, The representation, recognition, and locating of 3-D objects, *Int. J. Robotics Res.,* **5** (3): 27–52, 1986.
34. R. Bajcsy and F. Solina, Three-dimensional object representation, *Proc. 1st. Int. Conf. Comput. Vision,* (London), June 8–11, 1989, pp. 231–240.
35. D. Terzopoulos et al., Elastically deformable models, *Comput. Graphics,* **21** (4): 205–214, 1987.
36. R. Franke, Scattered data interpolation, *Math Comput.,* **38**: 181–200, 1982.

MOHAMED N. AHMED
SAMEH M. YAMANY
ALY A. FARAG
University of Louisville

FUNCTIONS, BESSEL. See BESSEL FUNCTIONS.
FUNDING RESEARCH. See RESEARCH INITIATIVES.
FUSES, ELECTRIC. See ELECTRIC FUSES.
FUSE TRANSFORMER PROTECTION. See TRANSFORMER PROTECTION.

Gaussian Filtered Representations of Images

Standard Article

S. Ravela[1] and R. Manmatha[1]

[1]University of Massachusetts, Amherst, MA

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2418

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (575K)

Abstract

The sections in this article are

The Gaussian Filtered Representation of Images

Matching Affine Deformed Images

Image Retrieval

Acknowledgment

| | |

Copyright © 1999-2008 All Rights Reserved.

GAUSSIAN FILTERED REPRESENTATIONS OF IMAGES

Gaussian filtered representations of images have been used to address several important visual tasks. In early work Marr and Hildreth used them to attenuate noise and detect edges (1). Specific operators such as the Laplacian of Gaussians and the difference of Gaussians have been used for multiresolution analysis of images (2,3). More recently, several researchers (4,5,6,7,8) have shown that Gaussian derivatives may be used to robustly represent the local image structure at multiple scales. In this article, the Gaussian derivative filter and its spatial- and frequency-domain properties are examined, and it is used to create a description of the local intensity surface. Motivated by its multiscale properties, Gaussian filtered representations are constructed to address two specific problems.

The first problem considered is of matching images that are affine deformed versions of each other. Solutions to this problem form an important component for several applications such as video mosaicking, registration, object recognition, structure from motion, and shape from texture. In particular, consider an example where successive views of a scene are observed in a video. These views will be deformed versions of each other and under certain circumstances may be approximated using an affine transformation. Gaussians and their derivatives are used to recover affine deformations. Consider two patches related by an affine transform. If the patches are filtered using Gaussian filters, the outputs are equal provided the Gaussian is affine-transformed in the same manner as the function. Thus, one can construct a solution that minimizes the error with respect to the affine parameters, where the error is defined between corresponding Gaussian derivative filter outputs. The affine matching problem is discussed in detail in the section after next.

The second focus of this article is in addressing the problem of image retrieval. Advances in computational power and the rapid increase in performance-to-cost ratio of most compu-

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

tational devices has led to the acquisition and storage of pictorial information on digital media. One of the important challenges that this trend presents is the development of algorithms for managing digitally stored pictorial information. While machine-stored text can be searched using any of several text search engines, there are as yet no good tools available to search and manage image collections.

The reason that *image retrieval* has proven to be hard is that users expect the system to find relevant images based on some personal or cultural semantics. Representing semantics is hard and requires solutions to problem such as automatic feature detection, segmentation, and recognition. These problems are as yet unsolved. However, in certain cases, many image attributes such as color, texture, shape, and "appearance" are often directly correlated with the semantics of the problem. For example, logos or product packages (e.g., a box of Tide) have the same color wherever they are found. The coat of a leopard has a unique texture, and Abraham Lincoln's appearance is distinctive. These image attributes can often be used to index and retrieve images.

One such approach is to exploit the structure of the image intensity surface. In recent work Ravela and Manmatha (9) have shown that representations of the intensity surface may be used to retrieve objects that appear visually similar. Arguably an object's visual appearance in an image is closely related to several factors, including, among others, its three-dimensional shape, albedo, and surface texture and the image viewpoint. It is nontrivial to separate the different factors constituting an object's appearance. For example, the face of a person has a unique appearance that cannot just be characterized by the geometric shape of the "component parts."

We argue that Gaussian filtered representation of images can be used for retrieval by appearance. A paradigm for retrieval that is used widely, and is adopted here, is that images in the database are processed and described by a set of feature vectors. These vectors are indexed ahead of time. During run time, a query is provided in the form of an example image, and its features are compared with those stored. Images are then retrieved in the order indicated by the comparison operator. In this work, feature vectors are constructed using responses to Gaussian derivatives filters at multiple scales. Using this approach, it is shown that whole images or parts thereof can be retrieved. This flexibility is important because a user interacting with an image retrieval system might be interested in the image as a whole, such as a trademark, or in only a part of the image, such as a face within a scene. In the former case the representation must capture the appearance of the whole image, and the similarity is global. In the latter case, the representation must allow for local similarity.

The remainder of this article is organized as follows. The next section provides a review of the Gaussian filter, some key properties and derives the features that will be used subsequently to address the affine image matching and retrieval tasks. In the subsequent section matching of images under an affine deformation is considered, and in the last section image retrieval by appearance is discussed.

THE GAUSSIAN FILTERED REPRESENTATION OF IMAGES

This section begins by examining the spatial and frequency characteristics of the Gaussian filter. Then, the role of the Gaussian filter in providing a robust representation of the intensity surface is discussed. The section ends with a discussion of how to implement discrete versions of Gaussians and their derivatives.

The Gaussian Filter: Preliminaries

The Gaussian and Its Derivatives. The isotropic normalized Gaussian in two dimensions is a C^∞ smooth function, defined as

$$G(\boldsymbol{p}, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\boldsymbol{p}^{\mathrm{T}}\boldsymbol{p}/2\sigma^2} \tag{1}$$

where $\boldsymbol{p} = \langle x, y \rangle \in R^2$, and $\sigma \in R$ is referred to as the scale. The derivatives of the normalized Gaussian are defined to arbitrary order. The nth derivative of the Gaussian (in two dimensions) is written in tensor form as

$$G_{i_1 \ldots i_n}(\cdot, \sigma) = \frac{\delta^n G(\cdot, \sigma)}{\delta i_1 \cdots \delta i_n} \tag{2}$$

where the free variables $i_1, \ldots, i_n$ cycle through all the degrees of freedom (x, y). Thus the first derivative is written as G_{i_1}, which in the Cartesian frame is G_x and G_y.

Filtering. In the spatial domain, the discrete two-dimensional (2D) image $Z(\boldsymbol{p})$ filtered with the Gaussian is expressed as the discrete convolution, $I(\boldsymbol{p}, \sigma) = (Z \bigstar G)(\boldsymbol{p}, \sigma)$ the Gaussian has infinite extent. Since discrete images are typically of finite extent, the truncation of the Gaussian extent is considered the subsection "Implementation." In the case of Gaussian derivatives, since differentiation commutes with convolution, the following expression may be written:

$$\begin{aligned} I_{i_1 \ldots i_n}(\boldsymbol{x}, \sigma) &= (Z \star G)_{i_1 \ldots i_n}(\boldsymbol{x}, \sigma) \\ &= (Z_{i_1 \ldots i_n} \star G)(\boldsymbol{x}, \sigma) = (Z \star G_{i_1 \ldots i_n})(\boldsymbol{x}, \sigma) \end{aligned} \tag{3}$$

Frequency-Domain Interpretation. The Fourier transform of the 2-D Gaussian defined in Eq. (1) is written as

$$\mathcal{G}(\boldsymbol{u}, \sigma) = e^{-\sigma^2 \boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}/2} \tag{4}$$

where $\boldsymbol{u} = \langle u_x, u_y \rangle \in R^2$ is the 2D frequency variable. Similarly, the Fourier transform of the nth derivative of the Gaussian defined in Eq. (2) is defined as

$$\mathcal{G}^{(i_1 \ldots i_n)}(\boldsymbol{u}, \sigma) = j^n (u_{i_1} \ldots u_{i_n}) \mathcal{G}(\boldsymbol{u}, \sigma) \tag{5}$$

Here, $i_1, \ldots, i_n$ are free variables that can be identified with any of the Cartesian degrees of freedom, and $j = \sqrt{-1}$. For example, the Fourier transform of the second mixed derivative G_{xy} is $\mathcal{G}^{(i_1 i_2)}(u, \sigma) = -u_x u_y \mathcal{G}(\boldsymbol{u}, \sigma)$, where the substitution $i_1 = x$ and $i_2 = y$ is made. In a Cartesian coordinate system the Fourier transform of the nth Gaussian derivative may be written as

$$\mathcal{G}^{(i_1 \ldots i_n)}(\boldsymbol{u}, \sigma) = \mathcal{G}^{(x^p y^q)}(\boldsymbol{u}, \sigma) = j^{p+q} (u_z^p u_y^q) \mathcal{G}(\boldsymbol{u}, \sigma) \tag{6}$$

for some integers $p, q \geq 0$, $p + q = n$, and then p free variables are instantiated to x and q to y.

From the above two definitions, using composition, one may immediately observe the following properties:

Cartesian Separability. The Fourier transform of the nth Gaussian derivative may be expressed as the composition of one-dimensional (1-D) filters:

$$\begin{aligned} j^{p+q}(u_x^p u_y^q)\mathcal{G}(\boldsymbol{u},\sigma) &= j^p u_x^p e^{-\sigma^2 u_x^2/2}\cdot j^q u_x^q e^{-\sigma^2 u_y^2/2} \\ &= \mathcal{H}_x^{(p)}(u_x,\sigma)\cdot\mathcal{H}_y^{(q)}(u_y,\sigma) \end{aligned} \tag{7}$$

Since composition implies convolution in the spatial domain, one can implement the nth Gaussian derivative using separable 1-D convolutions.

Cascade Property. The cascade property of Gaussian derivatives may be observed from the composition of two filters in the frequency domain:

$$\mathcal{G}^{(x^p y^q)}(\boldsymbol{u},\sigma_1)\cdot\mathcal{G}^{(x^m y^n)}(\boldsymbol{u},\sigma_2) = \mathcal{G}^{(x^{p+m} y^{q+n})}(\boldsymbol{u},\sqrt{\sigma_1^2+\sigma_2^2}) \tag{8}$$

Thus, filtering a signal successively with several Gaussian filters of scales $\sigma_1, \ldots, \sigma_n$ is equivalent to filtering the signal with a single Gaussian filter of scale $\sigma = \sqrt{\sigma_1^2 + \cdots + \sigma_n^2}$.

Center Frequency and Bandwidth. The Gaussian filter is a low-pass filter, and the derivatives are bandpass filters. In what follows, the center frequency and bandwidth of the nth derivative of the (spatial) Gaussian are derived. For the sake of simplicity 1-D Gaussians are considered.

The Fourier transform of the nth derivative of a 1D Gaussian G_{x^n} is written as

$$\mathcal{G}^{(x^n)}(u,\sigma) = j^n u^n e^{-\sigma^2 u^2/2}$$

Differentiating with respect to u and computing the extremum, one obtains the center frequency u_0:

$$\begin{aligned} \frac{d\mathcal{G}^{(x^n)}(u,\sigma)}{du} &= (n-u^2\sigma^2)j^n u^{n-1} e^{-\sigma^2 u^2/2} = 0 \\ u_0 &= \pm\frac{\sqrt{n}}{\sigma} \end{aligned} \tag{9}$$

It should be noted that the 1D function $\mathcal{G}^{(x^n)}$ is unimodal in either half plane and has a Gaussian envelope. It is odd and imaginary for odd-order spatial derivatives, and even and real for even-order spatial derivatives. This equation states that the center frequency is coupled both to the bandwidth and to the order of the derivative. As the order of the derivative increases, so does its center frequency, and therefore, higher derivatives enhance higher levels of spatial detail.

There are several ways to define the bandwidth of the filter. Here we adopt the equivalent-rectangular-bandwidth formulation (10). This formulation equates the area under the power spectrum of the filter with that of an equivalent ideal filter of width W and height equal to the peak instantaneous power of the filter. Therefore, one may write

$$W \times |(ju_0)^n e^{-\sigma^2 u_0^2/2}|^2 = \int_0^\infty |(ju)^n e^{-\sigma^2 u^2/2}|^2\, du \tag{10}$$

After some manipulation the solution to this relation can be shown to be the following:

$$W = W(n,\sigma) = \frac{e^n \pi}{4\sigma n^n}\prod_{i=1}^{i=n}(i-\tfrac{1}{2}), \qquad n \geq 1 \tag{11}$$

$$W(0,\sigma) = \frac{\sqrt{\pi}}{\sigma} \tag{12}$$

The above expressions show that the center frequency and bandwidth of the Gaussian and its derivatives are related to the order and scale of the derivative in the spatial domain. The Gaussian is a low-pass filter, filter while its derivatives are band-pass filters.

Noise Attenuation. Since the Gaussian derivatives are band-pass filters, they may be used to attenuate noise. In particular, consider a 1D function $Z(x) = P(x) + \epsilon \sin(\omega_1 x)$. The Fourier transform of $z(x)$ is

$$\mathcal{Z}(u) = \mathcal{P}(u) + N(u)$$

where

$$N(u) = j\pi[\delta(u+w_1) - \delta(u-w_1)]$$

Consider the application of the nth derivative of a 1D Gaussian to $Z(x)$ using the right-hand side (RHS) of Eq. (3), that is, $I_n(x,\sigma) = Z(x) \bigstar G_n(x,\sigma)$. The Fourier transform of $I_n(x,\sigma)$ is

$$\begin{aligned} \mathcal{I}^{(n)}(u,\sigma) &= \mathcal{G}^{(n)}(u,\sigma)\mathcal{P}(u) + \mathcal{G}^{(n)}(u,\sigma)N(u) \\ &= \cdots + j^n u^n e^{-\sigma^2 u^2/2} N(u) \end{aligned} \tag{13}$$

where $\mathcal{G}^{(n)}$ is the Fourier transform of the nth Gaussian derivative. The second term on the RHS of Eq. (13) can be made arbitrarily small by choosing an appropriate σ, eliminating the noise in the function Z. However, this also implies that the signal $\mathcal{P}(u)$ will be band-limited. Higher frequencies will get attenuated.

Representation of the Intensity Surface

The local spatial structure of the intensity surface can be approximated using the local spatial derivatives of the surface. This can be seen from the Taylor series expansion of the intensity surface. The Taylor expansion in the neighborhood of a point in the image will fully describe the local intensity surface up to the order to which the series is constructed. Consider an image I at point $\boldsymbol{p} = \langle x, y\rangle$. The value at a location $\boldsymbol{p} + \Delta\boldsymbol{p}$ can be estimated using the Taylor series expansion (written in tensor form):

$$I(\boldsymbol{p}+\Delta\boldsymbol{p}) \approx \sum_{n=0}^{N}\frac{1}{n!}\left(\Delta\boldsymbol{p}_{i_1}\cdots\Delta\boldsymbol{p}_{i_n}\frac{\delta^n I(\boldsymbol{p})}{\delta i_1\cdots\delta i_n}\right)$$

Each term i_j ($j = 1, \ldots, n$) is substituted for all the degrees of freedom, which in the case of a 2D image is two. Up to

order two, the expansion becomes

$$\begin{aligned} I(\boldsymbol{p}+\Delta\boldsymbol{p}) &= \sum_{n=0}^{2}\frac{1}{n!}\left(\Delta\boldsymbol{p}_{i_1}\cdots\Delta\boldsymbol{p}_{i_n}\frac{\delta^n I(\boldsymbol{p})}{\delta i_1\cdots\delta i_n}\right)\\ &= I(\boldsymbol{p})+\Delta\boldsymbol{p}_{i_1}\frac{\partial I(\boldsymbol{p})}{\partial i_1}+\frac{1}{2!}\left(\Delta\boldsymbol{p}_{i_1}\Delta\boldsymbol{p}_{i_2}\frac{\delta^2 I(\boldsymbol{p})}{\delta i_1\delta i_2}\right)\\ &= I(\boldsymbol{p})+\Delta\boldsymbol{p}_x\frac{\delta I(\boldsymbol{p})}{\delta x}\\ &\quad+\frac{1}{2!}\left(\Delta\boldsymbol{p}_x^2\frac{\delta^2 I(\boldsymbol{p})}{\delta x^2}+\Delta\boldsymbol{p}_x\Delta\boldsymbol{p}_y\frac{\delta^2 I(\boldsymbol{p})}{\delta x\delta y}\right)\\ &\quad+\Delta\boldsymbol{p}_y\frac{\delta I(\boldsymbol{p})}{\delta h}+\frac{1}{2!}\left(\Delta\boldsymbol{p}_y\Delta\boldsymbol{p}_x\frac{\delta^2 I(\boldsymbol{p})}{\delta y\delta x}+\Delta\boldsymbol{p}_y^2\frac{\delta^2 I(\boldsymbol{p})}{\delta y^2}\right) \end{aligned}$$

Rearranging the terms, we get the more familiar Cartesian form of the Taylor series (again up to order two):

$$\begin{aligned} I(\boldsymbol{p}+\Delta\boldsymbol{p}) &= I(\boldsymbol{p})+\Delta\boldsymbol{p}_x\frac{\partial I(\boldsymbol{p})}{\delta x}+\Delta\boldsymbol{p}_y\frac{\partial I(\boldsymbol{p})}{\delta y}\\ &\quad+\frac{1}{2!}\left(\Delta\boldsymbol{p}_x^2\frac{\delta^2 I(\boldsymbol{p})}{\delta x^2}+2\,\Delta\boldsymbol{p}_x\Delta\boldsymbol{p}_y\frac{\delta^2 I(\boldsymbol{p})}{\delta x\,\delta y}\right.\\ &\quad\left.+\Delta\boldsymbol{p}_y^2\frac{\delta^2 I(\boldsymbol{p})}{\delta y^2}\right) \end{aligned}$$

The above equation states that in order to estimate the intensity in the neighborhood of a point $\boldsymbol{p}$ the derivatives at $\boldsymbol{p}$ must be known. Therefore, it can be argued that spatial derivatives may be used to approximate the local intensity surface.

In the case of digital images, which are 2-D discrete functions of finite range, derivatives may be approximated using finite-difference operators. However, while finite differences can be computed, their outputs must be meaningful, or *well conditioned,* in the presence of noise. In the preceeding subsection it is shown that adding a high-frequency, low-amplitude noise may make the derivatives unstable. In discrete images this will result in noisy measurements.

A solution to the problem lies in the fact that the derivatives of a possibly discontinuous function become well conditioned if it is first convolved with the derivative of a smooth (C^∞) test function (8). The Gaussian is a smooth test function, and therefore the derivatives of the smoothed image $I(\boldsymbol{x},\sigma)=(Z\star G)(\boldsymbol{x},\sigma)$, $\boldsymbol{x}\in R^2$, are well conditioned for some value of σ. Another way of observing this is from the noise attenuation property presented in the preceding subsection.

The operational scheme for computing local structure at a given scale of observation σ is as follows. Each image is filtered with Gaussian derivatives (at a certain scale) to the order to which the local structure is desired to be approximated. Therefore, each pixel is associated with a set of derivatives that completely define the Taylor expansion to the desired order. Koenderink and van Doorn (5) have advocated the use of this representation and call it the local N-jet. The local N-jet of $I(\boldsymbol{x})$ at scale σ and order N is defined as the set

$$\mathcal{J}^N[I](\boldsymbol{x},\sigma)=\{I_{i_1\cdots i_n,\sigma}|n=0,\ldots,N\} \tag{14}$$

Observe that $\lim_{N\to\infty}\mathcal{J}^N[I](\boldsymbol{x},\sigma)$ bundles all the derivatives required to fully reconstruct the surface I_σ in a locality around $\boldsymbol{x}$ at a particular scale. This is the primary observation that is used to characterize local structure. That is, up to any order the derivatives locally approximate the regularized intensity surface. As a practical example consider the local 2-jet of an image $I(\boldsymbol{p})$, $\boldsymbol{p}=\langle x,y\rangle\in R^2$, at scale σ ($I_{yx}=I_{xy}$ and is therefore dropped):

$$\mathcal{J}^2[I](\boldsymbol{p},\sigma)=\{I,I_x,I_y,I_{xx},I_{xy},I_{yy}\}(\boldsymbol{p},\sigma)$$

The image I is filtered with the first two Gaussian derivatives (and the Gaussian itself) in both x and y directions. Point $\boldsymbol{p}$ is therefore associated with a *derivative feature vector* of responses at scale σ.

Multiscale Representation and Scale Space. The derivative feature vector is computed at a single scale, and therefore constitutes observations of the intensity surface at a fixed bandwidth. Equivalently, in the spatial domain, the intensity surface is observed at a fixed window size. In effect, the derivative feature vector constitutes observations, not of the original image, but of a smoothed version of it. Therefore, computing derivatives at a single scale is not likely to be a robust representation of local structure. Fundamentally, this is because the local structure of the image depends on the scale at which it is observed. An image will appear different at different scales. For example, at a small scale the texture of an ape's coat will be visible. At a large enough scale, the coat will appear homogeneous.

A better characterization of local structure is obtained by computing derivatives at several scales of observation. In the frequency domain this amounts to sampling the frequency spectrum of the original image using several bandwidths (scales) around multiple center frequencies (derivatives). In the spatial domain, it may be viewed as computing local derivatives at several neighborhoods around a point.

The Gaussian forms a very attractive choice for a multiscale operator, for several reasons. First, it is naturally defined with respect to a continuous scale parameter. Second, under certain conditions (4) it uniquely generates the linear scale space of an image.

The term *scale space* was introduced by Witkin (11) to describe the evolution of image structure over multiple scales. Starting from an original image, successively smoothed images are generated along a scale dimension. In this regard several researchers (4,6–8) have shown that the Gaussian uniquely generates the linear scale space of the image when it is required that structures present at a coarser scale must already be present at a finer scale. That is, no new structures must be introduced by the operator used to generate the scale space. Typically, these structures are the zero crossings or local maxima of the image intensity. This is a very significant result, because it provides a formal mechanism to represent multiscale information using a well-defined operator.

More formally, the scale space of an image $Z(\boldsymbol{p})$, $\boldsymbol{p}=\langle x,y\rangle\in R^2$, may be written as the one-parameter family of derived images obtained using the Gaussian operator G:

$$I(\boldsymbol{p},\sigma)=Z(\boldsymbol{p})\star G(\boldsymbol{p},\sigma)$$

The linear scale-space representation models an important physical observation. As an object moves away from a camera (in depth), its image appears less structured and finer contrasts get blurred. The change in intensity in a locality around a pixel that occurs with changing distance is accu-

rately represented in the scale-space trajectory of that pixel. A detailed analysis deriving the Gaussian as the unique linear scale-space operator is beyond the scope of this article. For an in-depth study the reader is pointed to Florack's dissertation (8).

From a practical perspective, the Gaussian allows local structure to be computed at several scales of observation that are related in a precise manner. Local structure as represented by the spatial derivatives may be computed directly across scales without explicitly computing the scale space. This may be seen from Eq. (3). In fact, Lindeberg (6) shows that the scale space is well defined for the Gaussian derivatives as well. Therefore, the Gaussian filtered representation is useful in at least two ways. First, it allows for the stable and efficient computation of local structure. Second, it is the only way to generate the linear scale space.

An argument is therefore made for a *multiscale feature vector* that describes the intensity surface locally at several scales. From an implementation standpoint a multiscale feature vector at a point p in an image I is simply the vector

$$J^N_{(\sigma_1 \ldots \sigma_k)}[I] = \{J^N[I](\boldsymbol{p}, \sigma_1), J^N[I](\boldsymbol{p}, \sigma_2), \ldots, J^N[I](\boldsymbol{p}, \sigma_k)\} \tag{15}$$

for some order N and a set of scales $\sigma_1, \ldots, \sigma_k$.

A natural question that arises in building representations for various applications is the parametrization required for the multiscale feature vector—that is, the number of scales to be used, their spacing, and the number of orders to be considered. In the applications described in this paper, the scales are placed half an octave ($\sqrt{2}$) apart and typically three to five scales are used. In all cases, only the first two orders are used and higher orders are ignored.

Behavior Under Coordinate Deformations

There are several additional properties that make the Gaussian a suitable operator for analysis of images. In this section we examine the behavior of the Gaussian and its derivatives with respect to coordinate deformations of the image. In particular, behavior with respect to size changes and 2-D rotations of the coordinate frame are considered.

Scaling Theorems. Gaussian derivatives may be used to compare image patches that are scaled versions of each other in a straightforward manner. Consider two images I_0 and I_1 that are scaled versions of each other (but otherwise identical). Assume that the scaling is centered at the origin. That is, $I_0(\boldsymbol{p}) = I_1(s\boldsymbol{p})$ Then the following relations hold (12,13):

$$\begin{aligned} I_0(\boldsymbol{p}) \star G(\cdot, \sigma) &= I_1(s\boldsymbol{p}) \star G(\cdot, s\sigma) \\ I_0(\boldsymbol{p}) \star G^{(k)}(\cdot, \sigma) &= I_1(s\boldsymbol{p}) \star G^{(k)}(\cdot, s\sigma) \end{aligned} \tag{16}$$

where

$$G^{(k)}(\cdot, t) = t^k G_{i_1 \ldots i_k}(\cdot, t)$$

We call these the *scale-shifting theorems* or simply *scaling theorems.* These equations state that if the image I_s is a scaled version of I_0 by a factor s, then in order to compare any two corresponding points in these images the filters must also be stretched (i.e. scaled) by the same factor. For example, if a point p_0 is being compared with a point p_1 in images I_0 and I_1, where I_1 is twice the size of I_0, then for the responses to be equal, the filter used to compute the response at p_1 must be at twice the scale of that applied at p_0. This property has been exploited for matching affine deformed images (see the next section), object recognition, (14) and image retrieval (15).

Steerability. The Gaussian derivatives may be combined under rotations to synthesize filters in an arbitrary orientation. This has been called the steering property (16). This property is interesting for two reasons. First, images may be filtered using Gaussian derivatives tuned to any arbitrary orientation without actually rotating the filters. The tuned filters may be expressed as a combination of filters in a normal coordinate frame. Therefore, responses to any steered direction may be computed as a simple rotation of the responses. Thus, separable implementations are feasible even for rotated filters. Second, it may be used as a basis for generating feature vectors that are invariant to 2-D rotations as discussed in the next section. The results for the first two orders are now derived.

Consider a 2-D rotated version of the Cartesian coordinate frame $\boldsymbol{p} = \langle x, y \rangle$ written as $\boldsymbol{q} = \langle x_2, y_2 \rangle$ such that $\boldsymbol{q} = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{p}$, where $\boldsymbol{p}$ and $\boldsymbol{q}$ are the respective coordinates and $\boldsymbol{R}$ is the rotation matrix. Assume for simplicity that all coordinates are right handed.

Isotropy. It is straightforward to show that $G(\boldsymbol{q}, \sigma) = G(\boldsymbol{R}^{\mathrm{T}}\boldsymbol{p}, \sigma) = G(\boldsymbol{p}, \sigma)$. That is, the Gaussian is isotropic.

First Derivatives. Consider the first derivatives of the 2-D Gaussian. The following relationship holds from the circular symmetry of the Gaussian.

$$\begin{aligned} \begin{bmatrix} G_{x_2}(\boldsymbol{q}, \sigma) \\ G_{y_2}(\boldsymbol{q}, \sigma) \end{bmatrix} &= -\frac{1}{\sigma^2}\boldsymbol{q}G(\boldsymbol{q}, \sigma) \\ &= -\frac{1}{\sigma^2}(\boldsymbol{R}^{\mathrm{T}}\boldsymbol{p})G(\boldsymbol{R}^{\mathrm{T}}\boldsymbol{p}, \sigma) \\ &= \boldsymbol{R}^{\mathrm{T}}\left(-\frac{1}{\sigma^2}(\boldsymbol{p})G(\boldsymbol{p}, \sigma)\right) \\ &= \boldsymbol{R}^{\mathrm{T}}\begin{bmatrix} G_x(\boldsymbol{p}, \sigma) \\ G_y(\boldsymbol{p}, \sigma) \end{bmatrix} \end{aligned} \tag{17}$$

Second Derivatives. Similarly, the second derivative may also be steered:

$$\begin{aligned} &\begin{bmatrix} G_{x_2y_2}(\boldsymbol{q}, \sigma) & G_{x_2y_2}(\boldsymbol{q}, \sigma) \\ G_{y_2x_2}(\boldsymbol{q}, \sigma) & G_{y_2y_2}(\boldsymbol{q}, \sigma) \end{bmatrix} \\ &\quad = \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}\begin{bmatrix} x_2^2 & x_2y_2 \\ x_2y_2 & y_2^2 \end{bmatrix} - \boldsymbol{I}_2\right) G(\boldsymbol{q}, \sigma) \\ &\quad = \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}\boldsymbol{q}\boldsymbol{q}^{\mathrm{T}} - \boldsymbol{I}_2\right) G(\boldsymbol{q}, \sigma) \\ &\quad = \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}\boldsymbol{R}^{\mathrm{T}}\boldsymbol{p}\boldsymbol{p}^{\mathrm{T}}\boldsymbol{R} - \boldsymbol{I}_2\right) G(\boldsymbol{R}\boldsymbol{p}, \sigma) \\ &\quad = \boldsymbol{R}^{\mathrm{T}}\frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}\boldsymbol{p}\boldsymbol{p}^{\mathrm{T}} - \boldsymbol{I}_2\right) G(\boldsymbol{p}, \sigma)\boldsymbol{R} \\ &\quad = \boldsymbol{R}^{\mathrm{T}}\begin{bmatrix} G_{xx}(\boldsymbol{p}, \sigma) & G_{xy}(\boldsymbol{p}, \sigma) \\ G_{yz}(\boldsymbol{p}, \sigma) & G_{yy}(\boldsymbol{p}, \sigma) \end{bmatrix}\boldsymbol{R} \end{aligned} \tag{18}$$

where

$$\boldsymbol{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Several authors have exploited the steerability property of Gaussian derivatives.

Rao and Ballard (14) steered multiscale derivative vectors so as to represent the orientation with the best local responses, and Ravela et al. (17) exploited steerability to track image patches. In the next section this property is used in conjunction with differential invariants to construct multiscale invariant vectors that are derivatives expressed in the local coordinate frame and are invariant to 2-D rotations.

Rotational Invariants. Gaussian derivatives may be steered to any given orientation. Therefore, the image derivatives can be stably computed along any orientation using Gaussian derivatives tuned to that orientation. This property allows the creation of *features* that are invariant to 2-D rotations of the image plane.

It is well known (6) that if some property of the local intensity surface is used to define a local coordinate frame that is a rotation of the image coordinate frame, then derivatives computed in the local frame will be invariant to rotations of the intensity surface. For example, assume that a new coordinate frame is defined by the local gradient direction in the image I. In the above-mentioned framework let the axis y_2 represent a direction parallel to the local gradient, and let x_2 be orthogonal to it in a right-hand coordinate sense. Then one may define an orthonormal matrix $\boldsymbol{R}$ such that

$$\boldsymbol{R} = \frac{1}{\sqrt{I_x^2 + I_y^2}} \begin{bmatrix} I_y & I_x \\ -I_x & I_y \end{bmatrix} \tag{19}$$

Note that the matrix $\boldsymbol{R}$ is defined locally at every point. The new coordinate frame $\langle x_2, y_2 \rangle$ will likely change from pixel to pixel, but is automatically determined. Thus, an image filtered with the first Gaussian derivative steered to the $\langle x_2, y_2 \rangle$ coordinate frame can be equivalently expressed in the image coordinate frame $\langle x, y \rangle$ in the following manner:

$$\begin{aligned} \begin{bmatrix} I_{x_2} \\ I_{y_2} \end{bmatrix} &= Z \star \begin{bmatrix} G_{x_2} \\ G_{y_2} \end{bmatrix} = Z \star \boldsymbol{R}^{\mathrm{T}} \begin{bmatrix} G_x \\ G_y \end{bmatrix} \\ &= \boldsymbol{R}^{\mathrm{T}} \begin{bmatrix} Z \star G_x \\ Z \star G_y \end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{I_x^2 + I_y^2} \end{bmatrix} \end{aligned} \tag{20}$$

The interpretation of this result is rather simple. The gradient magnitude is the directional derivative parallel to the direction of the gradient. It is also invariant to rotations since the gradient response at any pixel transforms to exactly the above defined vector (20). The second derivative may similarly be expressed. There are several ways in which one may construct the rotation matrix $\boldsymbol{R}$. Further, given a multiscale derivative feature vector to any order, an infinite number of rotational invariants may be constructed.

However, Florack (8) has shown that given the derivatives of an image I up to a certain order, only a finite number of *irreducible differential invariants* exist, and they may be computed in a systematic manner. The irreducible set of invariants up to order two of an image I are

$$\begin{aligned} d_0 &= I && \text{(intensity)} \\ d_1 &= I_x^2 + I_y^2 && \text{(magnitude)} \\ d_2 &= I_{zz} + I_{yy} && \text{(Laplacian)} \\ d_3 &= I_{xx} I_x I_x + 2 I_{xy} I_x I_y + I_{yy} I_y I_y \\ d_4 &= I_{xx}^2 + 2 I_{xy}^2 + I_{yy}^2 \end{aligned}$$

The reason these are termed irreducible is that other invariants (up to that order) may be expressed as combinations of them. Thus, the multiscale derivative vectors may be transformed so that they are invariant to 2-D rotations in the image plane. In the retrieval by appearance application we use the vector (minus the intensity term) $\boldsymbol{\Delta}_\sigma = \langle d_1, \ldots d_4 \rangle_\sigma$, computed at three different scales. This representation to a higher order has also been used by Schmid and Mohr (18) for object recognition.

Implementation

Filtering may be carried out either in the spatial domain using convolution or in the frequency domain using composition. In the latter case the Fourier transform and the inverse Fourier transform will need to be computed before and after the composition operation.

The choice of the domain for filtering is dependent on the size of the kernel. For an image of size N and filter of size w, the complexity of spatial domain filtering using separable filters is $O(wN)$, while that using frequency domain filtering is $O(N \log N + N^2/\omega^2)$. Thus, when the image and kernel sizes are small, spatial-domain filtering may be preferred, while for large images frequency-domain filtering may be advantageous. In addition, if several operations need to be performed, such as filtering with multiple derivatives, then frequency-domain filtering may be preferred.

In either domain, the issue of *discretization* has to be faced. The derivation in the previous section relied on continuous functions, whereas practical implementations require discrete versions of these filters. In addition, in the spatial domain, *truncation* effects need to be considered as well—that is, the effects of truncating the Gaussian to a finite extent. In this subsection the effects of discretizing and truncating the filter in the spatial domain will be discussed.

Discretizing Gaussians and Gaussian Derivatives. The derivations of the algorithms in the previous sections have assumed that the Gaussians and Gaussian derivatives were continuous functions. To apply them, they first need to be discretized. The discretization needs to be performed carefully, since errors can arise from it (6,19). A number of different procedures have been suggested in the literature. These include:

1. *Block Averaging.* The continuous kernel is averaged over each pixel, that is, the filter value is integrated over each pixel and then sampled (19). Let the discrete filter be defined over the values $-w$ to w (that is, its width is $2w + 1$). Then the value of the discrete Gaussian kernel at a point i is given by

$$g[i] = \int_{i-1/2}^{i+1/2} G(x, \sigma)\, dx = \operatorname{erf}\left(\frac{i + 1/2}{\sigma}\right) - \operatorname{erf}\left(\frac{i - 1/2}{\sigma}\right) \tag{21}$$

where er f$(x) = \int_0^x \exp(-t^2/2)\, dt$ is the error function. The first derivatives may be computed similarly. For example, the first derivative of the Gaussian in the x direction is given by

$$g_x[i] = \int_{i-1/2}^{i+1/2} G_x(x, \sigma)\, dx = G(i + 1/2, \sigma) - G(1 - 1/2, \sigma) \tag{22}$$

2. *Discrete Derivative.* The Gaussian is first sampled. The image is then convolved with the sampled Gaussian, and the output convolved with a discrete version of the derivative. This approach is widely used. The discrete Gaussian kernel at a point i is given by

$$g[i] = G(i, \sigma) \tag{23}$$

A discrete version of the first derivative is given by the kernel $D = [-1, 0, 1]$. Then, a discrete version of the first Gaussian derivative in the x direction is given by

$$g_x = D * g[i] \tag{24}$$

Since convolution is associative, the image may first be filtered with the discrete Gaussian and the result may then be filtered with the first derivative kernel D. Second derivatives may be computed using the second-derivative kernel $D_2 = [-1, 2, -1]$.

3. *Sampled Gaussian Derivatives.* The continuous version of the Gaussian or Gaussian derivative is sampled directly at each pixel, and the sample values used for the discrete version of the filter. The sampled Gaussian derivative is given by

$$g_{x^n}[i] = G_{x^n}(i, \sigma) \tag{25}$$

4. *Discrete Scale Space.* The values of the discrete Gaussian are computed using a discrete scale space. The image is filtered with the discrete Gaussian and then filtered with a discrete version of the derivative. See Ref. 6 for how to compute the discrete Gaussian.

The question arises as to which technique is appropriate. It may be argued that block averaging takes account of the imaging process and may therefore be assumed to be the best discretization (19,6). For example, when a scene is imaged by a charge coupled device (CCD) camera, the output of each pixel is proportional to the total light falling over the entire area of each pixel (that is, the integral of the brightness over that pixel). The area is actually better approximated as a weighted integral—the weight being a Gaussian (6).

The results obtained using sampled Gaussian derivatives approximate those due to block averaging provided the scales (σ) used are large. As the scale is reduced, the errors due to using a sampled Gaussian derivative increase. Typically, below $\sigma = 0.5$ sampled Gaussian derivatives should not be used. In practice, most scales used are larger and hence sampling Gaussian derivatives are usually a good method of computing discrete Gaussian derivatives.

Assume that a large number of Gaussian derivatives need to be computed. Then for each order of a Gaussian derivative, the Gaussian needs to be sampled and the image filtered with the appropriate discrete kernel. This can be expensive. Consider, for example, the 1-D case, and assume that the image needs to be filtered with derivatives up to order 2. Let the kernel width for the discrete versions of the Gaussian, first derivative, and second derivative be $2w + 1$. Then to filter the image with Gaussian derivatives up to order 2 requires time proportional to $3(2w + 1) = 6w + 3$. This time may be reduced by using discrete derivatives to compute Gaussian derivatives. First, the image is filtered with a sampled Gaussian. The output of this image is filtered with the kernels D and $D2$, and this is equivalent to filtering the image with the first and second derivatives of the Gaussian. The time taken to filter, however, is only $2w + 7$. Since w may often be large, Gaussian derivative filtering accomplished using discrete derivatives is cheaper to compute than using sampled Gaussian derivatives. The tradeoff is that the results obtained using discrete derivatives are not as accurate. That is, sampled Gaussian derivatives are a better approximation to block averages than discrete Gaussian derivatives (see Ref. 20).

Truncation of Gaussian Derivatives. Gaussians and Gaussian derivatives are infinite in extent. However, most of their energies reside in a small region around the origin. Thus, for all practical purposes they may be truncated. Truncation also reduces the time taken to filter the images, since the resulting kernel sizes are smaller. There has been some discussion about where Gaussians and Laplacians of Gaussians should be truncated (21,19), but a general discussion of how Gaussian derivatives should be truncated seems to be absent in the literature (but see Ref. 20).

Figure 1 shows the truncation errors for different values of the truncation radius. The truncation error is computed as follows:

$$\text{truncation error} = \frac{\int_{-\infty}^{\infty} |G_{x^i}(x, \sigma)|\, dx - \int_{-k\sigma}^{k\sigma} |G_{x^i}(x, \sigma)|\, dx}{\int_{-\infty}^{\infty} |G_{x^i}(x, \sigma)|\, dx} \tag{26}$$

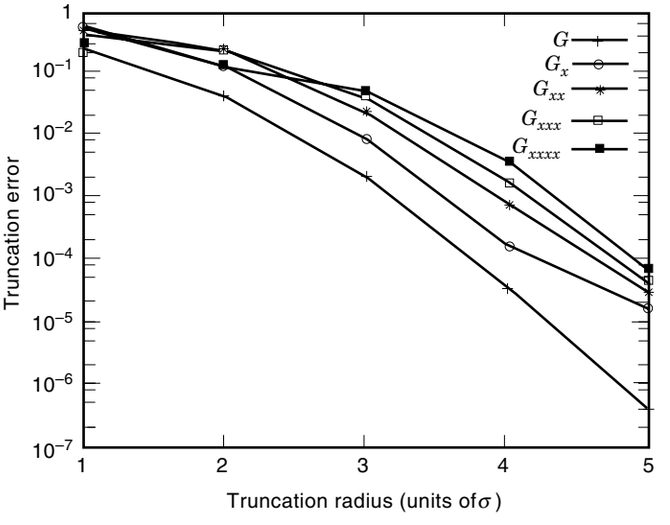


Figure 1. Truncation errors as a function of the truncation radius for Gaussians and Gaussian derivative filters. G denotes the Gaussian, and G_x, G_{xx}, G_{xxx}, and G_{xxxx} denote the first, second, third, and fourth Gaussian derivatives respectively. The truncation errors are taken as a proportion of the total integral of the absolute value of filter.

where $G_{x^i}(x, \sigma)$ is the ith-order Gaussian derivative (with G denoting the Gaussian) and k is the truncation radius. In the above equation, the difference between the integrals of the absolute values of the truncated function and the untruncated function is first computed. Then this difference is divided by the integral of the absolute values of the untruncated filter to give the truncation error as a proportion of the integral of the untruncated filter. Note that the truncation error is independent of σ. The truncation errors in Figure 1 were computed by summing over discrete versions of the filter using large σ's instead of computing the integrals analytically (the difference should be insignificant).

As Figure 1 shows, for a given truncation radius, the error increases with the order of the derivative. Many researchers assume that it suffices to truncate Gaussians so that the truncation radius is $\pm 2\sigma$, that is, the filter width is 4σ. For a truncation radius of 2σ, 96% of the energy of the Gaussian is contained within the filter width (i.e., the truncation error is 0.04). But for the same truncation radius, Gaussian derivatives have a much larger truncation error. For example, the first derivative of the Gaussian has an error of about 12% if it is truncated to within $\pm 2\sigma$, while higher derivatives have a much larger error. Figure 1 shows that the truncation error is less than 0.01 for the first four Gaussian derivatives if the truncation radius is greater than or equal to $\pm 4\sigma$. It may also be shown that the qualitative errors produced are large if the truncation radius is less than $\pm 4\sigma$ for derivatives up to order 4 (see Ref. (20).

Suggested Reading

Multiresolution representations are related to multiscale representations. A classical multiresolution representation, the Laplacian pyramid (2) may be generated as follows:

$$I^{(n)} = F[I^{(n-1)}]$$
$$I^{(0)} = Z_{xx} + Z_{yy}$$

where Z is the original image and $I^{(n)}$ is a representation at a coarser resolution. The operator F consists of two operations: the first one is a smoothing step, and the second is a subsampling step. The smoothing step is required to reduce aliasing effects due to subsampling and may be implemented using a Gaussian. A subsampling factor of 2 is typically used, so that a coarser image is a quarter the size of its immediate predecessor. Multiresolution representations may be used to compress images, detect features in a coarse-to-fine manner, and match features between images efficiently.

Multiresolution representations are related to, but somewhat different from, multiscale representations. Multiscale representations do not change the resolution of the original image, but rather vary the size of the operator. It is trivial to build a multiresolution representation from a multiscale representation, but the reverse is not true.

Although this section presents enough detail to motivate the use of Gaussian filtered representations, there are several aspects that are not covered. In particular, the study of scale space is abbreviated, and the user is referred to Refs. 11, 4, 6 for further review. Similarly, an in-depth study of rotational invariants is available in Ref. 8. For some additional properties of the Gaussian filter, such as optimality with respect to the uncertainty principle, the reader is referred to Ref. 22. In addition there is a physiological motivation for using Gaussian filtered representations. For example, Young (23) shows that visual receptive fields in the primate eye are better modeled by Gaussian derivatives.

Several researchers have used multiscale derivatives as a representation. In particular Ref. 14 uses multiscale vectors and the steerability property to recognize objects. In earlier work (15) derivatives were combined with the scaling theorems to retrieve visually similar objects at different sizes. The next section is also a good example of using multiscale derivatives. There they are used to recover the affine transform between deformed images.

MATCHING AFFINE DEFORMED IMAGES

In this section we will discuss how Gaussian and Gaussian-derivative filters may be used to match images under affine transforms. The ability to match two images or parts of images is required for many visual tasks. For example, recovering the structure of a scene requires matching two or more image patches arising from a scene viewed from different viewpoints. Other applications that require matching image patches include the registration of video sequences (24) and image mosaicking (25). Successive images of a scene when taken from different viewpoints are deformed with respect to each other. To first order, the transformation between images caused by viewpoint change may be modeled using an affine transform. The affine transform interprets the image motion in terms of an image translation and a deformation. In 2-D, the affine transformation may be described by the six parameters ($\boldsymbol{t}$, $\boldsymbol{A}$) where

$$r' = t + \boldsymbol{A}r \tag{27}$$

$\boldsymbol{r}'$ and $\boldsymbol{r}$ are the image coordinates related by an affine transform, $\boldsymbol{t}$ is a 2-by-1 vector representing the translation, and $\boldsymbol{A}$ the 2-by-2 affine deformation matrix. The affine transform is useful because the image projections of a small planar patch from different viewpoints are well approximated by it.

In general, affine transforms between image patches have been recovered in a number of different ways (for more details see Ref. 13):

1. Matching image intensities by searching through the space of all affine parameters. This approach adopts a brute force search strategy which is slow (26).
2. Linearizing the image intensities with respect to the affine parameters. This may be done at each pixel to give one equation per pixel. By assuming that the same affine transformation is valid over some region, an overconstrained system of equations is obtained. Linearization limits these algorithms to cases when the affine transforms are small (27–29).
3. Filtering the image with Gaussians and linearizing the filter outputs (30). The results are poor for general affine transforms because only the filter outputs from a single pixel are used.
4. Line-based methods that match the closed boundaries of corresponding regions (31,32). However, they are limited to homogeneous regions with closed boundaries.

5. Matching patches deformed under similarity transforms using the Mellin–Fourier transform (33). Although possible, recovery of the affine transform has not been demonstrated. The main drawback to these techniques is that they are inherently global and they are not applicable to general affine transforms.

The difficulty with measuring affine transforms is indicated in Fig. 2, where the image on the right is scaled to 1.4 times the image on the left. Even if the centroids of the two images are matched accurately, measuring the affine transform is difficult, since the sizes of every portion of the two images differ. This problem arises because traditional matching uses fixed correlation windows or filters. The correct way to approach this problem is to deform the correlation window or filter according to the image deformation.

Here, we present a computational scheme where Gaussian and derivative-of-Gaussian filters are used and the filters deformed according to the affine transformation. First, it is shown that if an image is filtered with a Gaussian (or Gaussian derivative), then the affine transformed version of the image needs to be filtered with a deformed Gaussian (or Gaussian derivative) if the two filter outputs are to be equal; the deformation is equal to the affine transform. Thus, the problem of recovering the affine transform may be recast into the problem of finding the deformation parameters of the Gaussian (or Gaussian derivative). For example let Z_1 and Z_2 be two images that differ by a scale change s. Then the output of Z_1 filtered with a Gaussian of σ will be equal to the output of Z_2 filtered with a Gaussian of $s\sigma$.

The resulting equations are solved by linearizing with respect to the affine parameters. Unlike the technique used in Ref. 30, the filter outputs from a number of points in a region are pooled together. This substantially improves the accuracy of the technique. For example, using Werkhoven and Koenderink's algorithm (30) on the images in Fig. 2 returns a scale factor of 1.16, while the algorithm here matches correctly and therefore returns a scale factor of 1.41.

Deformation of Filters

The initial discussion will assume zero image translation; translation can be recovered as suggested in the subsection "Finding the Image Translation" below. It is also assumed that shading and illumination effects may be ignored.

Consider two Riemann-integrable functions Z_1 and Z_2 related by an affine transform:

$$Z_2(\boldsymbol{A}\boldsymbol{r}) = Z_1(\boldsymbol{r}) \tag{28}$$



Figure 2. Dollar bill scaled 1.4 times.

Consider first the case where Z_1 is a scaled version of Z_2, that is,

$$Z_2(s\boldsymbol{r}) = Z_1(\boldsymbol{r}) \tag{29}$$

Then

$$\int Z_1(\boldsymbol{r})G(\boldsymbol{r},\sigma)\,d\boldsymbol{r} = \int Z_2(s\boldsymbol{r})G(\boldsymbol{r},\sigma)\,d\boldsymbol{r} \tag{30}$$

$$= \int Z_2(s\boldsymbol{r})G(s\boldsymbol{r},s\sigma)\,d(s\boldsymbol{r}) \tag{31}$$

That is, the output of Z_1 filtered with a Gaussian is equal to the output of Z_2 filtered with a scaled Gaussian. Note that this equation is also true for similarity transforms, that is, $\boldsymbol{A} = s\boldsymbol{R}$.

Define a generalized Gaussian by

$$G(\boldsymbol{r},\boldsymbol{M}) = \frac{1}{(2\pi)^{n/2}\det(\boldsymbol{M})^{1/2}}\exp\left(-\frac{\boldsymbol{r}^{\mathrm{T}}\boldsymbol{M}^{-1}\boldsymbol{r}}{2}\right) \tag{32}$$

where $\boldsymbol{M}$ is a symmetric positive semidefinite matrix. Then, if Z_1 and Z_2 are related by an affine deformation, the output of Z_1 filtered with a Gaussian is equal to the output of Z_2 filtered with a Gaussian deformed by the affine transform (see Refs. 20 and 34 for a derivation).

$$\int Z_1(\boldsymbol{r})G(\boldsymbol{r},\sigma^2\boldsymbol{I})\,d\boldsymbol{r} = \int Z_2(\boldsymbol{A}\boldsymbol{r})G(\boldsymbol{A}\boldsymbol{r},\boldsymbol{R}\boldsymbol{\Sigma}\boldsymbol{R}^{\mathrm{T}})\,d(\boldsymbol{A}\boldsymbol{r}) \tag{33}$$

where the integrals are taken from $-\infty$ to ∞. $\boldsymbol{R}$ is a rotation matrix and Σ a diagonal matrix with entries $(s_1\sigma)^2$, $(s_2\sigma)^2$, . . ., $(s_n\sigma)^2$, $(s_i \geq 0)$, and $\boldsymbol{R}\Sigma\boldsymbol{R}^{\mathrm{T}} = \sigma^2\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ (this follows from the fact that $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ is a symmetric, positive semidefinite matrix).

Intuitively, Eq. (33) expresses the notion that the Gaussian-weighted-average brightnesses must be equal, provided the Gaussian is affine transformed in the same manner as the function. The problem of recovering the affine parameters has been reduced to finding the deformation of a known function, the Gaussian, rather than the unknown brightness functions.

The level contours of the generalized Gaussian are ellipses rather than circles. The tilt of the ellipse is given by the rotation matrix, while its eccentricity is given by the matrix Σ, which is actually a function of the scales along each dimension. The equation clearly shows that to recover affine transforms by filtering, one must deform the filter appropriately—a point ignored in previous work (26–28). The equation is local because the Gaussians rapidly decay.

The integral may be interpreted as the result of convolving the function with a Gaussian at the origin. It may also be interpreted as the result of a filtering operation with a Gaussian. To emphasize these similarities, it may be written as

$$Z_1 * G(\boldsymbol{r},\sigma^2\boldsymbol{I}) = Z_2 * G(\boldsymbol{r}_1,\boldsymbol{R}\boldsymbol{\Sigma}\boldsymbol{R}^{\mathrm{T}}) \tag{34}$$

where $\boldsymbol{r}_1 = \boldsymbol{A}\boldsymbol{r}$.

Similar equations may be written using derivative-of-Gaussian filters (for details see Refs. 20, 34).

Solution for the Case of Similarity Transforms

To solve Eq. (33) requires finding a Gaussian of the appropriate scale $s\sigma$ given σ. A brute force search through the space

of scale changes is not desirable. Instead a more elegant solution is to linearize the Gaussians with respect to σ. This gives an equation linear in the unknown α:

$$\begin{aligned} Z_1 * G(\cdot, (\sigma)^2) &= Z_2 * G(\cdot, (s\sigma)^2) \\ &\approx Z_2 * G(\cdot, \sigma^2) + \alpha\sigma Z_2 * \frac{\partial G(\cdot, \sigma^2)}{\partial \sigma} \\ &= Z_2 * G(\cdot, \sigma^2) + \alpha\sigma^2 \nabla^2 Z_2 * G(\cdot, \sigma^2) \end{aligned} \tag{35}$$

where $s = 1 + \alpha$. The last equality follows from the diffusion equation $\partial G/\partial\sigma = \sigma\nabla^2 G$.

Equation (35) is not very stable if solved at a single scale. By using Gaussians of several different scales σ_i the following linear least-squares problem is obtained:

$$\sum_i \| Z_1 * G(\cdot, \sigma_i^2) - Z_2 * G(\cdot, \sigma_i^2) + \alpha\sigma_i^2 Z_2 * \nabla^2 G(\cdot, \sigma_i^2) \|^2 \tag{36}$$

It is solved using singular value decomposition (SVD).

It is not necessary to use every possible scale for σ_i. It turns out that the information at closely spaced scales is highly correlated and it usually suffices to use σ_i spaced apart by half an octave (a factor of about 1.4). For example, a possible set of scales would be (1.25, 1.7677, 2.5, 3.5355, 5.0).

Choosing a Different Operating Point

For large scale changes (say $\geq$1.2) the recovered scale tends to be poor. This is because the Taylor series approximation is good only for small values of α. The advantage of linearizing the Gaussian equations with respect to σ is that the linearization point can be shifted, that is, the right-hand side of Eq. (33) may be linearized with respect to a σ different from the one on the left-hand side to give the following equation:

$$Z_1 * G(\cdot, \sigma_i^2) \approx Z_2 * G(\cdot, \sigma_j^2) + \alpha' \sigma_j^2 Z_2 * \nabla^2 G(\cdot, \sigma_j^2) \tag{37}$$

where $s = \sigma_j/\sigma_i(1 + \alpha')$. The strategy therefore is to pick different values of σ_j and solve Eq. (37) (or actually an overconstrained version of it). Each of these σ_j will result in a value of α'. The correct value of α' is that which is most consistent with the equations. By choosing the σ_j appropriately, it can be ensured that no new convolutions are required.

In principle, arbitrary scale changes can be recovered using this technique. In practice, only a range of scales need to be recovered, and therefore a small set of operating points will suffice.

Finding the Image Translation

Image translation (optic flow) can be recovered in the following manner. Let Z_1 and Z_2 be similarity-transformed versions of each other (i.e., they differ by a scale change, a rotation, and a translation). Assume that an estimate of the translation $\boldsymbol{t}_0$ is available. Linearizing with respect to $\boldsymbol{r}$ and σ gives

$$\begin{aligned} Z_1(\boldsymbol{r} + t_0) * G(\boldsymbol{r}, \sigma^2) - \delta\boldsymbol{t}^{\mathrm{T}} Z_1(\boldsymbol{r} + \boldsymbol{t}_0) * G(\boldsymbol{r}, \sigma^2) \\ \approx Z_2 * G(\cdot, \sigma^2) + \alpha\sigma^2 Z_2 * \nabla^2 G(\cdot, \sigma^2) \end{aligned} \tag{38}$$

which is again linear in both the scale and the residual translation $\delta\boldsymbol{t}$. As before, an overconstrained version of this equation using multiple scales is obtained and solved for the unknown parameters. Large scales are handled as before.

$\boldsymbol{t}_0$ is obtained either by a local search or from a coarser level in a pyramid scheme, while $\delta\boldsymbol{t}$ is estimated from the equation.

Solving for the General Affine Transformation

There are two factors that need to be taken into account in the general case. First note that in the similarity case all the filtering was done at one point (the origin). The results may be further improved by filtering at many points rather than just one point. However, the rotation invariance will then be lost. In the general affine case, because of the larger number of parameters that have to be recovered, the filtering must be done at many points. The deformation must also be allowed for, and this can be done by linearizing the generalized Gaussian with respect to the affine parameters.

Filtering at a point $\boldsymbol{l}_i$ modifies the generalized Gaussian Eq. (33) as follows: Given a point with coordinates $\boldsymbol{l}_i$,

$$\int Z_1(\boldsymbol{r}) G(\boldsymbol{r} - \boldsymbol{l}_i, \sigma^2 \boldsymbol{I})\, d\boldsymbol{r} = \int Z_2(\boldsymbol{A}\boldsymbol{r}) G(\boldsymbol{A}(\boldsymbol{r} - \boldsymbol{l}_i)), \boldsymbol{R}\boldsymbol{\Sigma}\boldsymbol{R}^{\mathrm{T}})\, d(\boldsymbol{A}\boldsymbol{r}) \tag{39}$$

Thus if the image is filtered at point $\boldsymbol{l}_i$ in the first image patch, it must be filtered at point $\boldsymbol{A}\boldsymbol{l}_i$ in the second image patch.

$$\begin{aligned} Z_1 * G(\boldsymbol{r} - \boldsymbol{l}_i, \sigma) \approx Z_2 * G(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma) - \boldsymbol{B}\boldsymbol{l}_i)^{\mathrm{T}} Z_2 * G'(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma) \\ + \sigma^2 [b_{11} Z_2 * G_{xx}(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma) \\ + b_{22} Z_2 * G_{yy}(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma) \\ + (b_{12} + b_{21}) Z_2 * G_{xy}(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma)] \end{aligned} \tag{40}$$

where the b_{ij} are elements of $\boldsymbol{B} = \boldsymbol{A} - \boldsymbol{I}$ and $\boldsymbol{I}$ is the identity matrix. Note that this is linear in the affine parameters b_{ij}. A number of methods incorporate the idea of filtering at many points (27–29). However, none of these compensate for the deformation terms (in essence, the difference between the traditional linearization methods and the technique presented here is the additional second-derivative terms).

Translation may be incorporated by noticing that the effect of translation is similar to that of $\boldsymbol{I}_i$. Thus, with translation included, the above equation may be rewritten as

$$\begin{aligned} Z_1 * G(\boldsymbol{r} - \boldsymbol{l}_i, \sigma) \approx Z_2 * G(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma) - (\boldsymbol{B}\boldsymbol{l}_i)^{\mathrm{T}} Z_2 * G'(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma) \\ + \sigma^2 [b_{11} Z_2 * G_{xx}(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma) \\ + b_{22} Z_2 * G_{yy}(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma) \\ + (b_{12} + b_{21}) Z_2 * G_{xy}(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma)] \\ - \boldsymbol{t}^{\mathrm{T}} Z_2 * G'(\boldsymbol{r}_1 - \boldsymbol{l}_i, \sigma) \end{aligned} \tag{41}$$

The equation may be turned into an overconstrained linear system by choosing a number of scales σ_i and a number of points $\boldsymbol{I}_i$. Two or three scales are chosen as before. The points $\boldsymbol{I}_i$ are picked as follows: either every point in the region is used, or a regularly spaced subset of the points are used. The resulting overconstrained system may be solved using least mean squares and minimizing with respect to the affine parameters. One way of writing the solution to this least-mean-squares system is

$$\boldsymbol{b} = \boldsymbol{M}^{-1}\boldsymbol{z} \tag{42}$$

where $\boldsymbol{b} = [a_{11} - 1, a_{12}, a_{21}, a_{22} - 1, t_x, t_y]$ are the required affine parameters, and the ith row of the n-by-6 matrix $\boldsymbol{M}$ is given by

$$[\sigma^2 Z_2 * G_{xx} - x_i Z_2 * G_x, \sigma^2 Z_2 * G_{xy} - y_i Z_2 * G_x \\ \sigma^2 Z_2 * G_{xy} - x_i Z_2 * G_y, \sigma^2 Z_2 * G_{yy} - y_i Z_2 * G_y, -Z_2 * G_x, -Z_2 * G_y] \quad (43)$$

where the Gaussian derivatives are taken at point $\boldsymbol{r}_1 - \boldsymbol{l}_i$. The ith element of the vector $\boldsymbol{z}$ is equal to $Z_1 * G(\boldsymbol{r} - \boldsymbol{l}_1, \sigma) - Z_2 * G(\boldsymbol{r}_1 - \boldsymbol{l}_1, \sigma)$.

The solution is done iteratively. At each step, the affine transformation is solved for. The image is then warped according to the transformation and the residual affine transformation solved for. The convergence is very rapid. A good solution is obtained using two scales 1.25, 1.77 spaced half an octave apart and with a window of size 13 by 13 (that is, points from a region of size 13 by 13 are selected) (20). The technique allows fairly large affine transforms to be recovered (scaling of as much as 40%).

The technique has some limitations. For large translations, a good initial estimate of the translation is required. This may be obtained in a number of ways. A coarse-to-fine technique may be used to estimate the translation. Alternatively, the method used to find similar points in the next section may be used to provide an estimate of the translation.

IMAGE RETRIEVAL

In this section Gaussian filtered representations of images are applied to the task of retrieval by image appearance. The paradigm used for retrieval is that a collection of images are represented using feature vectors constructed from Gaussian derivatives. During run time, a user presents an example image or parts thereof as a query to the system. The query's feature vectors are compared with those in the database, and the images in the database are ranked and displayed to the user.

There are several objectives that govern the design of a retrieval system. Primary among those are speed and the ability to find visually similar objects within a reasonable space of deformations. There is a third objective that can provide considerable flexibility to a user: the ability of a system to query parts of images if required. This is because a user may be interested in a part of an image such as a face in a crowd rather than a whole image such as a trademark. The interesting aspect of the algorithms presented here is that both these types of retrieval can be achieved without a system automatically trying to compute salient features or regions, which can be extremely challenging. All the system does in either case is compare signals (feature vectors). When the user has a notion of what is important in an image, it is exploited to find other vectors similar to it.

However, the desired flexibility imposes different constraints on the retrieval algorithms. Finding parts of images requires measurement of local similarity, and the representation must be local. Therefore, individual feature vectors might be used. On the other hand, finding whole images implies global similarity, and distributions of features can be used.

In previous work (15) derivative feature vectors in conjunction with the scaling theorems were used for retrieval by appearance. Database images were filtered with the Gaussian derivatives up to the second order, at several scales. A query image (or parts of it) was also filtered, but at a single scale. Then, using the scaling theorems given earlier in this article, the query feature vectors were correlated across scales with each database image's feature vectors. The results indicated that visually similar objects can be retrieved within about 25° of rotation. Similar results were shown by Rao and Ballard (14) in object recognition experiments.

However, correlation is slow. Further, it does not allow the feature vectors to be indexed. Thus, one cannot expect to develop a system of reasonable speed even for moderately sized databases using this method. Here, we present two progressively faster and indexable methods to retrieve images. The first method may be used to find parts of images, and the second for finding whole images. Finding parts of images requires similarities of local image features to be computed. This implies an explicit representation of local features. On the other hand, whole-image matching. In the next two sections the local and global similarity retrieval algorithms are elaborated.

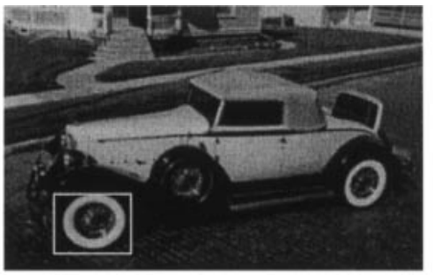
Local-Similarity Retrieval

Local-similarity retrieval is carried out as follows. Database images are uniformly sampled. At each sampled location, the multiscale invariant feature vector $\boldsymbol{\Delta}_\sigma$ defined in the subsection "Rotational Invariants" above is computed at three different scales. Then, vectors computed for all the images in the database are indexed using a binary tree structure. During run time, the user picks an example image and designs a query. Since an image is described spatially (uniform sampling), parts of images or (imaged objects) can be selected. For example, consider Fig. 3(a). Here the user wants to retrieve white-wheeled cars and therefore selects the white wheel. The feature vectors that lie within this region are submitted to the system. Database images with feature vectors that match the set of query vectors both in feature space (L_2 norm of the vector) and coordinate space (matched image locations spatially consistent with query points) are returned as retrievals.

The approach for local-similarity retrieval is divided into two parts. During the off-line phase, images are sampled and multiscale derivatives are computed. These are transformed into rotational invariants and indexed. During the on-line phase, the user designs a query by selecting regions within an image. Feature vectors within the query are matched with those in the database, both in feature space and in coordinate space. The off-line and on-line phases are discussed next.

Off-Line Operations: Invariant Vectors and Indexing

Computing Features. A multiscale invariant vector is computed at sampled locations within the image. The vector, $\boldsymbol{\Delta}_\sigma = \langle d_1, \ldots d_4 \rangle_\sigma$ (see "Rotational Invariants" above), is computed at three different scales. The element d_0 is not used, since it is sensitive to gray-level shifts. The resulting multiscale invariant vector has at most twelve elements. Computationally, each image in the database is filtered with the first five partial derivatives of the Gaussian (i.e. to order 2) at three different scales at uniformly sampled locations. Then the multiscale invariant vector $\boldsymbol{D} = (\boldsymbol{\Delta}_{\sigma 1}, \boldsymbol{\Delta}_{\sigma 2}, \boldsymbol{\Delta}_{\sigma 3})$ is computed at those locations.



(a)



(b)

Figure 3. A query and its retrieval: (a) car query; (b) ranked retrieval.

Indexing. A location across the entire database may be identified by the *generalized coordinates,* defined as, $\boldsymbol{c} = (i, x, y)$, where i is the image number and (x, y) a coordinate within this image. The computation described above generates an association between generalized coordinates and invariant vectors. This association may be viewed as a table $M:(i, x, y, D)$ with $3 + k$ columns (k is the number of fields in an invariant vector) and a number of rows, R, equal to the total number of locations (across all images) where invariant vectors are computed. To retrieve images, a *find-by-value* functionality is needed, with which a query invariant vector is found within M and the corresponding generalized coordinate is returned. Inverted files (or tables) based on each field of the invariant vector are first generated for M. To index the database by fields of the invariant vector, the table M is split into k smaller tables $M'_1, \ldots, M'_k$, one for each of the k fields of the invariant vector. Each of the smaller tables M'_p, $p = 1, \ldots, k$, contains the four columns $[D(p), i, x, y]$. At this stage any given row across all the smaller tables contains the same generalized coordinate entries as in M. Then, each M'_p is sorted and a binary tree is used to represent the sorted keys. As a result, the entire database is indexed. A given invariant

value can therefore be located in log R time (R = number of rows).

On-Line Operation. Run-time computation begins with the user selecting regions in an example image. At sampled locations within these regions, invariant vectors are computed and submitted as a query. The success of a retrieval in part depends on well-designed queries. More importantly, letting the user design queries eliminates the need for automatically detecting the salient portions of an object, and the retrieval may be customized so as to remove unwanted portions of the image. Based on the feedback provided by the results of a query, the user can quickly adapt and modify the query to improve performance.

The search for matching images is performed in two stages. In the first stage each query invariant is supplied to the find-by-value algorithm and a list of matching generalized coordinates is obtained. In the second stage a spatial check is performed on a per-image basis, so as to verify that the matched locations in an image are in spatial coherence with the corresponding query points. In this sub-subsection the find-by-value and spatial checking components are discussed.

Finding by Invariant Value. The multiscale invariant vectors at sampled locations within regions of a query image may be treated as a list. The nth element in this list contains the information $\boldsymbol{Q}_n = (\boldsymbol{D}_n, x_n, y_n)$, that is, the invariant vector and the corresponding coordinates. In order to find by invariant value, for any query entry $\boldsymbol{Q}_n$, the database must contain vectors that are within a threshold $\boldsymbol{t} = (t_1, \ldots, t_k) > 0$. The coordinates of these matching vectors are then returned. This may be represented as follows. Let $\boldsymbol{p}$ be any normalized invariant vector stored in the database. Then $\boldsymbol{p}$ matches the normalized query invariant entry $\boldsymbol{D}_n$ only if $\boldsymbol{D}_n - t < \boldsymbol{p} < \boldsymbol{D}_n + \boldsymbol{t}$. To implement the comparison operation two searches can be performed on each field. The first is a search for the lower bound, that is, the smallest entry larger than $D_n(j) - t(j)$, and the second is a search for the upper bound, that is the largest entry smaller than $D_n(j) + t(j)$. The block of entries between these two bounds are those that match the field j. In the inverted file, the generalized coordinates are stored along with the individual field values, and the block of matching generalized coordinates are copied from disk. Then an intersection of all the returned block of generalized coordinates is performed. The generalized coordinates common to all the k fields are the ones that match query entry $\boldsymbol{Q}_n$. The find-by-value routine is executed for each $\boldsymbol{Q}_n$, and as a result each query entry is associated with a list of generalized coordinates that it matches.

Spatial Fitting. The association between a query entry $\boldsymbol{Q}_n$ and the list of f generalized coordinates that match it by value may be written as

$$\begin{aligned}\boldsymbol{A}_n &= \langle x_n, y_n, c_{n1}, c_{n2}, \ldots, c_{nf}\rangle \\ &= \langle x_n, y_n, (i_{n1}, x_{n1}, y_{n2}), \ldots, (i_{nf}, x_{nf}, y_{nf})\rangle\end{aligned}$$

Here x_n, y_n are the coordinates of the query entry $\boldsymbol{Q}_n$ and c_{n1}, . . ., c_{nf} are the f matching generalized coordinates. The notation c_{nf} implies that the generalized coordinate c matches n and is the fth entry in the list. Once these associations are available, a spatial fit on a per-image basis can be performed. Any image u that contains two points (locations) that match some query entry m and n respectively are coherent with the query entries m and n only if the distance between these two points is the same as the distance between the query entries that they match. Using this as a basis, a binary fitness measure may be defined as

$$\mathscr{T}_{m,n}(u) = \begin{cases} 1 & \text{if} \quad \exists j \exists k : |\delta_{m,n} - \delta_{c_{mj},c_{nk}}| \le T \\ & \qquad\qquad i_{m_j} = i_{n_k} = u,\ m \ne n \\ 0 & \text{otherwise} \end{cases}$$

where $\delta_{m,n}$ is the Euclidean distance between the query points m and n, and $\delta_{c_{mj}c_{nk}}$ is the Euclidean distance between the generalized coordinates c_{mj} and c_{nk}. That is, if the distance between two matched points in an image is close to the distance between the query points that they are associated with, then these points are spatially coherent (with the query). Using this fitness measure, a match score for each image can be determined. This match score is simply the maximum number of points that together are spatially coherent (with the query). Define the match score by $\text{score}(u) \equiv \max_m \sum_{n=1}^{f} \mathscr{T}(u)_{m,n}$. The computation of $\text{score}(u)$ is at worst quadratic in the total number of query points. The array of scores for all images is sorted, and the images are displayed in the order of their score. T used in $\mathscr{T}$ is a threshold and is typically 25% of $\delta_{m,n}$. Note that this measure not only will admit points that are rotated, but will also tolerate other deformations as permitted by the threshold. It is placed to reflect the rationale that similar images will have similar responses but not necessarily under a rigid deformation of the query points.

Experiments. The database used for the local similarity retrieval has digitized images of cars, steam locomotives, diesel locomotives, apes, faces, people embedded in different backgrounds, and a small number of other miscellaneous objects such as houses. 1561 images were obtained from the Internet and the Corel photograph—CD collection to construct this database. These photographs were taken with several different cameras of unknown parameters, and under varying uncontrolled lighting and viewing geometry.

Prior to describing the experiments, it is important to clarify what a correct retrieval means. A retrieval system is expected to answer queries such as "find all cars similar in view and shape to this car" or "find all faces similar in appearance to this one." In the examples presented here the following method of evaluation is applied. First, the objective of the query is stated, and then retrieval instances are gauged against the stated objective. In general, objectives of the form "extract images similar in appearance to the query" will be posed to the retrieval algorithm.

A measure of the performance of the retrieval engine may be obtained by examining the recall–precision table for several queries. Briefly, recall is the proportion of the relevant material actually retrieved, and precision is the proportion of retrieved material that is relevant (35). Consider as an example the query described in Fig. 3(a). Here the user wishes to retrieve white-wheeled cars similar to the one outlined and submits the query. The top 25 results, ranked in textbook fashion, are shown in Fig. 3(b). Note that although there are several valid matches as far as the algorithm is concerned (for example, image 12 is a train with a wheel), they are not considered valid retrievals as stated by the user and are not used in measuring the recall and precision. This yields an

Table 1. Queries Submitted to the System and Expected Retrieval

Given (User Input)	Find	Precision (%) 3 pixels	5 pixels
Face	All faces	74.7	61.5
Face	Same person's face	61.7	75.5
Ape's coat [Fig. 6(a)]	Dark-textured apes [Fig. 6(b)]	57.5	57
Both wheels	White-wheeled cars	57.0	63.7
Coca-Cola logo	All Coca-Cola logos	49.3	74.9
Wheel [Fig. 3(a)]	White-wheeled cars [Fig. 3(b)]	48.6[a]	54.4
Patas monkey face	All vislble patas monkey faces	44.5	47.1

[a] See text.

inherently conservative estimate of the performance of the system. The average precision (over recall intervals of 10) is 48.6%. [The quantity n (= 10) is simply the number of retrievals up to recall n.]

One of the important parameters in constructing indices is the sample rate. Recall that indices are generated by computing multiscale invariant feature vectors at uniformly sampled locations within the image. The performance of the system is evaluated under sample rates of 3 pixels and 5 pixels. The case where every pixel is used could not be implemented due to prohibitive disk requirements and lack of resources to do so. Six other queries that were also submitted are depicted in Table 1. The recall–precision table over all seven queries is in Table 2. The second column of the table shows the average precision for each query with a database sampling of 5 pixels, while the third column displays the average precision for a sampling of 3 pixels. This compares well with text retrieval, where some of the best systems have an average precision of 50% (according to personal communication with Bruce Croft). The average precision over the same seven queries is 56.2% for the 5 pixel case and 61.7% for the 3 pixel case. However, while the increase in sampling improves the precision, it results in an increased storage requirement.

Unsatisfactory retrieval occurs for several reasons. First, it is possible that the query is poorly designed. In this case the user can design a new query and resubmit. A second source of error is in the matching itself. It is possible that locally the intensity surface may have very close values. Many of these false matches are eliminated in the spatial checking phase. Errors may also occur in the spatial checking phase because it admits much more than a rotational transformation of points with respect to the query configuration. Overall, the performance to date has been very satisfactory, and we believe that by experimentally evaluating each phase the system can be further improved. The time it takes to retrieve images depends linearly on the number of query points. On a Pentium Pro 200 MHz Linux machine, typical queries execute in between 1 and 6 min.

The primary limitations of the local matching technique are that it is relatively slow and that it requires considerable disk space. Further, as presented the system cannot search for images in their entirely. That is, it does not address global similarity.

Table 2. Precision at Standard Recall Points for Seven Queries

Recall	Precision (%) 5 pixels	3 pixels
0	100	100
10	95.8	100
20	90.3	90.4
30	80.1	80.9
40	67.3	75.7
50	48.9	55.9
60	39.9	49.4
70	34.2	47.6
80	31.1	40.6
90	18.2	20.7
100	12.4	17.1
Average	56.2	61.7

Global-Similarity Retrieval

The same Gaussian derivative model may be used to efficiently retrieve by global similarity of appearance. Since the task is to find similarity of whole images, significant improvements in space as well as speed may be achieved by representing images using distributions of feature vectors as opposed to the vectors themselves. One of the simplest ways of representing a nonparametric distribution is a histogram. Thus, a histogram of features may be used.

There are several features that may be exploited. Here the task is to robustly characterize the 3-D intensity surface. A 3-D surface is uniquely determined if the local curvatures everywhere are known. Thus, it is appropriate that one of the features be local curvature. The principal curvatures of the intensity surface are differential invariants. Further, they are invariant to monotonic intensity variations, and their ratios are in principle insensitive to scale variations of the entire image. However, spatial orientation information is lost when constructing histograms of curvature (or ratios thereof) alone. Therefore we augment the local curvature with local phase, and the representation uses histograms of local curvature and phase.

Computing the Global Similarity. Three steps are involved in computing global similarity. First, local derivatives are computed at several scales. Second, derivative responses are combined to generate local features, namely, the principal curvatures and phases, and their histograms are generated. Third, the 1-D curvature and phase histograms generated at several scales are matched. These steps are described next.

Computing Local Derivatives. Derivatives are computed stably using the formulation shown in Eq. (3). The first and second derivatives are computed at several scales by filtering the database images with Gaussian derivatives.

Feature Histogram. The normal and tangential curvatures of a 3D surface (X, Y, intensity) are defined by (8)

$$N(\boldsymbol{p},\sigma) = \frac{I_x^2 I_{yy} + I_y^2 I_{xx} - 2I_x I_y I_{xy}}{(I_x^2 + I_y^2)^{3/2}}\,(\boldsymbol{p},\sigma)$$

$$T(\boldsymbol{p},\sigma) = \frac{(I_x^2 - I_y^2) I_{xy} + (I_{xx} - I_{yy}) I_x I_y}{(I_x^2 + I_y^2)^{3/2}}\,(\boldsymbol{p},\sigma)$$



Figure 4. Image retrieval using curvature and phase.

where $I_x(\boldsymbol{p}, \sigma)$ and $I_y(\boldsymbol{p}, \sigma)$ are the local derivatives of the image I around point $\boldsymbol{p}$ using Gaussian derivatives at scale σ. Similarly, $I_{xx}(\cdot, \cdot)$, $I_{xy}(\cdot, \cdot)$, and $I_{yy}(\cdot, \cdot)$ are the corresponding second derivatives. The normal curvature N and tangential curvature T are then combined (36) to generate a shape index as follows:

$$C(\boldsymbol{p}, \sigma) = \arctan\left(\frac{N+T}{N-T}\right)(\boldsymbol{p}, \sigma)$$

The index value C is $\pi/2$ when $N = T$, and is undefined and therefore not computed when N and T are both zero. This is interesting because very flat portions of an image (or ones with constant ramp) are eliminated. For example in Fig. 4 (second row), the background in most of these face images does not contribute to the curvature histogram. The curvature index or shape index is rescaled and shifted to the range [0, 1], as is done in Ref. 37. A histogram is then computed of the valid index values over an entire image.

The second feature used is phase. The phase is simply defined as $P(\boldsymbol{p}, \sigma) = \arctan 2[I_y(\boldsymbol{p}, \sigma)/I_x(\boldsymbol{p}, \sigma)]$. Note that P is defined only at those locations where C is defined, and ignored elsewhere. As with the curvature index, P is rescaled and shifted to lie in the interval [0, 1].

Although the curvature and phase histograms are in principle insensitive to variations in scale, in early experiments we found that computing histograms at multiple scales dramatically improved the results. An explanation for this is that at different scales different local structures are observed, and therefore multiscale histograms are a more robust representation. Consequently, a feature vector is defined for an image I as the vector $\boldsymbol{V}_i = \langle H_c(\sigma_1), \ldots, H_c(\sigma_n), H_p(\sigma_1), \ldots, H_p(\sigma_n)\rangle$, where H_p and H_c are the curvature and phase histograms respectively. We found that using five scales gives good results, and the scales used were from 1 to 4 in steps of half an octave.

Matching Feature Histograms. Two feature vectors are compared using the normalized cross-covariance defined as

$$d_{ij} = \frac{\boldsymbol{V}_i^{(m)} \cdot \boldsymbol{V}_j^{(m)}}{\|V_i^{(m)}\| \, \|V_j^{(m)}\|}$$

where $\boldsymbol{V}_i^{(m)} = \boldsymbol{V}_i - \text{mean}(\boldsymbol{V}_i)$.

Retrieval is carried out as follows. A query image is selected, and the query histogram vector $\boldsymbol{V}_j$ is correlated with the database histogram vectors $\boldsymbol{V}_i$ using the above formula. Then the images are ranked by their correlation score and displayed to the user. In this implementation, and for evaluation purposes, the ranks are computed in advance, since every query image is also a database image.

Experiments. The curvature–phase method was tested using two databases. The first is a trademark database of 2048 images obtained from the US Patent and Trademark Office (PTO). The images obtained from the PTO are large and binary, and were converted to gray level and reduced for the experiments. The second database is the collection of 1561 assorted gray-level images used for the local-similarity case.

In the following experiments an image is selected and submitted as a query. The objective of this query is stated and the relevant images are decided in advance. Then the retrieval instances are gauged against the stated objective. In general, objectives of the form "extract images similar in appearance to the query" will be posed to the retrieval algorithm. The measure of the performance of the retrieval engine is obtained by examining the recall–precision table for several queries.

Queries were submitted to each of the collections (trademark and assorted image collection) separately for the purpose of computing recall and precision. The judgment of relevance is qualitative. For each query in both databases the

Table 3. Precision at Standard Recall Points for Six Queries

	Precision (%)	
Recall	Trademark	Assorted
0	100	100
10	93.2	92.7
20	93.2	90.0
30	85.2	88.3
40	76.3	87.0
50	74.5	86.8
60	59.5	83.8
70	45.5	65.9
80	27.2	21.3
90	9.0	12.0
100	9.0	1.4
Average	61.1	66.3

relevant images were decided in advance. These were restricted to 48. The top 48 ranks were then examined to check the proportion of retrieved images that were relevant. All images not retrieved within 48 were assigned a rank equal to the size of the database. That is, they are not considered retrieved. These ranks were used to interpolate and extrapolate the precision at all recall points. In the case of assorted images, relevance is easier to determine and more similar for different users. However, in the trademark case it may be quite difficult to determine relevance, and therefore the recall and precision may be subject to some error. The recall–precision results are summarized in Table 3, and both databases are individually discussed below.

Figure 5 shows the performance of the algorithm on the trademark images. Each strip depicts the top eight retrievals, given the leftmost image as the query. Most of the shapes have roughly the same structure as the query. Six queries were submitted for the purpose of computing recall and precision depicted in Table 3.

Experiments were also carried out with assorted gray-level images. Six queries submitted for recall and precision are shown in Fig. 4. The leftmost image in each row is the query and is also the first retrieved. The rest, from left to right, are seven retrievals depicted in rank order. Flat portions of the background are never considered, because the principal curvatures are very close to zero and therefore do not contribute to the final score. Thus, for example, the flat background in Fig. 4 (second row) is not used. Notice that visually similar images are retrieved even when there is some change in the background (first row). This is because the dominant object contributes most to the histograms. On using a single scale, poorer results are achieved and background affects the results more significantly.

The results of these and other examples are discussed below, with the average precision over all recall points depicted in parentheses:

1. *Find Similar Cars (65%).* Pictures of cars (Fig. 3) viewed from similar orientations appear in the top ranks because of the contribution of the phase histogram. This result also shows that some background variation can be tolerated. The eight retrieval, although a car, is a mismatch and is not considered a valid retrieval for the purpose of computing recall and precision.
2. *Find Same Face (87.4%) and Find Similar Faces.* In the face query (Fig. 4, second row) the objective is to find the same face. In experiments with a University of Bern face database of 300 faces with 10 relevant faces each, the average precision over all recall points for all 300 queries was 78%. It should be noted that the system presented here works well for faces with the same representation and parameters used for all the other databases. There is no specific "tuning" or learning involved to retrieve faces. The query "find similar faces" resulted in a 100% precision at 48 ranks because there are far more faces than 48. Therefore it was not used in the final precision computation.
3. *Find Dark-Textured Apes (64.2%).* The ape query (Fig. 6) results in several light-textured apes and country scenes with similar texture. Although these are not mismatches, they are not consistent with the intent of the query, which was to find dark-textured apes.
4. *Find Other Patas Monkeys (47.1%).* Here there are 16 Patas monkeys in all and 9 within a small view varia-

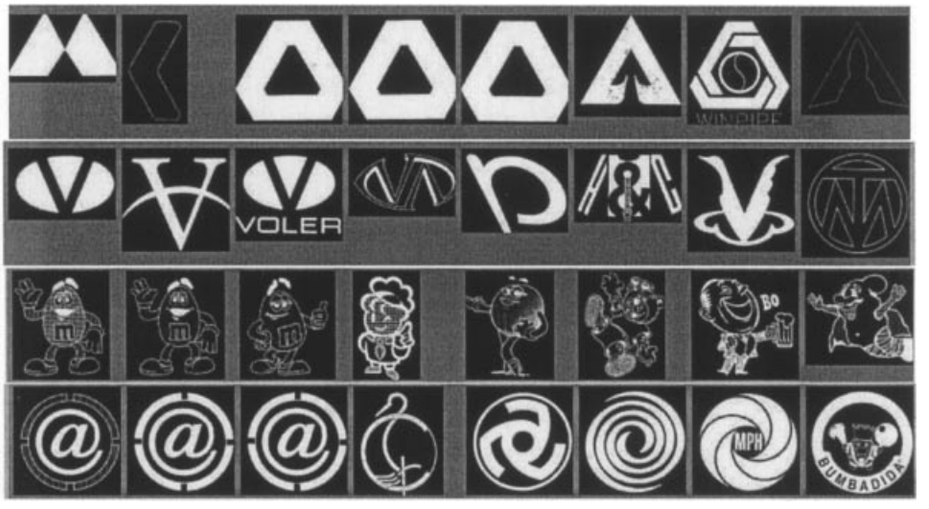
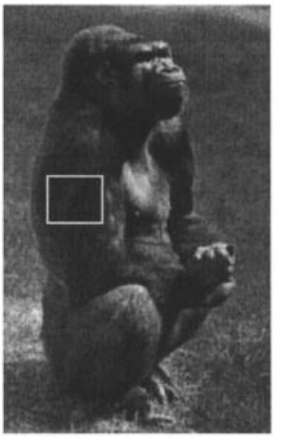


Figure 5. Trademark retrieval using curvature and phase.



(a)



(b)

Figure 6. A query and its retrieval: (a) ape query; (b) ranked retrieval.

tion. However, here the whole image is being matched, so the number of relevant Patas monkeys is 16. The precision is low because the method cannot distinguish between light and dark textures, leading to irrelevant images. Note that it finds other apes (dark-textured ones), but those are deemed irrelevant with respect to the query.

5. *Given a Wall with a Coca-Cola Logo, Find Other Coca-Cola Images (63.8%).* This query (Fig. 4, last row) clearly depicts the limitation of global matching. Al-

though all three database images that had a certain texture of the wall (and also had Coca-Cola logos) were retrieved (100% precision), two other very dissimilar images with Coca-Cola logos were not.

6. *Scenes with Bill Clinton (72.8%).* The retrieval in this case (Fig. 4, fifth row) results in several mismatches. However, three of the four are retrieved in succession at the top, and the scenes appear visually similar.

While the queries presented here are not optimal with respect to the design constraints of global similarity retrieval, they are realistic queries that can be posed to the system. Mismatches can and do occur. The first is the case where the global appearance is very different. The Coca-Cola retrieval is a good example of this. Second, mismatches may occur at the algorithmic level. Histograms represent spatial information coarsely and therefore will admit images with nontrivial deformations. The recall and precision presented here compare well with text retrieval. The time per retrieval is of the order of milliseconds. In ongoing work we are experimenting with a database of 63,000 images, and the amount of time taken to retrieve is still less than a second. The space required is also a small fraction of the database. These are the primary advantages of global-similarity retrieval: to provide low-storage, high-speed retrieval with good recall and precision.

Suggested Reading

Image retrieval has attracted the attention of several researchers in recent years, and several retrieval systems have been proposed. The earliest general image retrieval systems were designed by Flickner et al. (38) and Pentland et al. (39). In Ref. 38 the shape queries require prior manual segmentation of the database, which is undesirable and not practical for most applications.

Several authors have tried to characterize the appearance of an object via a description of the intensity surface. In the context of object recognition (40) one represents the appearance of an object using a parametric eigenspace description. This space is constructed by treating the image as a fixed-length vector and then computing the principal components across the entire database. The images, therefore, have to be size- and intensity-normalized, segmented, and trained. Similarly, using principal component representations described in Ref. 41, face recognition is performed in Ref. 42. In Ref. 43 the traditional eigenrepresentation is augmented by using more discriminant features and is applied to image retrieval. The authors apply eigenrepresentation to retrieval of several classes of objects. The issue, however, is that these classes are manually determined and training must be performed on each. The approach presented in this article is different from all the above in that eigendecompositions are not used at all to characterize appearance. Further, the method presented uses no learning, does not depend on constant-size images, and deals with embedded backgrounds (local similarity) and heterogeneous collections of images using local representations of appearance.

The method presented in Ref. 18 is similar to the local-similarity algorithm. However, there are some differences. Schmid and Mohr's algorithm does not allow the user to select query regions and relies on corners as features. The algorithm used to spatially compare the query vectors with those of a database image (spatial consistency) is also different. The motivation behind the algorithm presented here is that algorithms such as feature detection or segmentation, which are used to determine salient parts of an image, cannot be determined *a priori* in a general retrieval system, because it is not possible to define in advance the needs of a user. Thus instead of establishing an *a priori* bias towards any feature, images are uniformly sampled. The selection of salient portions of an image is obtained in the form of the user-defined query. In addition we find that using the lowest two orders rather than three orders gives better results in locating similar features.

With regard to the global retrieval algorithm, Schiele and Crowley (44) used a technique based on histograms for recognizing objects in gray-level images. Their technique used the outputs of Gaussian derivatives as local features. Several feature combinations were evaluated. In each case, a multidimensional histogram of these local features is then computed. Two images are considered to be of the same object if they had similar histograms. The difference between our approach and the one presented by Schiele and Crowley is that here we use ID (as opposed to multidimensional) histograms and further use the principal curvatures (which they do not use) as the primary feature.

Texture-based image retrieval is also related to the appearance-based work presented in this article. Using Wold modeling, Liu and Picard (45) try to classify the entire Brodatz texture, and Gorkani and Pickard (46) attempt to classify scenes, such as city and country. Of particular interest is work by Ma and Manjunath (47), who use Gabor filters to retrieve similar-texture images, without user interaction to determine region saliency.

ACKNOWLEDGMENT

The authors wish to thank Bruce Croft and the Center for Intelligent Information Retrieval for continued support of this work. This work is made possible due to efforts by Adam Jenkins, David Hirvonen, and Yasmina Chitti.

This material is based on work supported in part by the National Science Foundation, Library of Congress, and Department of Commerce under cooperative agreement number EEC-9209623, in part by the United States Patent and Trademark Office and Defense Advanced Research Projects Agency/ITO under ARPA order number D468, issued by ESC/AXS contract number F19628-95-C-0235, in part by the National Science Foundation under grant number IRI-9619117, and in part by NSF Multimedia CDA-9502639. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the sponsors.

BIBLIOGRAPHY

1. D. Marr and E. Hildreth, Theory of edge detection, *Proc. R. Soc. London B,* **B207**: 1980.
2. P. Burt and T. Adelson, The laplacian pyramid as a compact image code, *IEEE Trans. Commun.,* **COM-31**: 532–540, 1983.

3. J. L. Crowley and A. C. Parker, A representation for shape based on peaks and ridges in the difference of low-pass transform, *IEEE Trans. Pattern Anal. Mach. Intell.,* **6**: 156–169, 1984.
4. J. J. Koenderink, The structure of images, *Biol. Cybern.,* **50**: 363–396, 1984.
5. J. J. Koenderink and A. J. van Doorn, Representation of local geometry in the visual system, *Biol. Cybern.,* **55**: 367–375, 1987.
6. T. Lindeberg, *Scale-Space Theory in Computer Vision,* Norwell, MA: Kluwer, 1994.
7. B. M. ter Har Romeny, *Geometry Driven Diffusion in Computer Vision,* Norwell, MA: Kluwer, 1994.
8. L. M. J. Florack, *The syntactic structure of scalar images,* Ph.D. Thesis, University of Utrecht, 1993.
9. S. Ravela and R. Manmatha, Retrieving images by appearance, *IEEE Int. Conf. Comput. Vision,* 1998.
10. S. Haykin, *Communication Systems,* New York: Wiley, 1978.
11. A. P. Witkin, Scale-space filtering, *Proc. Int. Joint Conf. Artificial Intell.,* 1983, pp. 1019–1023.
12. R. Manmatha, Measuring the affine transform—I: Scale and rotation. *Proc. Comput. Vision Pattern Recognition Conf.,* 1993, pp. 754–755.
13. R. Manmatha and J. Oliensis, Measuring affine transform—I, scale and rotation, *Proc. DARPA Image Understanding Workshop,* Washington, 1993, pp. 449–458.
14. R. Rao and D. Ballard, Object indexing using an iconic sparse distributed memory, *Proc. Int. Conf. Comput. Vision,* 1995, pp. 24–31.
15. S. Ravela, R. Manmatha, and E. M. Riseman, Image retrieval using scale-space matching, *Computer Vision—ECCV '96, 4th Eur. Conf. Comput. Vision,* Cambridge, UK: Springer-Verlag, 1996, pp. 273–282.
16. W. T. Freeman and E. H. Adelson, The design and use of steerable filters, *IEEE Trans. Pattern Anal. Mach. Intell.,* **13**: 891–906, 1991.
17. S. Ravela et al., Adaptive tracking and model registration across distinct aspects, *Int. Conf. Intell. Robots Syst.,* 1995, vol. 1, pp. 174–180.
18. C. Schmid and R. Mohr, Combining greyvalue invariants with local constraints for object recognition, *Proc. Comput. Vision Pattern Recognition Conf.,* 1996.
19. R. Hummel and D. Lowe, Computational considerations in convolution and feature extraction in images, in J. C. Simon (ed.), *From Pixels to Features,* Amsterdam, The Netherlands: Elsevier, 1989, pp. 91–102.
20. R. Manmatha, *Matching affine-distorted images,* Ph.D. Thesis, Univ. Massachusetts, Amherst, 1997.
21. E. C. Hildreth, *The Measurement of Visual Motion,* Cambridge, MA: MIT Press, 1984.
22. G. Granlund and H. Knutsson, *Signal Processing for Computer Vision,* Norwell, MA: Kluwer, 1995.
23. R. A. Young, The Gaussian derivative model for spatial vision: I. Retinal mechanisms, *Spatial Vision,* **2** (4): 273–293, 1987.
24. H. S. Sawhney, S. Ayer, and M. Gorkani, Model-based 2D and 3D dominant motion estimation for mosaicing and video representation, *Proc. 5th Int. Conf. Comput. Vision,* 1995, pp. 583–590.
25. M. Irani, P. Anandan, and S. Hsu, Mosaic based representations of video sequences and their applications, *Proc. 5th Int. Conf. Comput. Vision,* 1995, pp. 605–611.
26. D. G. Jones and J. Malik, A computational framework for determining stereo correspondence from a set of linear spatial filters, *Proc. 2nd Eur. Conf. Comput. Vision,* 1992, pp. 395–410.
27. J. R. Bergen et al., Hierarchical model-based motion estimation, *Proc. 2nd Eur. Conf. Comput. Vision,* 1992, pp. 237–252.
28. M. Campani and A. Verri, Motion analysis from optical flow, *Comput. Vision Graph. Image Process. Image Understanding,* **56** (12): 90–107, 1992.
29. J. Shi and C. Tomasi, Good features to track, *Proc. Comput. Vision Pattern Recognition Conf.,* 1994, pp. 593–600.
30. P. Werkhoven and J. J. Koenderink, Extraction of motion parallax structure in the visual system I, *Biol. Cybern.,* 1990.
31. R. Cipolla and A. Blake, Surface orientation and time to contact from image divergence and deformation, *Proc. 2nd Eur. Conf. Comput. Vision,* 1992, pp. 187–202.
32. H. S. Sawhney and A. R. Hanson, Identification and 3D description of "shallow" environmental structure in a sequence of images, *Proc. Comput. Vision Pattern Recognition Conf.,* 1991, pp. 179–186.
33. J. Segman, J. Rubinstein, and Y. Y. Zeevi, The canonical coordinates method for pattern deformation: Theoretical and computational considerations, *IEEE Trans. Pattern Anal. Mach. Intell.,* **14**: 1171–1183, 1992.
34. R. Manmatha, A framework for recovering affine transforms using points, lines or image brightnesses, *Proc. Comput. Vision Pattern Recognition Conf.,* 1994, pp. 141–146.
35. C. J. van Rijsbergen, *Information Retrieval,* London: Butterworth, 1979.
36. J. J. Koenderink and A. J. Van Doorn, Surface shape and curvature scales, *Image and Vision Comput.,* **10** (8): 1992.
37. C. Dorai and A. Jain, Cosmos—a representation scheme for free form surfaces, *Proc. 5th Int. Conf. Comput. Vision,* 1995, pp. 1024–1029.
38. M. Flickner et al., Query by image and video content: The qbic system, *IEEE Comput. Mag.,* **28** (9): 23–30, 1995.
39. A. Pentland, R. W. Picard, and S. Sclaroff, Photobook: Tools for content-based manipulation of databases, *Proc. Storage Retrieval Image and Video Databases II,* SPIE, 1994, vol. 2, pp. 34–47.
40. S. K. Nayar, H. Murase, and S. A. Nene, Parametric appearance representation, in *Early Visual Learning,* London: Oxford Univ. Press, 1996.
41. M. Kirby and L. Sirovich, Application of the Karhunen–Loeve procedure for the characterization of human faces, *IEEE Trans. Pattern Anal. Mach. Intell.,* **12**: 103–108, 1990.
42. M. Turk and A. Pentland, Eigenfaces for recognition. *J. Cognitive Neurosci.,* **3**: 71–86, 1991.
43. D. L. Swets and J. Weng, Using discriminant eigenfeatures for retrieval, *IEEE Trans. Pattern Anal. Mach. Intell.,* **18**: 831–836, 1996.
44. B. Schiele and J. L. Crowley, Object recognition using multidimensional receptive field histograms, in B. Buxton and R. Cipolla (eds.), *Computer Vision—ECCV '96,* Lecture Notes in Computer Science 1, Cambridge, UK, *Proc. 4th European Conf. Comput. Vision,* New York: Springer-Verlag, 1996.
45. F. Liu and R. W. Picard, Periodicity, directionality, and randomness: Wold features for image modeling and retrieval, *IEEE Trans. Pattern Anal. Mach. Intell.,* **18**: 722–733, 1996.
46. M. M. Gorkani and R. W. Picard, Texture orientation for sorting photos "at a glance," *Proc. 12th Int. Conf. Pattern Recognition,* Oct. 1994, pp. A459–A464.
47. W. Y. Ma and B. S. Manjunath, Texture-based pattern retrieval from image databases, *Multimedia Tools Appl.,* **2** (1): 35–51, 1996.

S. RAVELA
R. MANMATHA
University of Massachusetts

GAUSSIAN WHITE NOISE. See KALMAN FILTERS AND OBSERVERS.

GENERALIZATION. See ARTIFICIAL INTELLIGENCE, GENERALIZATION.

GENERATING SET. See DIESEL-ELECTRIC POWER STATIONS.

GENERATION OF NOISE. See NOISE GENERATORS.

GENERATOR (OSCILLATOR), PUMP. See MICROWAVE PARAMETRIC AMPLIFIERS.

GENERATOR, RAMP. See RAMP GENERATOR.

GENERATORS, AC. See TURBOGENERATORS.

GENERATORS, DC. See DC MACHINES.

GENERATORS, DIESEL-ELECTRIC. See DIESEL-ELECTRIC GENERATORS.

GENERATORS, TURBINE. See TURBOGENERATORS.

Geometry
Standard Article
Mysore Narayanan[1]
[1]Miami University, Oxford, OH
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2419
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (115K)

Abstract

The sections in this article are

Analytic Geometry

Projective and Analytic Geometry

Differential Geometry

Copyright © 1999-2008 All Rights Reserved.

GEOMETRY

Geometry is a comprehensive branch of mathematics that discusses the properties of space and objects in space. Geometry deals with points and with the measurement and the relationship of lines, angles, two-dimensional surfaces, and three-dimensional solids. The word *geometry* is derived from the Greek words for "earth" and "measure," and the Greeks should be credited for developing geometry as a systematic science. Geometric figures have been found in cave paintings and pottery that date back as far as 12,500 B.C. The Egyptians had excellent capabilities with regard to applying

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

geometry (e.g., to build pyramids), but they did not lay out the laws, rules, and theorems systematically by modern standards. Greek philosophers, mathematicians, and astronomers provided a strong mathematical foundation for the empirical art and transformed it to a systematic science. Among the Greek pioneers, Pythagoras (ca. 540 B.C.), arranged sets of objects in geometrical shapes (Fig. 1), Plato (ca. 400 B.C.) tried to explain the nature of the universe using geometrical principles, and his pupil Aristotle developed laws for logical reasoning.

Euclid, who lived in Alexandria (ca. 330 to 275 B.C.), finally systematized all the work of Pythagoras, Plato, Aristotle and other learned scientists and philosophers in his book *Elements.* Euclid's mathematical accomplishments were taught in Plato's academy, and *Elements* was considered to be the one true geometry for almost two thousand years. Euclid is also supposed to have written four books on conic sections, but that work has been completely lost.

Euclid is considered to be one of the most influential mathematicians of all time. His greatness lies in the fact that he developed geometry logically as an exact science and that his work has been studied for more than 24 centuries all over the world. A complete text of all 13 books of Euclid can be read and studied using a Java applet called *Geometry Applet* (1). Another Greek mathematician, Menaechmus, first studied conic sections in fourth century B.C. Archimedes (ca. 287 to 212 B.C.) used Euclid's principles and found the area of an ellipse, the area of a sector of a parabola, the volume of a sphere, and the like. Apollonius (ca. 225 B.C.) wrote eight books on conic sections (only seven have survived). Apollonius was the first to develop the concept of coordinates, which was later perfected by the French mathematician René Descartes (1596–1650). The modern study of geometry makes extensive use of the Cartesian system of coordinates, named after him.

Euclidean geometry can be viewed as a study of *congruent figures.* In his 13 books, Euclid set forth the elementary part of the deductive system of geometry. Many principles were axioms and regarded as obviously true. (Example: All right angles are equal.) Some were "common notions." (Example: The whole is greater than a part.) Some were derived from experience and eventually developed into powerful theorems that have established strong foundations for modern-day mathematicians.

For almost 2000 years Euclid's 13 books defined geometry, and all schools and colleges used Euclid's books in one form or the other. The *Elements* was translated from Greek to Arabic (ca. 800 A.D.), then to Latin (ca. 1120 A.D.), and was first translated into English at the end of the sixteenth century.

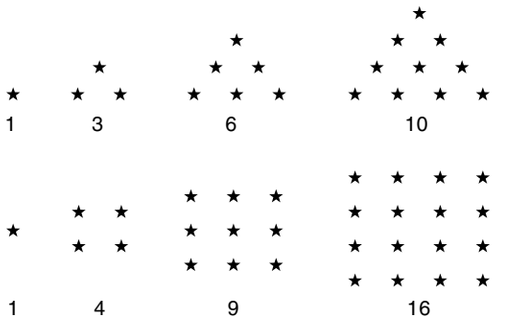


Figure 1. Objects arranged geometrically: 1, 3, 6, 10 are triangular numbers [$n(n + 1)/2$]; 1, 4, 9, 16 are square numbers (n^2).

The five chief axioms (or postulates) of Euclid are as follows:

1. A straight line can be drawn from any given point to another point: Given two points, there is an interval that joins them.
2. A finite straight line can be drawn continuously: An interval can be prolonged indefinitely.
3. A circle is determined by its center and its radius. A circle can be constructed when its center and a point on it are given.
4. All right angles are equal.
5. If a straight line falling on two straight lines makes the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which the angles are less than the two right angles.

The fifth postulate is the most famous of Euclid's axioms, and is called the *parallel axiom.* It is also the most controversial. Many questions have been raised concerning it. Euclid himself must have been aware of the fact that the fifth postulate is lengthy (compared with the first four) in making its statement. Euclid proved his first 28 propositions without making use of the fifth axiom. Mathematicians are intrigued by the fifth postulate and have tried in vain to prove it or disprove it. An equivalent statement, *Playfair's axiom,* states: "Through a point not on a given line, there passes not more than one parallel to the line."

The fifth axiom is considered inapplicable or invalid in *hyperbolic geometry,* because rays may converge at first, but later diverge after attaining a minimum distance. The fifth axiom is satisfied only marginally in the *elliptic geometry.* Various mathematicians have contributed to such non-Euclidean geometries. Some were attempting to verify the validity of the fifth postulate. The Italian logician Girolamo Saccheri; the German mathematicians Georg Simon Klügel, Carl Friedrich Gauss, and Friedrich Ludwig Wachter; the Russian Nikolay Ivanovich Lobachevsky; and the Hungarian father and son Farkas Bolyai and János Bolyai are among the most prominent contributors. Non-Euclidean geometry and the theory of relativity have dramatically changed the thinking of nineteenth- and twentieth-century scientists and mathematicians.

The collection of propositions based only on the first four axioms of Euclid is known as *absolute geometry.* It should be noted that in fact we might well be living in a non-Euclidean universe.

In addition to the above-mentioned five axioms, the five important "common notions" are listed below:

1. Things equal to the same thing are equal.
2. If equals are added to equals, the wholes are equal.
3. If equals are subtracted from equals, the remainders are equal.
4. Things that coincide with one another are equal.
5. The whole is greater than a part.

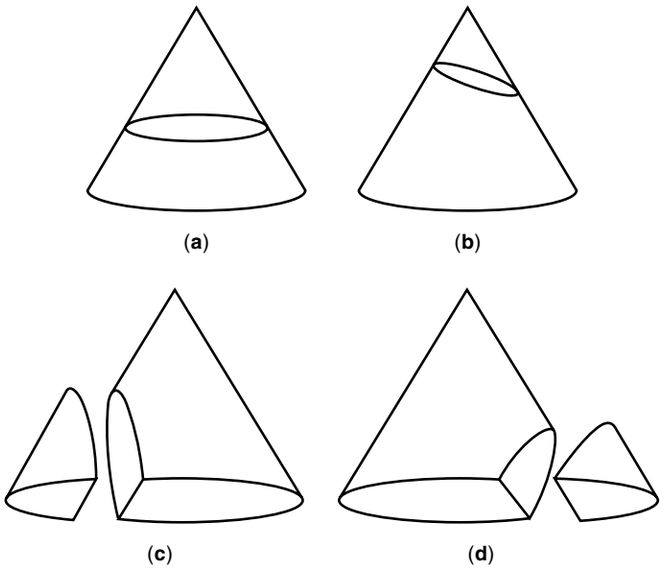


Figure 2. Conic sections: (a) circle, (b) ellipse, (c) hyperbola, and (d) parabola.

ANALYTIC GEOMETRY

In 1637, the French mathematician René Descartes applied algebra to geometry to calculate the dimensions of geometrical figures. The idea of negative distances developed by Sir Isaac Newton in the seventeenth century helped to perfect this new branch of geometry, also called *coordinate geometry,* wherein lines and curves could be represented by sets of equations. In two dimensions, an extensive study of mathematical expressions for conic sections was made possible. Figure 2 shows four important conic sections: the circle, ellipse, parabola, and hyperbola. Since Euclid, it is the German mathematician David Hilbert who has made the greatest impact on the mathematical world of geometry. His publication of *Grundlagen der Geometrie* in 1899 formalized the notions of hyperbolic geometry and elliptic geometry. A simplified view of the two non-Euclidean geometries can be seen in Fig. 3(a) and (b).

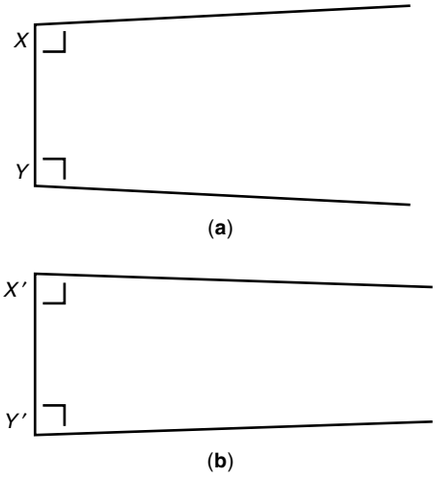


Figure 3. (a) Hyperbolic geometry: rays share perpendicular line, then diverge; (b) elliptic geometry: rays share perpendicular line then converge.

Consider two perpendicular rays emerging from the same side of a line that connects points X and Y. If two rays diverge, as they extend farther and farther from the line XY, then the geometry is said to be hyperbolic. If the two rays converge, then it is said to be elliptic. In considering this type of non-Euclidean universe, one has to develop intuition without the help of observation.

Hilbert, in his 21 axioms, generalized the foundations of geometry and replaced the classical concepts that had been derived from intuition. His generalization of classical geometry has found application in science and industry.

The German mathematicians Leonhard Euler and Gaspard Monge are credited with the development of *differential geometry* in the eighteenth century. They studied the geometry of curves and surfaces in space as an application of calculus. Another German mathematician, Bernhard Riemann, who had a thorough understanding of the limitations of Euclidean geometry, developed what is known as *double elliptic geometry.* This new geometry played a vital role in defining a four-dimensional Riemannian space and in developing the theory of relativity.

PROJECTIVE AND ANALYTIC GEOMETRY

The German astronomer Johannes Kepler extended the Euclidean plane to three dimensions and helped in the development of solid analytic geometry. Analytic geometry along with calculus enabled the development of a wide variety of special curves that are extremely useful in space-age applications. For example, a special case of the cubical parabola ($y = ax^3$) was studied by the German mathematician Gottfried Wilhelm Leibniz in 1675 and is shown in Fig. 4.

Some of the famous geometrical patterns and curves and shapes that are derived from the equation $r^x = a^x\theta$ are also shown in Fig. 5. When $x = 1$, we have $r = a\theta$ and the curve is known as the *spiral of Archimedes.* When $x = 2$ it is called Fermat's spiral and follows the equation $r^2 = a^2\theta$. Johann Bernoulli studied the case $x = -1$, or $a = r\theta$.

Another important curve follows an algebraic equation that is of the general form

$$y = e^{-0.5x}$$

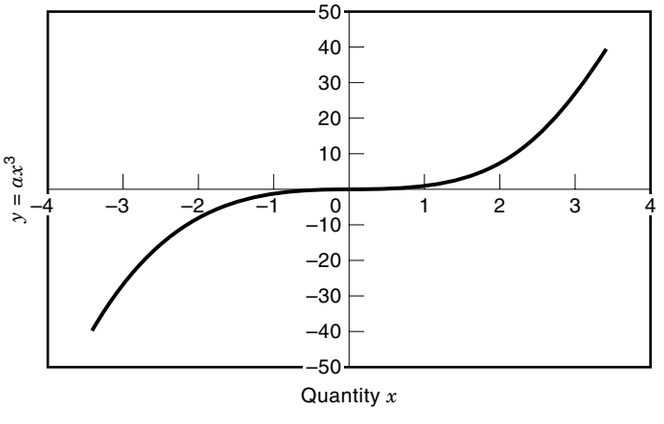


Figure 4. Cubical parabola.

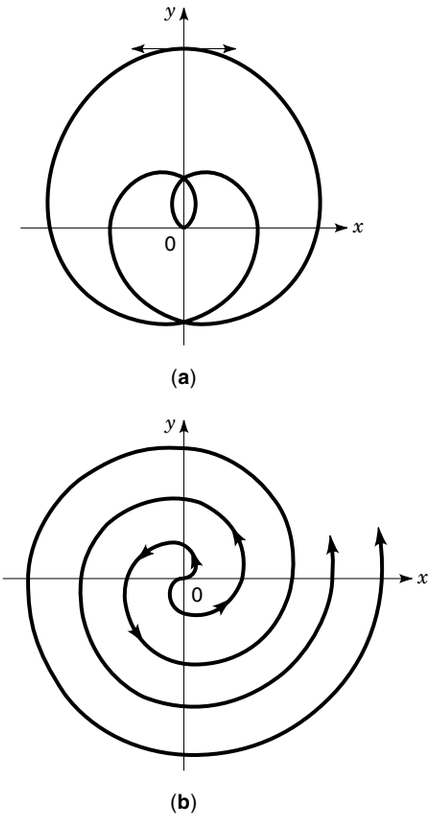


Figure 5. Geometrical patterns generated from equation $r^x = \alpha^x\theta$: (a) spiral of Archimedes ($r = \alpha\theta$); (b) Fermat's spiral ($r^2 = \alpha^2\theta$).

A version of this equation results in the well-known *bell-shaped curve* (Fig. 6). Although Abraham de Moivre, a French mathematician, was the first to study it, it is commonly known as the Gaussian distribution curve. It is associated with the French mathematician Pierre Simon Laplace as well.

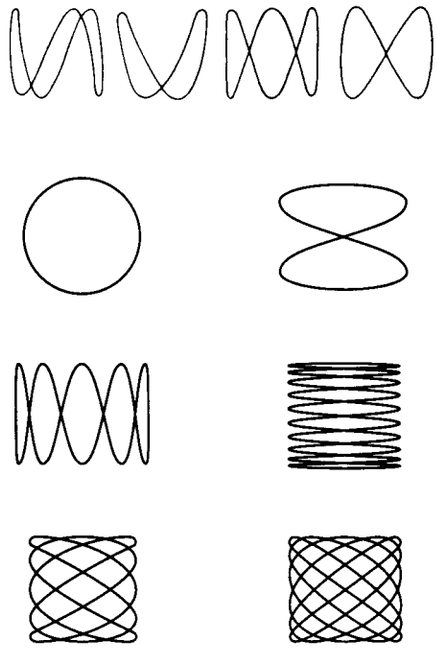


Figure 7. Bowditch curves or Lissajous figures.

Electrical engineers have greatly benefited from Lissajous figures, shown in Fig. 7. These are also known as Bowditch curves after Nathaniel Bowditch. By their use an oscilloscope can be used to compare frequencies. An electron beam can be deflected either in the x direction or in the y direction using suitable voltages at appropriate frequencies. If $x = A \sin \omega t$ and $y = B \sin(k\omega t + \phi)$, then the oscilloscope displays patterns as shown in Fig. 7 for different frequencies.

One of the most important of all special curves in the cycloid, which is obviously of great interest to engineers who study the motion of wheels. It follows the equations

$$x = a(\phi - \sin\phi) \quad \text{and} \quad y = a(1 - \cos\phi)$$

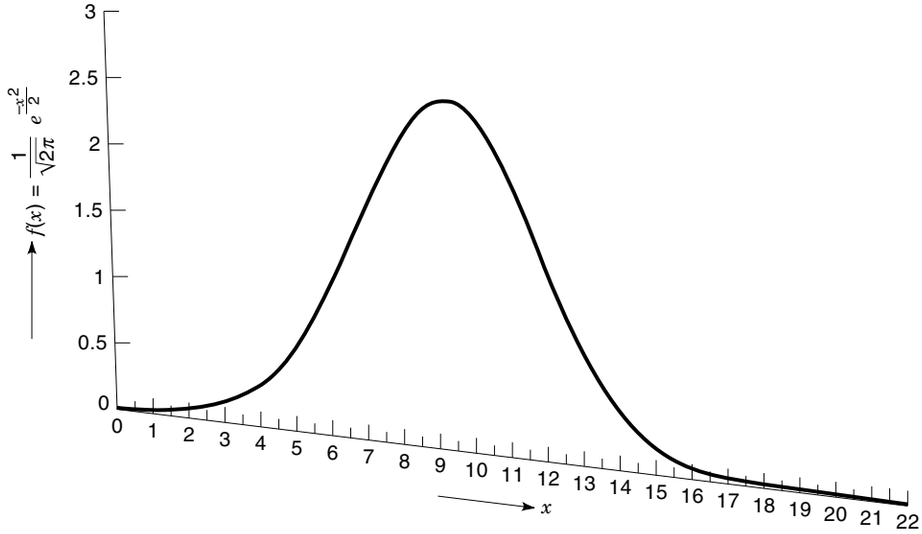


Figure 6. Bell-shaped curve or Gaussian distribution.

The curve is the path of a point on the circumference of a circle of radius a that rolls along a straight line without slipping or sliding. Galileo (1564–1642) was among the first to study the cycloid.

DIFFERENTIAL GEOMETRY

One can easily observe that the concept of the derivative of a function is similar to drawing a tangent to a curve and finding its slope at the given point. The area under a curve can almost always be determined by evaluating the integral of the equation defining the curve. Thus, differential calculus and integral calculus can be used to study the geometry of two-dimensional curves and three-dimensional surfaces. Leonhard Euler and Gaspard Monge are to be credited for the development of differential geometry. A clear understanding of the concepts of topology, elliptic operators, the Gauss–Bonnet formula, manifolds, and tensor bundles is essential for studying the principles of differential geometry in depth.

BIBLIOGRAPHY

D. E. Joyce, *Geometry Applet,* http://aleph0.clarku.edu/~djoyce/java/Geometry/Geometry.html.

Reading List

J. Stillwell, *Numbers and Geometry,* New York: Springer, 1998.

H. T. Croft, K. J. Falconer, and R. K. Guy, *Unsolved Problems in Geometry,* New York: Springer-Verlag, 1991.

G. Stephenson, *Worked Examples in Mathematics for Scientists and Engineers,* London: Longman, 1985.

G. Temple, *100 Years of Mathematics,* New York: Springer-Verlag, 1981.

J. Fang, *Mathematicians from Antiquity to Today,* Hauppauge, NY: Paideia Press, 1972.

D. E. Smith and J. Ginsburg, *A History of Mathematics in America Before 1900,* New York: Arno Press, 1980.

F. Cajori, *A History of Mathematics,* New York: Chelsea Publishing Co., 1980.

E. T. Bell, *The Development of Mathematics,* New York: McGraw-Hill, 1940.

R. Billstein and J. W. Lott, *Mathematics For Liberal Arts,* Reading, MA: Benjamin/Cummings, 1986.

R. A. Barnett and M. R. Ziegler, *Applied Mathematics,* 3rd Ed., San Francisco, CA: Dellen Publishing Co., 1989.

G. B. Arfken and H. J. Weber, *Mathematical Methods for Physicists,* 4th Ed., San Diego, CA: Academic Press, 1995.

C. E. Pearson (ed.), *Handbook of Applied Mathematics,* 2nd Ed., New York: Van Nostrand, 1983.

MYSORE NARAYANAN
Miami University

Graph Theory
Standard Article
Dimitrios Kagaris[1] and Spyros Tragoudas[2]
[1]Southern Illinois University, Carbondale, IL
[2]The University of Arizona, AZ
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2420
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (177K)

Abstract

The sections in this article are

Graph Theory Fundamentals

Algorithms and Time Complexity

Polynomial-Time Algorithms

Keywords: graphs; networks; combinatorial optimization problems; computational complexity; NP-completeness; graph algorithms; flow algorithms; operations research applications; VLSI CAD applications

| | |

Copyright © 1999-2008 All Rights Reserved.

GRAPH THEORY

GRAPH THEORY FUNDAMENTALS

A *graph* $G(V, E)$ consists of a set V of *vertices* (or *nodes*) and a set E of pairs of vertices from V, referred to as *edges.* An edge may have associated with it a direction, in which case the graph is called *directed* (as opposed to *undirected*), or a *weight,* in which case the graph is called *weighted.* Two vertices $u, v \in V$ for which an edge $e = (u, v)$ exists in E are said to be *adjacent* and edge e is said to be *incident* on them. The *degree* of a vertex is the number of edges adjacent to it. A (*simple*) *path* is a sequence of distinct vertices $(a_0, a_1, \ldots, a_k)$ of V such that every two vertices in the sequence are adjacent. A (*simple*) *cycle* is a sequence of vertices $(a_0, a_1, \ldots, a_k, a_0)$ such that $(a_0, a_1, \ldots, a_k)$ is a path and a_k, a_0 are adjacent. A graph is *connected* if there is a path between every pair of vertices. A graph $G'(V', E')$ is a *subgraph* of graph

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

$G(V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. A *spanning tree* of a connected graph $G(V, E)$ is a subgraph of G that comprises all vertices of G and has no cycles. Given a subset $V' \subseteq V$, the *induced subgraph* of $G(V, E)$ *by* V' is a subgraph $G'(V', E')$, where E' comprises all edges (u, v) in E with $u, v \in V'$.

In a directed graph, each edge (sometimes referred to as *arc*) is an ordered pair of vertices and the graph is denoted by $G(V, A)$. For an edge $(u, v) \in A$, v is called the *head* and u the *tail* of the edge. The number of edges for which u is a tail is called the *out-degree* of u and the number of edges for which u is a head is called the *in-degree* of u. A (*simple*) *directed path* is a sequence of distinct vertices $(a_0, a_1, \ldots, a_k)$ of V such that (a_i, a_{i+1}), $\forall i$, $0 \leq i \leq k - 1$, is an edge of the graph. A (*simple*) *directed cycle* is a sequence of vertices $(a_0, a_1, \ldots, a_k, a_0)$ such that $(a_0, a_1, \ldots, a_k)$ is a directed path and $(a_k, a_0) \in A$. A directed graph is *strongly connected* if there is a directed path between every pair of vertices. A directed graph is *weakly connected* if there is an undirected path between every pair of vertices.

Graphs are a very important modeling tool that can be used to model a great variety of problems in areas such as operations research [e.g., see (1,2,3)], very large scale integration (VLSI) computer-aided design (CAD) for digital circuits [e.g., see (4)], computer and communications networks [e.g., see (5)]. For example, in operations research a graph can be used to model the assignment of workers to tasks, the distribution of goods from warehouses to customers, etc. In VLSI CAD a graph can be used to represent a digital circuit at any abstract level of representation (gate level, module level, etc.). Each vertex in this case corresponds to a gate or module and each edge corresponds to a circuit line that connects the respective components. In computer and communications networks, a graph can be used to represent any given interconnection, with vertices representing host computers or routers and edges representing communication links.

There are many special cases of graphs. Some of the most common ones are listed below. A *tree* is a connected graph that contains no cycles. A *bipartite graph* is a graph with the property that its vertex set V can be partitioned into two disjoint subsets V_1 and V_2, $V_1 \cup V_2 = v$, such that every edge in E comprises one vertex from V_1 and one vertex from V_2. A *directed acyclic graph* is a directed graph that contains no directed cycles. Directed acyclic graphs can be used to represent combinational circuits in VLSI CAD. A *transitive graph* is a directed graph with the property that for any vertices u, v, $w \in V$ for which there exist edges (u, v), $(v, w) \in A$, edge (u, w) also belongs to A. A *planar graph* is a graph with the property that its edges can be drawn on the plane so as not to cross each other. A typical application of planar graphs is in VLSI, where the requirement is for all the circuit lines to be routed on a single layer.

A *cycle graph* is a graph that is obtained by a cycle with chords as follows: For every chord (a, b) of the cycle, there is a vertex $v_{(a,b)}$ in the cycle graph. There is an edge $(v_{(a,b)}, v_{(c,d)})$ in the cycle graph if and only if the respective chords (a, b) and (c, d) intersect. Cycle graphs find application in VLSI CAD as a channel with two terminal nets, or a switchbox with two terminal nets can be represented as a cycle graph. Then the problem of finding the maximum number of nets in the channel (or switchbox) that can be routed on the plane amounts to finding a maximum independent set in the respective cycle graph.

A *permutation graph* is a special case of a cycle graph. It is based on the notion of a *permutation diagram.* A permutation diagram is simply a sequence of N integers in the range from 1 to N (but not necessarily ordered). Given an ordering, there is a vertex for every integer in the diagram, and there is an edge (u, v) if and only if the integers u, v are not in the correct order in the permutation diagram. A permutation diagram can be used to represent a special case of a *permutable channel* in VLSI, where all nets have two terminals that belong to opposite channel sides. The problem of finding the maximum number of nets in the permutable channel that can be routed on the plane amounts to finding a maximum independent set in the respective permutation graph. This, in turn, amounts to finding the maximum increasing (or decreasing) subsequence of integers in the permutation diagram.

ALGORITHMS AND TIME COMPLEXITY

An *algorithm* is an unambiguous description of a finite set of operations for solving a computational problem in a finite amount of time. The set of allowable operations corresponds to the operations supported by a specific computing machine (computer) or to a model of that machine.

A *computational problem* comprises a set of parameters that have to satisfy a set of well-defined mathematical constraints. A specific assignment of values to these parameters constitutes an *instance* of the problem. For some computational problems there is no algorithm as defined above to find a solution. For example, the problem of determining whether an arbitrary computer program terminates in a finite amount of time given a set of input data cannot be solved (it is "undecidable") (6). For the computational problems for which there does exist an algorithm, the point of concern is how "efficient" that algorithm is. The *efficiency* of an algorithm is primarily defined in terms of how much time the algorithm takes to terminate. (Sometimes, other considerations such as the space requirement in terms of the physical information storage capacity of the computing machine are also taken into account, but in this exposition we concentrate exclusively on time.)

In order to formally define the efficiency of an algorithm, the following notions are introduced:

The *size* of an instance of a problem is defined as the total number of symbols for the complete specification of the instance under a finite set of symbols and a "succinct" encoding scheme. A "succinct" encoding scheme is considered to be a logarithmic encoding scheme, in contrast to a unary encoding scheme. The time requirement (*time complexity*) of an algorithm is expressed then as a function $f(n)$ of the size n of an instance of the problem and gives the total number of "basic" steps that the algorithm needs to go through to solve that instance. Most of the time, the number of steps is taken with regard to the worst case, although alternative measures like the average number of steps can also be considered. What constitutes a "basic" step is purposely left unspecified, provided that the time the basic step takes to be completed is bounded from above by a *constant,* that is, a value not dependent on the instance. This hides implementation details and machine-dependent timings and provides the required degree of general applicability.

An algorithm with time complexity $f(n)$ is said to be of the *order of* $g(n)$ [denoted as $O(g(n))$], where $g(n)$ is another function, if there is a constant c such that $f(n) \leq c \cdot g(n)$ for all $n \geq 0$. For example, an algorithm for finding the minimum element of a list of size n takes time $O(n)$, an algorithm for finding a given element in a sorted list takes time $O(\log n)$, algorithms for sorting a list of elements can take time $O(n^2)$, $O(n \log n)$, or $O(n)$ (the latter when additional information about the range of the elements is known). If moreover there are constants c_L and c_H such that $c_L \cdot g(n) \leq f(n) \leq c_H \cdot g(n)$ for all $n \geq 0$, then $f(n)$ is said to be $\Theta(g(n))$.

The smaller the "order-of" function, the more efficient an algorithm is generally taken to be, but in the analysis of algorithms, the term "efficient" is applied liberally to any algorithm whose "order-of" function is a polynomial $p(n)$. The latter includes time complexities like $O(n \log n)$ or $O(n\sqrt{n})$, which are clearly bounded by a polynomial. Any algorithm with a nonpolynomial time complexity is not considered to be efficient. All nonpolynomial-time algorithms are referred to as *exponential* and include algorithms with such time complexities as $O(2^n)$, $O(n!)$, $O(n^n)$, $O(n^{\log n})$ (the latter is sometimes referred to as *subexponential*). Of course, in practice, for an algorithm of polynomial time complexity $O(p(n))$ to be actually efficient, the degree of polynomial $p(n)$ as well as the constant of proportionality in the expression $O(p(n))$ should rather be small. In addition, because of the worst-case nature of the $O(\ )$ formulation, an "exponential" algorithm might exhibit exponential behavior in practice only in rare cases (the latter seems to be the case with the simplex method for linear programming). However, the fact on the one hand that most of the polynomial-time algorithms for the problems that occur in practice tend indeed to have small polynomial degrees and small constants of proportionality, and on the other that most nonpolynomial algorithms for the problems that occur in practice eventually resort to the trivial approach of exhaustively searching (enumerating) all candidate solutions, justifies the use of the term "efficient" for only the polynomial-time algorithms.

Given a new computational problem to be solved, it is of course desirable to find a polynomial-time algorithm to solve it. The determination of whether such a polynomial-time algorithm actually exists for that problem is a subject of primary importance. To this end, a whole discipline dealing with the classification of the computational problems and their interrelations has been developed.

P, NP, and *NP*-Complete Problems

The classification starts technically with a special class of computational problems known as decision problems. A computational problem is a *decision problem* if its solution can actually take the form of a yes or no answer. For example, the problem of determining whether a given graph contains a simple cycle that passes through every vertex is a decision problem (known as the Hamiltonian Cycle problem). In contrast, if the graph has weights on the edges and the goal is to find a simple cycle that passes through every vertex and has minimum sum of edge weights is not a decision problem, but an *optimization* problem (the latter problem is known as the Traveling Salesman problem). An optimization problem (sometimes referred to also as *combinatorial optimization problem*) seeks to find the best solution, in terms of a well-defined objective function $Q(\ )$, over a set of feasible solutions. Interestingly, every optimization problem has a "decision" version in which the goal of minimizing (or maximizing) the objective function $Q(\ )$ in the optimization problem corresponds to the question of whether there exists a solution with $Q(\ ) \leq k$ (or $Q(\ ) \geq k$) in the decision problem, where k is now an additional input parameter to the decision problem. For example, the decision version of the Traveling Salesman problem is, given a graph and an integer K, to find a simple cycle that passes through every vertex and whose sum of edge weights is no more than K.

All decision problems that can be solved in polynomial time comprise the so-called class P (for polynomial). Another established class of decision problems is the *NP* class which consists of all decision problems for which a polynomial-time algorithm can *verify* if a candidate solution (which has polynomial size with respect to the original instance) yields a yes or no answer. The initials *NP* stand for nondeterministic polynomial, in that if a yes answer exists for an instance of an *NP* problem, that answer can be obtained *nondeterministically* (in effect, guessed) and then verified in polynomial time. (The requirement for polynomial-time verification is readily met for most of the common problems, but there are problems, like the minimum equivalent expression (see below), for which this seems not to be the case.) Every problem in class P belongs clearly to *NP*, but the question of whether class *NP* strictly contains P or not is a famous unresolved problem. It is conjectured that $NP \neq P$, but there is no actual proof up to now. Notice that in order to simulate the nondeterministic guess in the statement above, an obvious deterministic algorithm would have to enumerate all possible cases, which is an exponential-time task. It is in fact the question of whether such an "obvious" algorithm is actually the best one can do that has not been resolved.

Showing that an *NP* decision problem actually belongs to P is equivalent to establishing a polynomial-time algorithm to solve that problem. In the investigation of the relations between problems in P and in *NP*, the notion of *polynomial reducibility* plays a fundamental role. A problem R is said to be *polynomially reducible* to another problem S if the existence of a polynomial-time algorithm for S implies the existence of a polynomial-time algorithm for R. That is, in more practical terms, if the assumed polynomial-time algorithm for problem S is viewed as a subroutine, then an algorithm that solves R by making a polynomially bounded number of calls to that subroutine and taking a polynomial amount of time for some extra work would constitute a polynomial-time algorithm for R.

There is a special class of *NP* problems with the property that if and only if *any one* of those problems could be solved polynomially, then so would *all* of the *NP* problems (i.e., *NP* would be equal to P). These *NP* problems are known as NP-*complete.* An *NP*-complete problem is an *NP* problem to which every other *NP* problem reduces polynomially. The first problem to be shown *NP*-complete was the Satisfiability problem (6). This problem concerns the existence of a truth assignment to a given set of boolean variables so that the conjunction of a given set of disjunctive clauses formed from these variables and their complements becomes true. The proof (given by Stephen Cook in 1971) was done by showing that every *NP* problem reduces polynomially to the Satisfiability problem. After the establishment of the first *NP*-complete case, an extensive

and on-going list of *NP*-complete problems has been established [see (6)]. The interest in showing that a particular problem R is *NP*-complete lies exactly in the fact that if it finally turns out that *NP* strictly contains P, then R cannot be solved polynomially (or, from another point of view, if a polynomial-time algorithm happens to be discovered for R, then $NP = P$). The process of showing that a decision problem is *NP*-complete involves showing that the problem belongs to *NP* and that some known *NP*-complete problem reduces polynomially to it. The difficulty of this task lies in the choice of an appropriate *NP*-complete problem to reduce from as well as in the mechanics of the polynomial reduction. An example of an *NP*-complete proof is given below.

We consider a problem that occurs in the testing of digital circuits (7): We are given a collection C of subsets of a set U with maximum subset size w, and an integer bound $k \leq w$. We want to determine whether there exists a mapping $f: U \rightarrow [1 \ldots |U|]$, so that for each set s in C, the corresponding set $s_f = \{f(a) \bmod k : a \in s\}$ has size $\min(|s|, k)$. This problem (referred to as the SETMOD problem) is *NP*-complete and this is established as follows (8):

Theorem 1 The SETMOD problem is *NP*-complete.

Proof. The problem belongs clearly to *NP*, since once a satisfying mapping f has been guessed, the verification can be done in polynomial (actually linear) time. We make the reduction from a known *NP*-complete problem that is called Not-All-Equal-3SAT (NAE) (6). The latter problem is a special case of the Satisfiability problem in which each disjunctive clause comprises exactly three literals and the goal is to find a truth assignment so that each clause has at least one true and at least one false literal. Let φ be a NAE instance with n variables $x_1, \ldots, x_n$ and m clauses $C_1, \ldots, C_m$. For each variable x_i, we consider the set $\{x_i, \bar{x}_i\}$, and for each clause $C_j = x_{j1} \vee x_{j2} \vee x_{j3}$, we consider the set $\{x_{j1}, x_{j2}, x_{j3}\}$ (each element in each clause is actually identical with some variable or its complement.) Set U is the set of all literals, that is, $|U| = 2n$.

Let $k = 2$. Suppose first that φ is satisfiable. For variable x_i, $1 \leq i \leq n$, we assign $f(x_i) = 2i - 1$, $f(\bar{x}_i) = 2i$ if the variable is true and $f(x_i) = 2i$, $f(\bar{x}_i) = 2i - 1$ if the variable is false. Since each clause has at least one true and one false literal, we have that each set s has at least one element with remainder 0 and at least one element with remainder 1 modulo $k = 2$, namely $|s'| = k = \min(|s|, k)$. Conversely, suppose that there is a solution for an instance of the problem in question that consists of the above collection of sets and $k = 2$. Then, since the three elements in each clause set cannot all be labeled odd or even, and also the two elements in each literal set cannot both be labeled odd or even, we have a satisfying assignment for the NAE instance.

Some representative *NP*-complete problems that occur in various areas in operations research, digital design automation, and computer networks are listed below.

- *3-Satisfiability.* Given a set of boolean variables and a set of disjunctive clauses over the variables, each one comprising exactly three literals, is there a satisfying assignment for the conjunction of all the clauses?
- *Hamiltonian Cycle.* Given a graph $G(V, E)$, does G contain a simple cycle that includes all the vertices of G?
- *Longest Path.* Given a graph $G(V, E)$ and an integer $K \leq |V|$, is there a simple path in G with at least K edges?
- *Vertex Cover.* Given a graph $G(V, E)$ and an integer $K \leq |V|$, is there a subset $V' \subseteq V$ such that $|V'| \leq K$ and for each edge $(u, v) \in E$, at least one of u, v belongs to V'?
- *Independent Set.* Given a graph $G(V, E)$ and an integer $K \leq |V|$, is there a subset $V' \subseteq V$ such that $|V'| \geq K$ and no two vertices in V' are joined by an edge in E?
- *Feedback Vertex Set.* Given a directed graph $G(V, A)$ and an integer $K \leq |V|$, is there a subset $V' \subseteq V$ such that $|V'| \leq K$ and every directed cycle in G has at least one vertex in V'?
- *Graph Colorability.* Given a graph $G(V, E)$ and an integer $K \leq |V|$, is there a "coloring" function $f: V \rightarrow 1, 2, \ldots, K$ such that for every edge $(u, v) \in E$, $f(u) \neq f(v)$?
- *Graph Bandwidth.* Given a graph $G(V, E)$ and an integer $K \leq |V|$, is there a one-to-one function $f: V \rightarrow 1, 2, \ldots, |V|$ such that for every edge $(u, v) \in E$, $|f(u) - f(v)| \leq K$?
- *Graph Isomorphism.* Given two graphs $G(V_1, E_1)$ and $G(V_2, E_2)$, is there a one-to-one function $f: V_1 \rightarrow V_2$ such that $(u, v) \in E_1$ if and only if $(f(u), f(v)) \in E_2$?
- *Induced Bipartite Subgraph.* Given a graph $G(V, E)$ and an integer $K \leq |V|$, is there a subset $V' \subseteq V$ such that $|V'| \geq K$ and the subgraph induced by V' is bipartite?
- *Planar Subgraph.* Given a graph $G(V, E)$ and an integer $K \leq |E|$, is there a subset $E' \subseteq E$ such that $|E'| \geq K$ and the subgraph $G'(V, E')$ is planar?
- *Steiner Tree.* Given a weighted graph $G(V, E)$, a subset $V' \subseteq V$, and a positive integer bound B, is there a subgraph of G that is a tree, comprises at least all vertices in V', and has a total sum of weights no more than B?
- *Graph Partitioning.* Given a graph $G(V, E)$ and two positive integers K and J, is there a partition of V into disjoint subsets $V_1, V_2, \ldots, V_m$ such that each subset contains no more than K vertices and the total number of edges that are incident on vertices in two different subsets is no more than J?
- *Traveling Salesman.* Given a set of cities, a distance between every pair of cities, and a bound K, is there a tour that visits each city exactly once and has total distance no more than K?
- *Bin Packing.* Given a finite set S of positive integers, and two positive integers B and K, is there a partition of S into K disjoint subsets $S_1, S_2, \ldots, S_K$ such that the sum of the elements in each S_i is no more than B?
- *Subset Sum.* Given a finite set S of positive integers and a positive integer B, is there a subset $S' \subseteq S$ whose sum of elements is exactly B?
- *Knapsack.* Given a finite set S of items, each with an integer size and an integer value, and two integer bounds B and P, is there a subset $S' \subseteq S$ whose total sum of sizes is at most B, and the total sum of values is at least P?
- *One-Processor Scheduling with Release Times and Deadlines.* Given a set T of tasks, each task $t \in T$ having a duration $l(t) \in Z^+$, a release time $r(t) \in Z_0^+$, and a deadline $d(t) \in Z+$, is there a function $q: T \rightarrow Z_0^+$ such that for all $t \in T$, $q(t) \geq r(t)$, $q(t) + l(t) \leq d(t)$, and for any pair t, t' with $q(t) > q(t')$, $q(t) \geq q(t') + l(t')$?

- *Multiprocessor Scheduling.* Given a set T of tasks, each task $t \in T$ having a duration $l(t) \in Z^+$, and two positive integers m and D, is there a function $q : T \to Z_0^+$ such that the number of tasks that have overlapping intervals $[q(t), q(t) + l(t)]$ is no more than m and $\max_{t \in T} \{(q(t) + l(t))\} \leq D$?
- *Integer Linear Programming.* Given K integer vectors x_i and K integers b_i, $1 \leq i \leq K$, as well as an integer vector c and an integer B, is there an integer vector y such that $x_i \cdot y \leq b_i$, $1 \leq i \leq K$, and $c \cdot y \geq B$?

NP-Hard Problems

A generalization of the *NP*-complete class is the *NP*-hard class. The *NP*-hard class is extended to comprise optimization problems, as well as decision problems that do not seem to belong to *NP*. All that is required for a problem to be *NP*-hard is that some *NP*-complete problem reduce polynomially to it. For example, the optimization version of the Traveling Salesman problem is an *NP*-hard problem, since if it were polynomially solvable, the decision version of the problem (which is *NP*-complete) would be trivially solved polynomially. An example of a decision problem that is *NP*-hard but not known to be *NP*-complete is the *K*th Heaviest Subset problem: Given a finite set S of integers and two integers K and B, are there K distinct subsets $S_1, S_2, \ldots, S_K \subseteq S$, each of which has a sum of elements at least B? This problem has been shown to be *NP*-hard by a reduction from the Partition problem (6), but it is not known to be in *NP* since the obvious candidate solution to be verified (i.e., the list of the K subsets) does not have polynomial size with respect to the original instance (as K can be as large as $2^{|S|}$). Another problem that is *NP*-hard but not known to belong to *NP* is the Minimum Equivalent Expression problem: Given a well-formed boolean expression E involving a set of variables, the constants "true" and "false," and the logical connectives "and," "or," "not," and "implies," and a positive integer K, is there another expression E' that is equivalent to E and contains no more than K literals? Minimum Equivalent Expression is *NP*-hard since the Satisfiability problem reduces polynomially to it. But it is not known to be in *NP* because, although a candidate solution E' can be described polynomially with respect to the original instance size, the obvious verification of that solution involves the use of an algorithm for the Satisfiability of Boolean Expressions problem, which is itself *NP*-complete (as a generalization of the Satisfiability problem).

If the decision version of an optimization problem is *NP*-complete, then the optimization problem is *NP*-hard, since the yes or no answer sought in the decision version can readily be given in polynomial time once the optimum solution in the optimization version has been obtained. But it is also the case that for most *NP*-hard optimization problems, a reverse relation holds: That is, these *NP*-hard optimization problems can reduce polynomially to their *NP*-complete decision versions. The strategy is to use a binary search procedure that establishes the optimal value after a logarithmically bounded number of calls to the decision version subroutine. Such *NP*-hard problems are sometimes referred to as *NP-equivalent.* The latter fact is another motivation for the study of *NP*-complete problems: a polynomial-time algorithm for any *NP*-complete (decision) problem would provide actually a polynomial-time algorithm for all such *NP*-equivalent optimization problems.

Algorithms for *NP*-Hard Problems

Once a new problem for which an algorithm is sought is proved to be *NP*-complete or *NP*-hard, the search for a polynomial-time algorithm is abandoned (unless one seeks to prove that $NP = P$), and the following basic four approaches remain to be followed:

1. Try to improve as much as possible over the straightforward exhaustive (exponential) search by using techniques like branch-and-bound, dynamic programming, cutting plane methods, or Lagrangian techniques.
2. For optimization problems, try to obtain a polynomial-time algorithm that finds a solution that is probably close to the optimal. Such an algorithm is known as *approximation algorithm* and is generally the next best thing one can hope for to solve the problem.
3. For problems that involve numerical bounds, try to obtain an algorithm that is polynomial in terms of the instance size *and* the size of the maximum number occuring in the encoding of the instance. Such an algorithm is known as *pseudopolynomial-time algorithm* and becomes practical if the numbers involved in a particular instance are not too large. An *NP*-complete problem for which a pseudopolynomial-time algorithm exists is referred to as *weakly* NP-*complete* (as opposed to *strongly* NP-*complete*).
4. Use a polynomial-time algorithm to find a "good" solution based on rules of thumb and insight. Such an algorithm is known as a heuristic. No proof is provided about how good the solution is, but well-justified arguments and empirical studies justify the use of these algorithms in practice.

In addition, before any of these approaches is examined, one should check whether the problem of concern is actually a special case of an *NP*-complete or an *NP*-hard problem, since many special cases can be solved polynomially or pseudopolynomially. Examples include polynomial algorithms for finding a longest path in a directed acyclic graph, finding a maximum independent set in a transitive graph, finding a schedule in the One-Processor Scheduling with Release Times and Deadlines problem when all tasks have unit length; pseudopolynomial-time algorithms for solving the Bin Packing problem when the number K of bins is fixed, finding a schedule in the One-Processor Scheduling with Release Times and Deadlines problem when the release times and deadlines are bounded by a constant, and so on.

In the next section we give some more information about approximation and pseudopolynomial-time algorithms.

POLYNOMIAL-TIME ALGORITHMS

Graph Representations and Traversals

There are two basic schemes for representing graphs in a computer program. Without loss of generality, we assume that the graph is directed [represented by $G(V, A)$]. Undirected graphs can always be considered as bi-directed. In the first scheme, known as the *adjacency matrix* representation, a $|V| \times |V|$ matrix M is used, where every row and column of the matrix corresponds to a vertex of the graph, and entry

$M(a, b)$ is 1 if and only if $(a, b) \in A$. This simple representation requires $O(|V|^2)$ time and space.

In the second scheme, known as the *adjacency list* representation, an array $L[1 \ldots |V|]$ of linked lists is used. The linked list starting at entry $L[i]$ contains the set of all vertices that are the heads of all edges with tail vertex i. The time and space complexity of this scheme is $(|V| + |E|)$.

Both schemes are widely used as part of polynomial-time algorithms for working with graphs [e.g., see (9)]. The adjacency list representation is more economical to construct, but locating an edge using the adjacency matrix representation is very fast (takes $O(1)$ time compared to the $O(|V|)$ time required using the adjacency list representation). The choice between the two depends on the way the algorithm needs to access the information on the graph.

A basic operation on graphs is the *graph traversal,* where the goal is to visit systematically all the vertices of the graph. There are three graph traversal methods: *Depth-first search* (*DFS*), *breadth-first search* (*BFS*), and *topological search*. The last applies only to directed acyclic graphs. Assume that all vertices are marked initially as unvisited, and that the graph is represented using an adjacency list L.

Depth-first search traverses a graph following the deepest (forward) direction possible. The algorithm starts by selecting the lowest numbered vertex v and marking it as visited. DFS selects an edge (v, u), where u is still unvisited, marks u as visited, and starts a new search from vertex u. After completing the search along all paths starting at u, DFS returns to v. The process is continued until all vertices reachable from v have been marked as visited. If there are still unvisited vertices, the next unvisited vertex w is selected and the same process is repeated until all vertices of the graph are visited.

The following is a recursive implementation of subroutine of DFS(v) that determines all the vertices reachable from a selected vertex v. $L[v]$ represents the list of all vertices that are the heads of edges with tail v, and array $M[u]$ contains the visited or unvisited status of every vertex u.

```
Procedure DFS(v)
M[v] := visited;
FOR each vertex u ∈ L[v] DO
    IF M[u] = unvisited THEN Call DFS(u);
END DFS
```

The time complexity of DFS(v) is $O(|V_v| + |E_v|)$ where $|V_v|$, $|E_v|$ are all the number of vertices and edges that have been visited by DFS(v). The total time for traversing the graph using DFS is $O(|E| + |V|) = O(|E|)$.

Breadth-first search visits all vertices at distance k from the lowest numbered vertex v before visiting any vertices at distance $k + 1$. Breadth-first search constructs a breadth-first search tree, initially containing only the lowest numbered vertex. Whenever an unvisited vertex w is visited in the course of scanning the adjacency list of an already visited vertex u, vertex w and edge (u, w) are added to the tree. The traversal terminates when all vertices have been visited. The approach can be implemented using queues so that it terminates in $O(|E|)$ time.

The final graph traversal method is the topological search which applies only to directed acyclic graphs. In directed acyclic graphs there are vertices that have no incoming edges and vertices that have no outgoing edges. Topological search visits a vertex only if it has no incoming edges or all its incoming edges have been explored. The approach can also be implemented to run in $O(|E|)$ time.

Design Techniques for Polynomial-Time Algorithms

There are three frameworks that can be used to obtain polynomial-time algorithms for combinatorial optimization (or decision) problems: (1) greedy algorithms, (2) divide-and-conquer algorithms, and (3) dynamic programming algorithms.

Greedy Algorithms. These are algorithms that use a greedy (straightforward) approach to solve a combinatorial optimization problem. Consider the following problem, known as the *Program Storage problem.* The instance consists of a set of n programs that are to be stored on a tape of length L. Every program has an integer length l_i, $1 \le i \le n$. When a program is to be retrieved, the tape is positioned at the start. Thus, if the order of the programs in the tape is $p_1, p_2, \ldots, p_n$, the time to retrieve program $p_m = \sum_{k=1}^{m} l_{p_k}$. The goal is to find the best possible way of storing the programs on the tape so that the total program retrieval time $\sum_{j=1}^{n} \sum_{k=1}^{m} l_{p_k}$ is minimized.

The greedy algorithm for this problem is to store the programs on the tape in increasing order of their lengths. The time complexity of this algorithm is $O(n \log n)$ and is determined by the procedure that sorts the programs according to their lengths. We refer the reader to Ref. (9) for sorting algorithms.

This greedy algorithm is very simple, and we avoid a more formal description. This is normally the case for all greedy algorithms. Despite the simplicity in their description, the proof of the optimality of a greedy solution is not a trivial task. In general, the proof is based on a systematic sequence of contradiction arguments. The following theorem proves that the greedy algorithm for the program storage problem is optimal (10). Similar methodology to the one in the proof of the theorem can be used to show the optimality of greedy algorithms.

Theorem 2 If $l_1 \le l_2 \le \ldots \le l_n$, the ordering $i_j = j$, $1 \le j \le n$ minimizes $\sum_{k=1}^{n} \sum_{j=1}^{k} l_{i_j}$ over all ordering permutations.

Proof. Let $I = i_1, i_2, \ldots i_n$ be the optimal permutation of the index set $\{1, 2, \ldots, n\}$. Then the retrieval time $R(I) = \sum_{k=1}^{n} \sum_{j=1}^{k} l_{i_j}$ is equal to $\sum_{k=1}^{n} (n - k + 1) l_{i_k}$.

Assume that the opposite holds and that in the optimal ordering there exist programs a, b such that $a < b$ and $l_{i_a} > l_{i_b}$ (here a and b denote the relative positions of the programs in the permutation). In particular, let a and b be the two leftmost programs for which the above condition holds. If this is the case, interchanging the order of i_a and i_b results in a permutation I' for which it is shown that the total program retrieval time $R(I')$ is less than $R(I)$. This is enough to show that this greedy algorithm is optimal, and for any two programs a and b (from left to right in the current program permutation) for which $l_{i_a} > l_{i_b}$, the same argument can be applied to reduce the retrieval cost $R(\ )$. That is, the cost of the greedy permutation is no more than the cost of the optimal permutation I, and thus it is optimal.

The fact that the total program retrieval time $R(I')$ is less than $R(I)$ is shown as follows: Observe that $R(I') = \sum_{k, \neq a, b} ((n - k + 1) l_{i_k}) + (n - a + 1) l_{i_b} + (n - b + 1) l_{i_a}$. Then $R(I') - R(I) = (n - a + 1)(l_{i_a} - l_{i_b}) + (n - b + 1)(l_{i_b} - l_{i_a}) = (b - a)(l_{i_a} - l_{i_b}) > 0$.

Divide-and-Conquer Algorithms. This methodology is based on a systematic partition of the input instance in a top-down manner into smaller instances until small enough instances are obtained for which the problem degenerates to trivial computations. The overall optimal solution, which is the optimal solution on the input instance, is then calculated by appropriately working on the already calculated results on the subinstances. The recursive nature of the methodology necessitates the solution of one or more recurrence relations for determining the execution time.

As an example, we show how divide-and-conquer can be applied to find the maximum and the minimum integer in an array $A[1\ .\ .\ n]$. More explicitly, the goal is to assign to variables *max* and *min* the largest and smallest integers in the array. This problem is central in many areas of computer science, computer engineering, and operations research, among others.

The idea is to recursively find the maximum and the minimum element in subarrays $A[i\ .\ .\ j]$ that have at least two elements by first locating the midpoint m, and recursively solve the problem in the two smaller arrays $A[i\ .\ .\ m]$ and $A[m+1\ .\ .\ j]$. Once the maximum and minimum on the latter two arrays are computed, we take the maximum of the two calculated maxima and the minimum of the two calculated minima and we assign them as the maximum and the minimum of $A[i\ .\ .\ j]$, respectively.

```
Procedure MaxMin(i, j)
IF i = j THEN rmax = A[i]; rmin = A[i];
ELSE IF i = j − 1 THEN IF A[i] < A[j] THEN
  rmax = A[j]; rmin = A[i];
  ELSE rmax = A[i]; rmin = A[j];
ELSE
    m = ⌊(i + j)/2⌋;
    (m1, m2) := MaxMin(i, m); (m3, m4) :=
      MaxMin(m + 1, j);
    rmax = max(m1,m3); rmin = min(m2,m4);
Return(rmax, rmin);
END MaxMin
```

We analyze the time complexity of the approach by counting element comparisons. Let $T(n)$ be this number. The recursive nature of the MaxMin procedure allows us to express $T(n)$ by the following recurrence relation:

$$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + T\left(\left\lceil \frac{n}{2} \right\rceil\right) + 2 \text{ when } n > 2$$

$$T(n) = 1 \text{ when } n = 2, \quad T(n) = 0 \text{ when } n = 1$$

When n is a power of 2, exactly $3n/2 - 2$ number of comparisons are needed in the average, worst, and best case. In general, the worst case on the number of comparisons is $O(n)$.

Dynamic Programming Algorithms. In dynamic programming the optimal solution is calculated by starting from the simplest subinstances and combining the solutions of the smaller subinstances to solve larger subinstances, in a bottom-up manner. In order to guarantee a polynomial-time algorithm, the total number of subinstances that have to be solved must be polynomially bounded. Once a subinstance has been solved, any larger subinstance that needs that subinstance's solution, does not recompute it, but rather looks it up in table where it has been stored. Dynamic programming is applicable only to problems which obey the *principle of optimality.* This principle holds whenever in an optimal sequence of choices, each subsequence is also optimal. The difficulty in this approach is to come up with a decomposition of the problem into a sequence of subproblems for which the principle of optimality holds and can be applied in polynomial time.

We illustrate the approach by finding the maximum independent set in a cycle graph in $O(n^2)$ time, where n is the number of chordal endpoints in the cycle [see (4)]. Note that the maximum independent set is *NP*-hard on general graphs.

Let $G(V, E)$ be a cycle graph, and that $v_{ab} \in V$ corresponds to a chord in the cycle. We assume that no two of the n chords share an end point, and that the end points are labeled from 0 to $2n - 1$ clockwise around the cycle. Let G_{ij} be the subgraph induced by the set of vertices $v_{ab} \in V$ such that $i \le a$, $b \le j$.

Let $M(i, j)$ denote a maximum independent set of $G_{i,j}$. $M(i, j)$ is computed for every pair of chords, but $M(i, a)$ must be computed before $M(i, b)$ if $a < b$. Observe that if $i \ge j$, $M(i, j) = 0$ because $G_{i,j}$ has no chords. In general, in order to compute $M(i, j)$, the end point k of the chord with end point j must be found. If k is not in the range $[i, j - 1]$, then $M(i, j) = M(i, j - 1)$ because graph $G_{i,j}$ is identical to graph $G_{i,j-1}$. Otherwise, we consider two cases: First, if $v_{kj} \in M(i, j)$, then $M(i, j)$ does not have any vertex v_{ab} where $a \in [i, k - 1]$ and $b \in [k + 1, j]$. In this case $M(i, j) = M(i, k - 1) \cup M(K = 1, j - 1) \cup v_{kj}$; second, if $v_{kj} \notin M(i, j)$, then $M(i, j) = M(i, j - 1)$. Either of these two cases may apply, but the largest of the two maximum independent sets will be allocated to $M(i, j)$. The flowchart of the algorithm is given below:

```
Procedure MIS(V)
FOR j = 0 TO 2N − 1 DO
    Let (j, k) be the chord whose one end point
      is j;
    FOR i = 0 TO j − 1 DO
       IF i ≤ k ≤ j − 1 AND |M(i, k − 1)| + 1
         + |M(k + 1, j − 1)| > |M(i, j − 1)| THEN
         M(i, j) = M(i, k − 1) ∪ v_kj ∪ M(k + 1,
           j − 1);
       ELSE M(i, j) = M(i, j − 1);
END MIS
```

Three Basic Graph Problems

In this section we define formally and present polynomial time algorithms for three problems that are widely used in operations research, networking, and VLSI CAD, in the sense that many problems are reduced to solving these basic graph problems. They are the *shortest path problem,* the *flow problem,* and the *matching problem.*

Shortest Paths. The instance consists of a graph $G(V, E)$ with lengths $l(u, v)$ on its edges (u, v), a given source $s \in V$ and a target $t \in V$. We assume without loss of generality that the graph is directed. The goal is to find a shortest length path from s to t. The weights can be positive or negative numbers but there is no cycle for which the sum of the weights on its edges is negative. (If negative length cycles are allowed the problem is *NP*-hard.) Variations of the problem include

the all-pair of vertices shortest paths, and the m shortest path calculation in a graph.

We present here a dynamic programming algorithm for the shortest path problem which is known as the Bellman-Ford algorithm. The algorithm has $O(n^3)$ time complexity, but faster algorithms exist when all the weights are positive [e.g., the Dijkstra algorithm with complexity $O(n \cdot \min\{\log |E|, |V|)\}$] or when the graph is acyclic (based on topological search and with linear time complexity). All existing algorithms for the shortest path problem are based on dynamic programming. The Bellman–Ford algorithm works as follows:

Let $l(i, k)$ be the length of edge (i, j) if directed edge (i, j) exists and ∞ otherwise. Let $s(j)$ denote the length of the shortest path from the source s to vertex j. Assume that the source has label 1 and that the target has label $n = |V|$. We have that $s(1) = 0$. We also know that in a shortest path to any vertex j there must exist a vertex k, $k \neq j$, such that $s(j) = s(k) + l(k, j)$. Therefore

$$s(j) = \min_{k \neq j}\{s(k) + l(k, j)\}, \qquad j \geq 2$$

Bellman–Ford's algorithm, which eventually computes all $s(j)$, $1 \leq j \leq n$, calculates optimally the quantity $s(j)^{m+1}$ defined as the length of the shortest path to vertex j subject to the condition that the path does not contain more than $m + 1$ edges, $0 \leq m \leq |V| - 2$. In order to be able to calculate quantity $s(j)^{m+1}$ for some value $m + 1$, the $s(j)^m$ values for all vertices j have to be calculated.

Given the initialization $s(1)^1 = 0$, $s(j)^1 = l(1, j)$, $j \neq 1$, the computation of $s(j)^{m+1}$ for any values of j and m can be recursively computed using the formula

$$s(j)^{m+1} = \min\{s(j)^m\}, \quad \min\{s(k)^m + l(k, j)\}$$

The computation terminates when $m = |V| - 1$, because no shortest path has more than $|V| - 1$ edges.

Flows. All flow problem formulations consider a directed or undirected graph $G + (V, E)$, a designated source s, a designated target t, and a nonnegative integer capacity $c(i, j)$ on every edge (i, j). Such a graph is sometimes referred to as a *network*. We assume that the graph is directed. A *flow* from s to t is an assignment F of numbers $f(i, j)$ on the edges, called the amount of flow through edge (i, j), subject to the following conditions:

$$0 \leq f(i, j) \leq c(i, j) \tag{1}$$

Besides s and t, any vertex i must satisfy the *conservation of flow*. That is,

$$\sum_j f(j, i) - \sum_j f(i, j) = 0 \tag{2}$$

Let $v = \sum_j f(s, j)$. Then clearly $\sum_j f(s, j) = v = \sum_j f(j, t)$. v is called the value of the flow. From Eq. (2) we have

$$\begin{aligned} \sum_j f(j, i) - \sum_j f(i, j) &= -v \qquad \text{if } i = 2 \\ \sum_j f(i, j) &= 0 \qquad \text{if } i \neq s, t \\ \sum_j f(i, j) &= v \qquad \text{if } i = t \end{aligned} \tag{3}$$

A flow F that satisfies Eqs. (1) and (3) is called *feasible. In the max flow problem* the goal is to find a feasible flow F for which v is maximized. Such a flow is called a *maximum flow.* There is a problem variation, called the *minimum flow problem,* where condition (1) is substituted by $f(i, j) \geq c(i, j)$ and the goal is to find a flow F for which v is minimized. The minimum flow problem can be solved by modifying algorithms that compute the maximum flow in a graph.

Finally, another flow problem formulation is the *minimum cost flow problem.* Here each edge has, in addition to its capacity $c(i, j)$, a *cost* $p(i, j)$. If $f(i, j)$ is the flow through the edge, then the cost of the flow through the edge is $p(i, j) \cdot f(i, j)$ and the overall cost C for a flow F of a value v is $\sum_{i,j} p(i, j) \cdot f(i, j)$. The problem is to find a *minimum cost flow* F for a given value v.

Many problems in operations research, networking, scheduling, and VLSI CAD can be modeled or reduced to one of these three flow problem variations. All three problems can be solved in polynomial time using as subroutines shortest path calculations. Here we describe an $O(|V^3|)$ algorithm for the maximum flow problem. However, faster algorithms exist in the literature. We first give some definitions and theorems.

Let P be an *undirected* path from s to t, i.e., the direction of the edges is ignored. An edge $(i, j) \in P$ is said to be a *forward* edge if it is directed from s to t and *backward* otherwise. P is said to be an *augmenting path* with respect to a given flow F if $f(i, j) < c(i, j)$ for each forward edge, and $f(i, j) > 0$ for each backward edge in P.

Observe that if the flow in each forward edge of the augmenting path is increased by one unit and the flow in each backward edge is decreased by one unit, the flow is feasible and its value has been increased by one unit. We will show that a flow has maximum value if and only if there is no augmenting path in the graph. Then the maximum flow algorithm is simply a series of calls to a subroutine that finds an augmenting path and increments the value of the flow as described earlier.

Let $S \subset V$ be a subset of the vertices. The pair (S, T) is called a *cutset* if $T = V - S$. If $s \in S$ and $t \in T$, the (S, T) is called an (s, t) *cutset.* The *capacity of the cutset* (S, T) is defined as $c(S, T) = \sum_{i \in S} \sum_{j \in T} c(i, j)$, which is the sum of the capacities of all edges from S to T. We note that many problems in networking, operations research, scheduling, and VLSI CAD (physical design, synthesis, and testing) are formulated as minimum capacity (s, t) cutset problems. We show below that the minimum capacity (s, t) problem can be solved with a maximum flow algorithm.

Lemma 3.1 The value of any (s, t) flow cannot exceed the capacity of any (s, t) cutset.

Proof. Let F be an (s, t) flow with value v. Let (S, T) be an (s, t) cutset. From Eq. (3) the value of the flow v is also $v = \sum_{i \in S}(\sum_j f(i, j) - \sum_j f(j, i)) = \sum_{i \in S} \sum_{j \in S}(f(i, j) - f(j, i)) + \sum_{i \in S} \sum_{j \in T}(f(i, j) - f(j, i)) = \sum_{i \in S} \sum_{j \in T}(f(i, j) - f(j, i))$, since $\sum_{i \in S} \sum_{j \in S}(f(i, j) - f(j, i))$ is 0. But $f(i, j) \leq c(i, j)$ and $f(j, i) \geq 0$. Therefore $v \leq \sum_{i \in S} \sum_{j \in S} c(i, j) = c(S, T)$.

Theorem 3 A flow F has maximum value v if and only if there is no augmenting path from s to t.

Proof. If there is an augmenting path then we can modify the flow to get a larger value flow. This contradicts the assumption that the original flow has a maximum value.

Suppose, on the other hand, that F is a flow such that there is no augmenting path from s to t. We want to show that F has the maximum flow value. Let S be the set of all the vertices j (including s) for which there is an augmenting path from s to j. By the assumption that there is an augmenting path from s to t, we must have that $t \notin S$. Let $T = V - S$ (recall that $t \in T$). From the definition of S and T, it follows that $f(i, j) = c(i, j)$ and $f(j, i) = 0$, $\forall i \in S, j \in T$.

Now $v = \sum_{i \in S}(\sum_j f(i, j) - \sum_j f(j, i)) = \sum_{i \in S} \sum_{j \in S}(f(i, j) - f(j, i)) + \sum_{i \in S} \sum_{j \in T}(f(i, j) - f(j, i)) = \sum_{i \in S} \sum_{j \in S} c(i, j)$, since $c(i, j) = f(i, j)$ and $f(j, i) = 0$, $\forall i, j$. By Lemma 3.1 the flow has the maximum value.

Next we state two theorems whose proof is rather straightforward.

Theorem 4 If all the capacities are integers, then there exists a maximum flow F, where all $f(i, j)$ are integers.

Theorem 5 The maximum value of an (s, t) flow is equal to the minimum capacity of an (s, t) cutset.

Finding an augmenting path in a graph can be done by a systematic graph traversal in linear time. Thus a straightforward implementation of the maximum flow algorithm repeatedly finds an augmenting path and increments the amount of the (s, t) flow. This is a pseudopolynomial-time algorithm (see the next section), whose worst-case time complexity is $O(v \cdot |E|)$. In many cases such an algorithm may turn out to be very efficient. For example, when all capacities are uniform, then the overall complexity becomes $O(|E|^2)$.

In general, the approach needs to be modified using the Edmonds–Karp modification (1) so that each flow augmentation is made along an augmenting path with a minimum number of edges. With this modification, it has been proven that a maximum flow is obtained after no more than $|E| \cdot |V|/2$ augmentations, and the approach becomes fully polynomial. Faster algorithms for maximum flow computation rely on capacity scaling techniques and are described in (9,3), among others.

Matchings. Matching problems are defined on undirected graphs $G(V, E)$. A *matching* in a graph is a set $M \subset E$ such that no two edges in M are incident to the same vertex.

The *maximum cardinality matching* problem is the most common version of the matching problem. Here the goal is to obtain a matching so that the size (cardinality) of M is maximized.

In the *maximum weighted matching* version, each edge $(i, j) \in V$ has a nonnegative integer weight, and the goal is to find a matching M so that $\sum_{e \in M} w(e)$ is maximized.

In the *min-max matching problem* the goal is to find a maximum cardinality matching M where the minimum weight on an edge in M is maximized. The *max-min matching problem* is defined in an analogous manner.

All the above matching variations are solvable in polynomial time and find important applications. For example, a variation of the min-cut graph partitioning problem that is central in physical design automation for VLSI asks for partitioning the vertices of a graph into sets of size at most two so that the sum of the weights on all edges with end points in different sets is minimized. It is easy to see that this partitioning problem reduces to the maximum weighted matching problem.

Matching problems often occur on bipartite $G(V_1 \cup V_2, E)$ graphs. The maximum cardinality matching problem amounts to the *maximum assignment* of elements in V_1 ("workers") on to the elements of V_2 ("tasks") so that no worker in V_1 is assigned more than one task. This finds various applications in operations research.

The maximum cardinality matching problem on a bipartite graph $G(V_1 \cup V_2, E)$ can be solved by a maximum flow formulation. Simply, each vertex $v \in V_1$ is connected to a new vertex s by an edge (s, v) and each vertex $u \in V_2$ to a new vertex t by an edge (u, t). In the resulting graph, every edge is assigned unit capacity. The maximum flow value v corresponds to the cardinality of the maximum matching in the original bipartite graph G.

Although the matching problem variations on bipartite graphs are amenable to easily described polynomial-time algorithms, such as the one given above, the existing polynomial-time algorithms for matchings on general graphs are more complex [see (1)].

Approximation and Pseudopolynomial Algorithms

Approximation and pseudopolynomial-time algorithms concern mainly the solution of problems that are proved to be *NP*-hard (although they can sometimes be used on problems that are solvable in polynomial time) but for which the corresponding polynomial-time algorithm involves large constants.

An α-approximation algorithm A for an optimization problem R is a polynomial-time algorithm such that for any instance I of R, $|S_A(I) - S_{\text{OPT}}(I)|/S_{\text{OPT}}(I) \leq \alpha + c$, where $S_{\text{OPT}}(I)$ is the cost of the optimal solution for instance I, $S_A(I)$ is the cost of the solution found by algorithm A, and c is a constant.

Two examples of approximation algorithms are given below: For the optimization version of the Bin Packing problem, an approximation algorithm is the following: Sort the items into decreasing order of sizes. Assign each item in this order into the first bin that has room for it. If no such bin exists, introduce a new bin. This simple algorithm has been shown (6) to always give a solution that is no more than 11/9 times the optimal plus 4, namely $|S_A(I) - S_{\text{OPT}}(I)|/S_{\text{OPT}}(I) \leq 2/9 + c$.

The second example concerns a special but practical version of the Traveling Salesman problem that obeys the triangular inequality for all city distances. Given a weighted graph $G(V, E)$ of the cities, the algorithm first finds a minimum spanning tree T of G (i.e., a spanning tree that has minimum sum of edge weights). Then it finds a minimum weight matching M among all vertices that have odd degree in T. Lastly, it forms the subgraph $G'(V, E')$, where E' is the set of all edges in T and M and finds a path that starts from and terminates to the same vertex and passes through every edge exactly once (such a path is known as Eulerian tour). Every step in this algorithm takes polynomial time. It has been shown that $|S_A(I) - S_{\text{OPT}}(I)|/S_{\text{OPT}}(I) \leq 1/2$.

Unfortunately, obtaining a polynomial-time approximation algorithm for an *NP*-hard optimization problem can be very difficult. In fact it has been shown for some cases that this may be impossible. For example, it has been shown that un-

less $NP = P$ there is no α-approximation algorithm for the general Traveling Salesman problem for any $\alpha > 0$.

A pseudopolynomial-time algorithm for a problem R is an algorithm with time complexity $O(p(n, m))$, where $p(\ )$ is a polynomial of two variables, n is the size of the instance, and m is the magnitude of the largest number occuring in the instance. Only problems involving numbers that are not bounded by a polynomial on the size of the instance are applicable for solution by a pseudopolynomial-time algorithm. In principle, a pseudopolynomial-time algorithm is exponential given that the magnitude of a number is exponential to the size of its logarithmic encoding in the problem instance, but in practice, such an algorithm may be useful in cases where the numbers involved are not large.

NP-complete problems for which a pseudopolynomial-time algorithm exists are referred to as *weakly NP-complete,* whereas *NP*-complete problems for which no pseudopolynomial-time algorithm exists (unless $NP = P$) are referred to as *strongly NP-complete.* Examples of weakly *NP*-complete problems are the Knapsack and Subset Sum problems. A pseudopolynomial-time algorithm for the latter is the following: Given a set of positive integers $S = \{s_1, s_2, \ldots, s_n\}$ and an integer B, let $T[i, j]$, $1 \le i \le n$, $1 \le j \le B$, be 1 whenever it is true that there is a subset of the first i integers $s_1, s_2, \ldots, s_i$ with sum exactly B (otherwise, $T[i, j] = 0$). The entries of matrix T are systematically assigned as: $T[1, j] = 1$ if $j = 0$ or $j = s_1$; and, for $2 \le i \le n$, $1 \le j \le B$, $T[i, j] = 1$ if $T[i - 1, j] = 1$, or $s_i \le j$ and $T[i - 1, j - s_i] = 1$. This algorithm (a case of dynamic programming) takes time $O(nB)$, and the answer to the Subset Sum problem is given by the value of $T[n, B]$.

Examples of problems that are strongly *NP*-complete are the Traveling Salesman and the Bin Packing problems.

Probabilistic Algorithms

Probabilistic algorithms are a class of algorithms that do not depend exclusively on their input to carry out the computation. Instead, at one or more points in the course of the algorithm where a choice has to be made, they use a pseudorandom number generator to select "randomly" one out of a finite set of alternatives for arriving at a solution. Probabilistic algorithms are fully programmable (i.e., they are not like the nondeterministic algorithms), but in contrast with the deterministic algorithms, they may give different results each time for the same input instance (assuming that the initial state of the pseudorandom number generator is different each time).

Probabilistic algorithms trade off the certainty of a correct solution with a reduction in the computation time. As a simple example, in order to find a number in a list of n numbers that is greater than or equal to the median, $n/2$ elements have to be examined. However, assuming that the numbers are equally distributed, by examining only k numbers and keeping the maximum, the probability that the number is greater than or equal to the median is $1 - (\frac{1}{2})^k$, which is very practical if, for example, $k = 20$ while $n = 1{,}000{,}000$. Two major classes of probabilistic algorithms are the Monte Carlo type and the Las Vegas type. In the former, the algorithm guarantees that the solution it gives has a high probability of being correct for *every* instance. In the latter, the algorithm guarantees that any solution it reports is correct with certainty, but occasionally it may terminate by reporting failure. A representative case of probabilistic algorithms is testing whether a given very large integer is prime (this has applications to cryptography).

There is also a class of algorithms that resemble the probabilistic in that they use a pseudorandom number generator, but the solution they give is always correct. Such algorithms make certain choices that are not determined by the input instance in order to obtain the solution faster (the solution itself does not depend on these choices). These algorithms are known as *randomized.* A typical example is the Quicksort algorithm, which takes $O(n^2)$ time in the worst case for sorting a list of n elements, but $O(n \log n)$ time on average if the order of the elements in the list is randomized by the algorithm (the latter average time complexity is not conditioned on any distribution for the elements of the list).

BIBLIOGRAPHY

1. E. L. Lawler, *Combinatorial Optimization: Algorithms and Matroids,* New York: Holt, Rinehart and Winston, 1976.
2. C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity,* Englewood Cliffs, NJ: Prentice-Hall, 1982.
3. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows,* Englewood Cliffs, NJ: Prentice-Hall, 1993.
4. N. A. Sherwani, *Algorithms for VLSI Physical Design Automation,* Norwell, MA: Kluwer, 1993.
5. D. Bertsekas and R. Gallagher, *Data Networks,* Upper Saddle River, NJ: Prentice-Hall, 1992.
6. M. R. Garey and D. S. Johnson, *Computers and Intractability—A Guide to the Theory of NP-Completeness,* San Francisco, CA: W. H. Freeman, 1979.
7. M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design,* Rockville, MD: Computer Science Press, 1990.
8. D. Kagaris and S. Tragoudas, Avoiding linear dependencies in LFSR test pattern generators, *J. Electron. Test. Theor. Appl.,* **6**: 229–241, 1995.
9. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms,* Cambridge, MA: MIT Press, 1990.
10. E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms,* Rockville, MD: Computer Science Press, 1984.

DIMITRIOS KAGARIS
Southern Illinois University

SPYROS TRAGOUDAS
The University of Arizona

GRAPH THEORY. See GEOMETRY.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Green's Function Methods
Standard Article

Jian-Ming Jin[1] and Weng Cho Chew[2]
[1]University of Illinois at Urbana–Champaign, Urbana, IL,
[2]University of Illinois at Urbana–Champaign, Urbana, IL,

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2421
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (238K)

Abstract

The sections in this article are

Scalar Green's Functions

Dyadic Green's Functions

| | |

Copyright © 1999-2008 All Rights Reserved.

GREEN'S FUNCTION METHODS

The Green's function method is a powerful technique for solving boundary-value problems. Green's function was named after George Green (1793–1841), who developed a general method to obtain solutions of Poisson's equation in potential theory. This method was described in an essay by Green entitled "On the application of mathematical analysis to the theories of electricity and magnetism," published in 1828.

To illustrate the Green's function method, consider the electric potential produced by a point electric charge q_1 placed at $\mathbf{r}_1$ in an unbounded homogeneous free space. It is well known from the elementary theory of electricity (1) that this potential at $\mathbf{r}$ is given by

$$\phi_1(\mathbf{r}) = \frac{q_1}{4\pi\epsilon|\mathbf{r} - \mathbf{r}_1|} \tag{1}$$

where $|\mathbf{r} - \mathbf{r}_1|$ denotes the distance between the points $\mathbf{r}$ and $\mathbf{r}_1$ and ϵ is a constant called the permittivity. If there is another point charge q_2 placed at $\mathbf{r}_2$, the potential produced by this charge is

$$\phi_2(\mathbf{r}) = \frac{q_2}{4\pi\epsilon|\mathbf{r} - \mathbf{r}_2|} \tag{2}$$

The total potential produced by q_1 and q_2 is then the linear superposition of ϕ_1 and ϕ_2:

$$\phi(\mathbf{r}) = \phi_1(\mathbf{r}) + \phi_2(\mathbf{r}) = \frac{q_1}{4\pi\epsilon|\mathbf{r} - \mathbf{r}_1|} + \frac{q_2}{4\pi\epsilon|\mathbf{r} - \mathbf{r}_2|} \tag{3}$$

If there are N point charges in the space, the total potential is given by

$$\phi(\mathbf{r}) = \sum_{i=1}^{N} \phi_i(\mathbf{r}) = \sum_{i=1}^{N} \frac{q_i}{4\pi\epsilon|\mathbf{r} - \mathbf{r}_i|} \tag{4}$$

where Σ denotes the summation over all point charges and ϕ_i denotes the potential due to the ith point charge placed at $\mathbf{r}_i$. The procedure described above is known as the principle of linear superposition.

Next, consider the electric potential produced by a volume electric charge whose charge density is denoted by $\rho(\mathbf{r})$. To find the potential, we divide the volume of the charge into many small cubes. The charge within each small cube is then given by

$$q_i \approx \rho(\mathbf{r}_i)\Delta V_i \tag{5}$$

where $\mathbf{r}_i$ denotes the center of the ith cube and ΔV_i denotes the volume of the cube. Since each cube is very small, it can be approximated as a point charge, whose potential is given by

$$\phi_i(\mathbf{r}) \approx \frac{q_i}{4\pi\epsilon|\mathbf{r} - \mathbf{r}_i|} \approx \frac{\rho(\mathbf{r}_i)\Delta V_i}{4\pi\epsilon|\mathbf{r} - \mathbf{r}_i|} \tag{6}$$

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

According to the principle of linear superposition, the total potential is then given by

$$\phi(\mathbf{r}) = \sum_{i=1}^{N} \phi_i(\mathbf{r}) \approx \sum_{i=1}^{N} \frac{\rho(\mathbf{r}_i)\Delta V_i}{4\pi\epsilon|\mathbf{r} - \mathbf{r}_i|} \tag{7}$$

Clearly, the approximation improves as the volume is divided into smaller cubes. In the limit when $\Delta V_i \to 0$, Eq. (7) becomes exact. Hence, one obtains

$$\phi(\mathbf{r}) = \lim_{\Delta V_i \to 0} \sum_{i=1}^{\infty} \frac{\rho(\mathbf{r}_i)\Delta V_i}{4\pi\epsilon|\mathbf{r} - \mathbf{r}_i|} \tag{8}$$

which can be written in the integral form as

$$\phi(\mathbf{r}) = \int_V \frac{\rho(\mathbf{r}')\, dV'}{4\pi\epsilon|\mathbf{r} - \mathbf{r}'|} \tag{9}$$

where V denotes the volume of the electric charge.

The potential produced by a point source of unit strength is called the Green's function. In the example above, the Green's function is

$$G(\mathbf{r}, \mathbf{r}') = \frac{1}{4\pi\epsilon|\mathbf{r} - \mathbf{r}'|} \tag{10}$$

and the total potential can then be written as

$$\phi(\mathbf{r}) = \int_V \rho(\mathbf{r}')G(\mathbf{r}, \mathbf{r}')\, dV' \tag{11}$$

It is clear that the Green's function method treats an arbitrary source for the potential as a linear superposition of weighted point sources. It then finds the potential as the corresponding linear superposition of the potentials produced by the point sources.

Obviously, once the Green's function corresponding to the potential due to a point source is found, the potential produced by an arbitrary distribution of sources can be obtained easily. Therefore, for a specific boundary-value problem, instead of finding the potential for each new source encountered by solving Poisson's equation repeatedly, one can find the Green's function for that problem only once and obtain solutions to any sources by the principle of linear superposition. The procedure of finding the Green's function is usually much simpler than finding the solution to an arbitrary source. To a large extent, a Green's function plays the same role as an impulse response of a linear circuit system. The system response to any input function can be determined by convolving the input function with the impulse response of the system. The Green's function method has since been expanded to deal with a large number of different partial differential equations.

In electrodynamics, both the source (electric current density) and the response (electric or magnetic field) are vectors, each of which has three components. Since each component of an electric current density can produce all three components of the electric or magnetic field, one has nine Green's functions that relate the response to the source. This unwieldiness can be alleviated by introducing the concept of the dyadic Green's function. A dyadic Green's function, which can be expressed as a 3×3 matrix, can be considered as a compact representation of the nine scalar Green's functions. The first use of dyadic Green's function was made by Julian Schwinger. The subject was also covered by Morse and Feshbach (2) in their well-known treatise on the methods of theoretical physics. A more comprehensive treatment of the dyadic Green's functions in electromagnetic theory was presented by Tai (3), who has done much original work on this topic. In his well-known book, Tai derived dyadic Green's functions for a variety of electromagnetic problems of practical importance. Discussions on dyadic Green's functions can also be found in Collin (4), Kong (5), and Chew (6).

As shall be shown later, the Green's function method not only provides a solution to many boundary-value problems involving canonical geometries, but it also leads to integral equations for problems involving more complex geometries. These integral equations form the basis for a numerical solution of complex boundary-value problems.

SCALAR GREEN'S FUNCTIONS

When both the source and response are scalar functions, the corresponding Green's function is also scalar and, hence, the name scalar Green's function.

The Delta Function

Since the Green's function method is based on the representation of an arbitrary source by the superposition of point sources, the mathematical representation of a point source will first be described. Consider an electric charge of unit strength located at point $\mathbf{r}'$. When the volume of the charge approaches zero, the charge density can be described by a function

$$\delta(\mathbf{r} - \mathbf{r}') = \begin{cases} \infty & \text{for } \mathbf{r} = \mathbf{r}' \\ 0 & \text{for } \mathbf{r} \neq \mathbf{r}' \end{cases} \tag{12}$$

Since the total charge remains at unity,

$$\int_V \delta(\mathbf{r} - \mathbf{r}')\, dV = \begin{cases} 1 & \text{for } \mathbf{r}' \text{ in } V \\ 0 & \text{for } \mathbf{r}' \text{ not in } V \end{cases} \tag{13}$$

The function defined in Eqs. (12) and (13) is known as the Dirac delta function, named after P. A. M. Dirac. Clearly, given an arbitrary function $f(\mathbf{r})$, which is continuous at $\mathbf{r} = \mathbf{r}'$,

$$\int_V f(\mathbf{r})\delta(\mathbf{r} - \mathbf{r}')\, dV = \begin{cases} f(\mathbf{r}') & \text{for } \mathbf{r}' \text{ in } V \\ 0 & \text{for } \mathbf{r}' \text{ not in } V \end{cases} \tag{14}$$

This expression represents a volume source $f(\mathbf{r}')$ as a linear superposition of an infinite number of point sources $\delta(\mathbf{r} - \mathbf{r}')$.

In one dimension, the delta function can be considered as the limit of a function,

$$\delta(x - x') = \lim_{\epsilon \to 0} u_\epsilon(x - x') \tag{15}$$

where $u_\epsilon(x - x')$ is called a delta family. It can be a rectangular function of width ϵ and height $1/\epsilon$, or a triangular function

of width 2ϵ and height $1/\epsilon$, or a Gaussian function $e^{-(x-x')^2/2\epsilon^2}/\epsilon\sqrt{2\pi}$, all centered at $x = x'$. The important feature of the delta function is not its shape, but the fact that its effective width approaches zero, while its area remains at unity, that is,

$$\int_a^b \delta(x-x')\,dx = \begin{cases} 1 & \text{for } x' \text{ in } (a,b) \\ 0 & \text{for } x' \text{ not in } (a,b) \end{cases} \tag{16}$$

such that

$$\int_a^b f(x)\delta(x-x')\,dx = \begin{cases} f(x') & \text{for } x' \text{ in } (a,b) \\ 0 & \text{for } x' \text{ not in } (a,b) \end{cases} \tag{17}$$

The delta function so defined is not a function in the classical sense. For this reason, it is called a symbolic or generalized function (7).

Clearly, the delta function is a symmetric function

$$\delta(x-x') = \delta(x'-x) \tag{18}$$

The three-dimensional delta function in the rectangular, cylindrical, and spherical coordinate systems is related to the one-dimensional delta function by

$$\delta(\mathbf{r}-\mathbf{r}') = \delta(x-x')\delta(y-y')\delta(z-z') \tag{19}$$

$$\delta(\mathbf{r}-\mathbf{r}') = \frac{\delta(\rho-\rho')\delta(\phi-\phi')\delta(z-z')}{\rho} \tag{20}$$

$$\delta(\mathbf{r}-\mathbf{r}') = \frac{\delta(r-r')\delta(\theta-\theta')\delta(\phi-\phi')}{r^2\sin\theta} \tag{21}$$

All of the above satisfy Eq. (13).

One-Dimensional Green's Function

To introduce the concept of Green's function in one dimension, consider an infinitely long transmission line with a distributed current source $K(x)$ (3), as illustrated in Fig. 1. Using Kirchhoff's voltage and current laws, one finds the relations between the voltage and current as

$$\frac{dV(x)}{dx} + (j\omega L + R)I(x) = 0 \tag{22}$$

$$\frac{dI(x)}{dx} + (j\omega C + G)V(x) = K(x) \tag{23}$$

where ω denotes the angular frequency and L, C, R, and G are the inductance, capacitance, resistance, and conductance

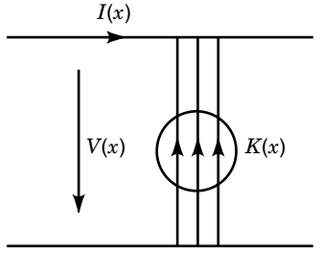


Figure 1. An infinitely long transmission line excited by a distributed current source.

of the transmission line per unit length. Eliminating $I(x)$ in Eqs. (22) and (23), one obtains the differential equation for the voltage as

$$\frac{d^2V(x)}{dx^2} - \gamma^2 V(x) = -(j\omega L + R)K(x) \tag{24}$$

where $\gamma^2 = (j\omega L + R)(j\omega C + G)$. Since the line is infinitely long, there is no reflected wave; hence, $V(x)$ satisfies the boundary conditions

$$\frac{dV(x)}{dx} + \gamma V(x) = 0 \qquad \text{for } x \to \infty \tag{25}$$

$$\frac{dV(x)}{dx} - \gamma V(x) = 0 \qquad \text{for } x \to -\infty \tag{26}$$

Since these boundary conditions are imposed when $|x| \to \infty$, they are also called radiation conditions.

Instead of solving for $V(x)$ directly from Eqs. (24)–(26), one can consider the solution of the following differential equation

$$\frac{d^2g_0(x,x')}{dx^2} - \gamma^2 g_0(x,x') = -\delta(x-x') \tag{27}$$

where $g_0(x)$ satisfies the same radiation conditions as $V(x)$. Since $g_0(x, x')$ is a point source response and $V(x)$ in Eq. (24) is due to the source $(j\omega L + R)K(x)$, according to the principle of linear superposition, $V(x)$ can be expressed as a convolution of $g_0(x, x')$ with $(j\omega L + R)K(x)$:

$$V(x) = \int_{-\infty}^{\infty} (j\omega L + R)K(x')g_0(x,x')\,dx' \tag{28}$$

It is evident that once we obtain $g_0(x, x')$, the voltage on the transmission line can be evaluated via a simple integration using Eq. (28).

To find $g_0(x, x')$, note that since

$$\frac{d^2g_0(x,x')}{dx^2} - \gamma^2 g_0(x,x') = 0 \qquad \text{for } x > x' \text{ or } x < x' \tag{29}$$

one has

$$g_0(x,x') = Ae^{-\gamma x} \qquad \text{for } x > x' \tag{30}$$

$$g_0(x,x') = Be^{\gamma x} \qquad \text{for } x < x' \tag{31}$$

where the radiation conditions in Eqs. (25) and (26) were used to determine the sign in front of γ. To determine the unknown coefficients A and B, consider Eq. (27). First, note that $g_0(x, x')$ must be continuous at $x = x'$, that is,

$$g_0(x,x')|_{x=x'+0} = g_0(x,x')|_{x=x'-0} \tag{32}$$

where $x = x' + 0$ stands for the right-hand side of x' and $x = x' - 0$ stands for the left-hand side of x' since a discontinuity in $g_0(x, x')$ at $x = x'$ would result in a derivative on $\delta(x - x')$ on the left-hand side of Eq. (27). Next, integrate Eq. (27) over the region from $x' - \epsilon$ to $x' + \epsilon$ and in the limit when $\epsilon \to 0$,

$$\left.\frac{dg_0(x,x')}{dx}\right|_{x=x'+0} - \left.\frac{dg_0(x,x')}{dx}\right|_{x=x'-0} = -1 \tag{33}$$

Applying these two conditions to Eqs. (30) and (31), one finds

$$g_0(x,x') = \frac{1}{2\gamma}\, e^{-\gamma(x-x')} \qquad \text{for } x > x' \tag{34}$$

$$g_0(x,x') = \frac{1}{2\gamma}\, e^{\gamma(x-x')} \qquad \text{for } x < x' \tag{35}$$

or, more compactly,

$$g_0(x,x') = \frac{1}{2\gamma}\, e^{-\gamma|x-x'|} \tag{36}$$

This is the Green's function for the infinitely long transmission line.

Two- and Three-Dimensional Green's Functions

Consider the electric and magnetic fields produced by a time harmonic electric source whose current density is denoted by $\mathbf{J}(\mathbf{r})$ and charge density is denoted by $\rho(\mathbf{r})$. These fields satisfy Maxwell's equations given by (1)

$$\nabla \times \mathbf{E}(\mathbf{r}) = -j\omega\mathbf{B}(\mathbf{r}) \tag{37}$$

$$\nabla \times \mathbf{H}(\mathbf{r}) = j\omega\mathbf{D}(\mathbf{r}) + \mathbf{J}(\mathbf{r}) \tag{38}$$

$$\nabla \cdot \mathbf{D}(\mathbf{r}) = \rho(\mathbf{r}) \tag{39}$$

$$\nabla \cdot \mathbf{B}(\mathbf{r}) = 0 \tag{40}$$

and the constitutive relations given by $\mathbf{B} = \mu\mathbf{H}$ and $\mathbf{D} = \epsilon\mathbf{E}$, where μ is the magnetic permeability and ϵ is the electric permittivity. Again, assume that the space is homogeneous. Taking the curl of Eq. (37), one has

$$\nabla \times \nabla \times \mathbf{E}(\mathbf{r}) = -j\omega\mu\nabla \times \mathbf{H}(\mathbf{r}) \tag{41}$$

Using Eq. (38) in Eq. (41), one obtains

$$\nabla \times \nabla \times \mathbf{E}(\mathbf{r}) - k^2\mathbf{E}(\mathbf{r}) = -j\omega\mu\mathbf{J}(\mathbf{r}) \tag{42}$$

where $k^2 = \omega^2\mu\epsilon$. Since $\nabla \times \nabla \times \mathbf{E} = \nabla(\nabla \cdot \mathbf{E}) - \nabla^2\mathbf{E}$, Eq. (42) can be written as

$$\nabla^2\mathbf{E}(\mathbf{r}) + k^2\mathbf{E}(\mathbf{r}) = j\omega\mu\mathbf{J}(\mathbf{r}) + \frac{1}{\epsilon}\nabla\rho(\mathbf{r}) \tag{43}$$

where Eq. (39) has been applied. Similarly, one obtains the equation for $\mathbf{H}$ as

$$\nabla^2\mathbf{H}(\mathbf{r}) + k^2\mathbf{H}(\mathbf{r}) = -\nabla \times \mathbf{J}(\mathbf{r}) \tag{44}$$

Equations (43) and (44) are called inhomogeneous Helmholtz wave equations.

If one uses ϕ to represent each component of $\mathbf{E}$ or $\mathbf{H}$ in a Cartesian coordinate system, then ϕ satisfies the inhomogeneous Helmholtz equation

$$\nabla^2\phi(\mathbf{r}) + k^2\phi(\mathbf{r}) = -f(\mathbf{r}) \tag{45}$$

When $\phi(\mathbf{r})$ propagates in an infinite unbounded space, there is no reflected wave. Hence, $\phi(\mathbf{r})$ satisfies the radiation condition

$$r\left(\frac{\partial\phi}{\partial r} + jk\phi\right) = 0 \qquad \text{for } r \to \infty \tag{46}$$

where r represents the radial variable in spherical coordinates. Instead of solving for $\phi(\mathbf{r})$ directly from Eqs. (45) and (46) for each $f(\mathbf{r})$, one first finds its Green's function, which is the solution of the following partial differential equation

$$\nabla^2 G_0(\mathbf{r},\mathbf{r}') + k^2 G_0(\mathbf{r},\mathbf{r}') = -\delta(\mathbf{r}-\mathbf{r}') \tag{47}$$

subject to the radiation condition in Eq. (46). If G_0 can be found, using the principle of linear superposition, one obtains

$$\phi(\mathbf{r}) = \int_V G_0(\mathbf{r},\mathbf{r}')f(\mathbf{r}')\,dV' \tag{48}$$

where V is the support of $f(\mathbf{r})$, which is the volume having nonzero $f(\mathbf{r})$.

To find G_0, we introduce a new coordinate system with its origin located at $\mathbf{r}'$. Thus, the problem has a spherical symmetry with respect to this point. Equation (47) then becomes

$$\frac{1}{r_1^2}\frac{d}{dr_1}\left[r_1^2\frac{dG_0(\mathbf{r}_1,0)}{dr_1}\right] + k^2 G_0(\mathbf{r}_1,0) = -\delta(\mathbf{r}_1) \tag{49}$$

where $\mathbf{r}_1 = \mathbf{r} - \mathbf{r}'$. When $r_1 \neq 0$, Eq. (49) can be written as

$$\frac{d^2[r_1 G_0(\mathbf{r}_1,0)]}{dr_1^2} + k^2 r_1 G_0(\mathbf{r}_1,0) = 0 \tag{50}$$

which has a well-known solution

$$r_1 G_0(\mathbf{r}_1,0) = Ae^{-jkr_1} \qquad \text{or} \qquad G_0(\mathbf{r}_1,0) = A\frac{e^{-jkr_1}}{r_1} \tag{51}$$

The sign in the exponent is chosen such that Eq. (51) satisfies the radiation condition in Eq. (46). To determine the unknown coefficient A, substitute Eq. (51) into Eq. (49) and integrate over a small sphere centered at $\mathbf{r}_1 = 0$ with its radius $\epsilon \to 0$. The result is $A = (4\pi)^{-1}$. Therefore,

$$G_0(\mathbf{r}_1,0) = \frac{e^{-jkr_1}}{4\pi r_1} \tag{52}$$

and in the original coordinates, it becomes

$$G_0(\mathbf{r},\mathbf{r}') = \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \tag{53}$$

Following the same procedure, one can obtain the two-dimensional Green's function for the Helmholtz equation as

$$G_0(\boldsymbol{\rho},\boldsymbol{\rho}') = \frac{1}{4j}H_0^{(2)}(k|\boldsymbol{\rho}-\boldsymbol{\rho}'|) \tag{54}$$

where $\boldsymbol{\rho} = x\hat{x} + y\hat{y}$ and $H_0^{(2)}(k|\boldsymbol{\rho}-\boldsymbol{\rho}'|)$ is the zeroth-order Hankel function of the second kind.

When one deals with the static electric field, Maxwell's equations for $\mathbf{E}(\mathbf{r})$ reduce to

$$\nabla \times \mathbf{E}(\mathbf{r}) = 0 \qquad \text{and} \qquad \nabla \cdot \mathbf{E}(\mathbf{r}) = \frac{\rho(\mathbf{r})}{\epsilon} \tag{55}$$

These two equations can be solved conveniently by introducing the electric potential $\phi(\mathbf{r})$, which is defined as

$$\mathbf{E}(\mathbf{r}) = -\nabla\phi(\mathbf{r}) \tag{56}$$

The first equation in Eq. (55) is automatically satisfied because of the identity $\nabla \times \nabla\phi(\mathbf{r}) \equiv 0$. Substituting Eq. (56) into the second equation in Eq. (55), one obtains

$$\nabla^2\phi(\mathbf{r}) = -\frac{\rho(\mathbf{r})}{\epsilon} \tag{57}$$

This equation is known as Poisson's equation, which can be considered as a special case of Eq. (45) with $k = 0$.

Using the procedure described in this section, one obtains the three-dimensional Green's function for Poisson's equation as

$$G_0(\mathbf{r}, \mathbf{r}') = \frac{1}{4\pi|\mathbf{r} - \mathbf{r}'|} \tag{58}$$

and the two-dimensional Green's function as

$$G_0(\boldsymbol{\rho}, \boldsymbol{\rho}') = -\frac{1}{2\pi}\ln|\boldsymbol{\rho} - \boldsymbol{\rho}'| \tag{59}$$

Classification of Green's Functions

The Green's functions derived above are for the infinite unbounded space where no other objects are present. They are called the free-space Green's functions and are denoted by the subscript "0." When the region of interest is bounded, one then has to consider boundary conditions for the Green's function. Different boundary conditions lead to different Green's functions. For this reason, Green's functions are classified into three categories: Green's function of the first, second, and third kind (3).

The Green's function of the first kind, denoted by G_1, satisfies the Dirichlet boundary condition, that is,

$$G_1(\mathbf{r}, \mathbf{r}') = 0 \qquad \text{for } \mathbf{r} \text{ on } S \tag{60}$$

where S denotes the boundary of the problem. For a half space with an infinite ground plane coincident with the $z = 0$ plane, the Green's function of the first kind for Poisson's equation is given by

$$G_1(\mathbf{r}, \mathbf{r}') = G_0(\mathbf{r}, \mathbf{r}') - G_0(\mathbf{r}, \mathbf{r}_i') = \frac{1}{4\pi|\mathbf{r} - \mathbf{r}'|} - \frac{1}{4\pi|\mathbf{r} - \mathbf{r}_i'|} \tag{61}$$

where $\mathbf{r}_i' = \mathrm{r}' - 2z'\hat{z} = x'\hat{x} + y'\hat{y} - z'\hat{z}$. This result can be derived conveniently using the method of images. It is easy to see that the Dirichlet boundary condition is satisfied by $G_1(\mathbf{r}, \mathbf{r}')$ in the $z = 0$ plane.

The Green's function of the second kind, denoted by G_2, satisfies the Neumann boundary condition, that is,

$$\frac{\partial G_2(\mathbf{r}, \mathbf{r}')}{\partial n} = 0 \qquad \text{for } \mathbf{r} \text{ on } S \tag{62}$$

where S denotes the boundary of the problem and $\partial/\partial n$ denotes the normal derivative. For a half space with an infinite magnetic (symmetry) plane coincident with the $z = 0$ plane, the Green's function of the second kind for Poisson's equation is given by

$$G_2(\mathbf{r}, \mathbf{r}') = G_0(\mathbf{r}, \mathbf{r}') + G_0(\mathbf{r}, \mathbf{r}_i') = \frac{1}{4\pi|\mathbf{r} - \mathbf{r}'|} + \frac{1}{4\pi|\mathbf{r} - \mathbf{r}_i'|} \tag{63}$$

where $\mathbf{r}_i'$ is the same as the one in Eq. (61). It satisfies the Neumann boundary condition in the $z = 0$ plane.

The Green's function of the third kind is defined for problems involving two or more media. It can be denoted as $G^{(ij)}(\mathbf{r}, \mathbf{r}')$, where i indicates the medium where the field point $\mathbf{r}$ is located and j indicates the medium where the source point $\mathbf{r}'$ is located. Consider, for example, a potential problem involving two half spaces. The upper half space (medium 1) above $z = 0$ has a permittivity of ϵ_1, and the lower half space (medium 2) has a permittivityof ϵ_2. The Green's function for Poisson's equation is given by (8)

$$\begin{aligned} G^{(11)}(\mathbf{r}, \mathbf{r}') &= G_0(\mathbf{r}, \mathbf{r}') - \frac{\epsilon_2 - \epsilon_1}{\epsilon_2 + \epsilon_1} G_0(\mathbf{r}, \mathbf{r}_i') \\ &= \frac{1}{4\pi|\mathbf{r} - \mathbf{r}'|} - \frac{\epsilon_2 - \epsilon_1}{\epsilon_2 + \epsilon_1}\frac{1}{4\pi|\mathbf{r} - \mathbf{r}_i'|} \end{aligned} \tag{64}$$

and

$$G^{(21)}(\mathbf{r}, \mathbf{r}') = \frac{2\epsilon_2}{\epsilon_2 + \epsilon_1} G_0(\mathbf{r}, \mathbf{r}') = \frac{2\epsilon_2}{\epsilon_2 + \epsilon_1}\frac{1}{4\pi|\mathbf{r} - \mathbf{r}'|} \tag{65}$$

Exchanging ϵ_1 and ϵ_2 in $G^{(11)}$ and $G^{(21)}$, one obtains the expressions for $G^{(22)}$ and $G^{(12)}$, respectively. This method of obtaining the Green's functions of the third kind works only for Poisson's equation, but not for the Helmholtz equation because the standard image method is not applicable to the Helmholtz equation in this case.

Eigenfunction Expansion

In addition to the conventional method described earlier, another general method for deriving Green's functions is called the method of Ohm-Rayleigh or the method of eigenfunction expansion (3). In this section, one rederives the Green's functions in Eq. (36) and Eq. (53) to illustrate the process of the Ohm-Rayleigh method.

Consider first the solution of Eq. (27). Expand $g_0(x, x')$ in terms of a Fourier integral

$$g_0(x, x') = \int_{-\infty}^{\infty} A(h)e^{jhx}\,dh \tag{66}$$

The e^{jhx}, which is the solution of the homogeneous differential equation $d^2\psi(x)/dx^2 + h^2\psi(x) = 0$, is called the eigenfunction and h^2 is the corresponding eigenvalue. Therefore, Eq. (66) can be considered as the eigenfunction expansion of $g_0(x, x')$. To determine $A(h)$, substitute Eqs. (66) into Eq. (27) and note that

$$\delta(x - x') = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{jh(x-x')}\,dh \tag{67}$$

This yields

$$A(h) = \frac{e^{-jhx'}}{2\pi(h^2 + \gamma^2)} \tag{68}$$

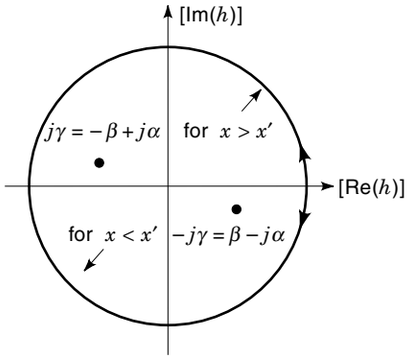


Figure 2. Locations of the two poles in the complex plane and the closed contours for integration.

Hence,

$$g_0(x, x') = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{e^{jh(x-x')}}{h^2 + \gamma^2}\, dh \tag{69}$$

This is known as the spectral representation of $g_0(x, x')$. The integral in this equation can be evaluated using Cauchy's residue theorem (9). For this, one needs to form a closed contour for the integral in Eq. (69). In order to satisfy the boundary conditions in Eqs. (25) and (26), for $x - x' > 0$ the infinite integration path must be closed in the upper half-plane and for $x - x' < 0$ the infinite path must be closed in the lower half-plane, as shown in Fig. 2. The application of Cauchy's residue theorem yields

$$g_0(x, x') = \frac{1}{2\gamma} \begin{cases} e^{-\gamma(x-x')} & \text{for } x > x' \\ e^{\gamma(x-x')} & \text{for } x < x' \end{cases} \tag{70}$$

which is the same as Eqs. (34) and (35).

Next, consider the solution of Eq. (47). First expand $G_0(\mathbf{r}, \mathbf{r}')$ in terms of Fourier integrals

$$G_0(\mathbf{r}, \mathbf{r}') = \int_{-\infty}^{\infty} A(\mathbf{h}) e^{j\mathbf{h}\cdot\mathbf{r}}\, d\mathbf{h} \tag{71}$$

where $\mathbf{h} = h_x\hat{x} + h_y\hat{y} + h_z\hat{z}$. The $e^{j\mathbf{h}\cdot\mathbf{r}}$, which is the solution of the homogeneous partial differential equation $\nabla^2\psi(\mathbf{r}) + h^2\psi(\mathbf{r}) = 0$, is called the eigenfunction and $h^2 = |\mathbf{h}|^2$ is the corresponding eigenvalue. Again, Eq. (71) can be considered as the eigenfunction expansion of $G_0(\mathbf{r}, \mathbf{r}')$. Substituting Eq. (71) into Eq. (47), and noting that

$$\delta(\mathbf{r} - \mathbf{r}') = \frac{1}{(2\pi)^3} \int_{-\infty}^{\infty} e^{j\mathbf{h}\cdot(\mathbf{r}-\mathbf{r}')}\, d\mathbf{h} \tag{72}$$

one finds

$$A(\mathbf{h}) = \frac{e^{-j\mathbf{h}\cdot\mathbf{r}'}}{(2\pi)^3(h^2 - k^2)} \tag{73}$$

Therefore,

$$G_0(\mathbf{r}, \mathbf{r}') = \frac{1}{(2\pi)^3} \int_{-\infty}^{\infty} \frac{e^{j\mathbf{h}\cdot(\mathbf{r}-\mathbf{r}')}}{h^2 - k^2}\, d\mathbf{h} \tag{74}$$

This is the spectral representation of the three-dimensional Green's function. To evaluate the spectral integral, let

$$h_x = h\sin\theta\cos\varphi, \qquad h_y = h\sin\theta\sin\varphi, \qquad h_z = h\cos\theta \tag{75}$$

so that

$$d\mathbf{h} = h^2 \sin\theta dh d\theta d\varphi \tag{76}$$

Furthermore, because of the spherical symmetry of G_0 with respect to the point $\mathbf{r}'$, the value of G_0 is independent of the direction of $\mathbf{r} - \mathbf{r}'$. Therefore, one can choose an arbitrary $\mathbf{r} - \mathbf{r}'$ for the evaluation of G_0. If one chooses the direction of $\mathbf{r} - \mathbf{r}'$ to coincide with the z-direction, Eq. (74) may be written as

$$\begin{aligned} G_0(\mathbf{r}, \mathbf{r}') &= \frac{1}{(2\pi)^3} \int_0^{\infty} \int_0^{\pi} \int_0^{2\pi} \frac{e^{jh\cos\theta|\mathbf{r}-\mathbf{r}'|}}{h^2 - k^2} h^2 \sin\theta dh d\theta d\varphi \\ &= \frac{j}{(2\pi)^2|\mathbf{r} - \mathbf{r}'|} \int_0^{\infty} [e^{-jh|\mathbf{r}-\mathbf{r}'|} - e^{jh|\mathbf{r}-\mathbf{r}'|}] \frac{hdh}{h^2 - k^2} \\ &= \frac{j}{(2\pi)^2|\mathbf{r} - \mathbf{r}'|} \int_{-\infty}^{\infty} \frac{he^{-jh|\mathbf{r}-\mathbf{r}'|}}{h^2 - k^2}\, dh \end{aligned} \tag{77}$$

This integral can now be evaluated using Cauchy's residue theorem. The integrand has two poles: one at $h = k$ and the other at $h = -k$. Although the problem considered here is lossless, treat it as a limiting case of a lossy problem for which k has a small negative imaginary part. Consequently, the pole at $h = k$ is on the lower side of the real axis and the pole at $h = -k$ is on the upper side of the real axis. In order to satisfy the radiation condition in Eq. (46), the infinite integration path must be closed in the lower half-plane, as shown in Fig. 3. Applying Cauchy's residue theorem, one obtains

$$G_0(\mathbf{r}, \mathbf{r}') = \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r} - \mathbf{r}'|} \tag{78}$$

which is the same as Eq. (53).

Finally, note that, although the process of the Ohm-Rayleigh method is more involved than the conventional method, it is more general and can be used to find Green's functions in many problems.

Green's Functions in a Bounded Region

As can be seen in the preceding section, the spectrum (eigenvalue) for infinite-space problems is continuous and, as a re-

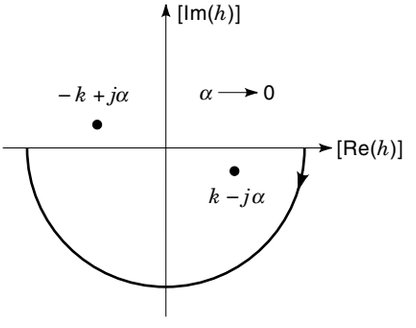


Figure 3. Locations of the two poles in the complex plane and the closed contour for integration.

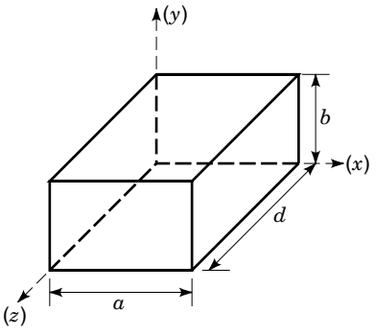


Figure 4. A grounded rectangular cavity.

sult, the spectral representation of the Green's function involves spectral integrals. When the region of interest is finite, the spectrum will be discrete. To demonstrate this fact, consider a grounded rectangular cavity of dimension $a \times b \times d$, depicted in Fig. 4. The Green's function for Poisson's equation satisfies the partial differential equation

$$\nabla^2 G_1(\mathbf{r}, \mathbf{r}') = -\delta(\mathbf{r} - \mathbf{r}') \tag{79}$$

and the Dirichlet boundary condition

$$G_1(\mathbf{r}, \mathbf{r}') = 0 \qquad \text{for } \mathbf{r} \text{ on cavity's walls} \tag{80}$$

This Green's function can be derived in a number of different ways, such as the conventional method, the method of images, and the Ohm-Rayleigh method. Here, the Ohm-Rayleigh method is employed. First, consider the solution of

$$\nabla^2 \psi + h^2 \psi = 0 \tag{81}$$

subject to the condition in Eq. (80). Using the method of separation of variables, one finds

$$\psi_{mnp} = \sin\frac{m\pi x}{a} \sin\frac{n\pi y}{b} \sin\frac{p\pi z}{d} \tag{82}$$

which is the eigenfunction of Eq. (81) with eigenvalue $h^2 = (m\pi/a)^2 + (n\pi/b)^2 + (p\pi/d)^2$. This can be used to expand G_1:

$$G_1(\mathbf{r}, \mathbf{r}') = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\sum_{p=1}^{\infty} A_{mnp} \sin\frac{m\pi x}{a} \sin\frac{n\pi y}{b} \sin\frac{p\pi z}{d} \tag{83}$$

Substituting this expression into Eq. (79), one has

$$\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\sum_{p=1}^{\infty} A_{mnp} h^2 \sin\frac{m\pi x}{a} \sin\frac{n\pi y}{b} \sin\frac{p\pi z}{d} = \delta(x - x')\delta(y - y')\delta(z - z') \tag{84}$$

The coefficient A_{mnp} can be determined by multiplying both sides by $\sin(m'\pi x/a)$ $\sin(n'\pi y/b)$, $\sin(p'\pi z/d)$ and integrating over x, y, and z. The result is

$$G_1(\mathbf{r}, \mathbf{r}') = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\sum_{p=1}^{\infty} \frac{8}{abdh^2} \sin\frac{m\pi x}{a} \sin\frac{m\pi x'}{a} \sin\frac{n\pi y}{b} \sin\frac{n\pi y'}{b} \sin\frac{p\pi z}{d} \sin\frac{p\pi z'}{d} \tag{85}$$

The triple summation can be reduced to a double summation using the formula (4)

$$\sum_{p=1}^{\infty} \frac{1}{h^2} \sin\frac{p\pi z}{d} \sin\frac{p\pi z'}{d} = \frac{d}{2k_c \sinh k_c d} \sinh k_c z_< \sinh k_c (d - z_>) \tag{86}$$

where $kc = \sqrt{(m\pi/a)^2 + (n\pi/b)^2}$, $z_< = z$ when $z < z'$, and $z_< = z'$ when $z' < z$, and $z_> = z$ when $z > z'$, and $z_> = z'$ when $z' > z$. As a result, Eq. (85) becomes

$$G_1(\mathbf{r}, \mathbf{r}') = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \frac{4}{ab} \sin\frac{m\pi x}{a} \sin\frac{m\pi x'}{a} \sin\frac{n\pi y}{b} \sin\frac{n\pi y}{b} \frac{\sinh k_c z_< \sinh k_c (d - z_>)}{k_c \sinh k_c d} \tag{87}$$

Next, consider the problem of a parallel-plate waveguide, which is finite in the y direction and infinite in the x and z directions, as shown in Fig. 5. Assuming that the source is uniform in the z direction, the Green's function of the first kind for the Helmholtz equation satisfies the partial differential equation

$$\nabla^2 G_1(\boldsymbol{\rho}, \boldsymbol{\rho}') + k^2 G_1(\boldsymbol{\rho}, \boldsymbol{\rho}') = -\delta(\boldsymbol{\rho} - \boldsymbol{\rho}') \tag{88}$$

and the boundary conditions

$$G_1(\boldsymbol{\rho}, \boldsymbol{\rho}') = 0 \qquad \text{for } y = 0, b \tag{89}$$

and the radiation conditions

$$\frac{\partial G_1(\boldsymbol{\rho}, \boldsymbol{\rho}')}{\partial x} + jkG_1(\boldsymbol{\rho}, \boldsymbol{\rho}') = 0 \qquad \text{for } x \to \infty \tag{90}$$

$$\frac{\partial G_1(\boldsymbol{\rho}, \boldsymbol{\rho}')}{\partial x} - jkG_1(\boldsymbol{\rho}, \boldsymbol{\rho}') = 0 \qquad \text{for } x \to -\infty \tag{91}$$

The eigenfunction for this problem is found as

$$\psi_n(h_x) = e^{jh_x x} \sin\frac{n\pi y}{b} \tag{92}$$

from which G_1 can be expanded as

$$G_1(\boldsymbol{\rho}, \boldsymbol{\rho}') = \int_{-\infty}^{\infty} \sum_{n=1}^{\infty} A_n(h_x) e^{jh_x x} \sin\frac{n\pi y}{b}\, dh_x \tag{93}$$

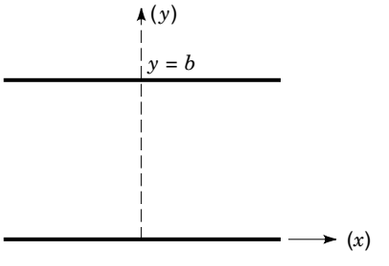


Figure 5. A parallel-plate waveguide.

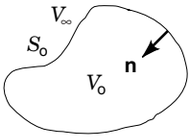


Figure 6. An object occupying volume V_o.

Substituting this expression into Eq. (88), one obtains

$$\int_{-\infty}^{\infty}\sum_{n=1}^{\infty}A_n(h_x)\left[k^2-h_x^2-\left(\frac{n\pi}{b}\right)^2\right]e^{jh_x x}\sin\frac{n\pi y}{b}\,dh_x = -\delta(x-x')\delta(y-y') \quad (94)$$

The coefficient $A_n(h_x)$ can be determined by multiplying both sides by $e^{jh_x' x}\sin(n'\pi y/b)$ and integrating over x and y. The result is

$$G_1(\boldsymbol{\rho},\boldsymbol{\rho}') = \frac{1}{\pi b}\int_{-\infty}^{\infty}\sum_{n=1}^{\infty}\left[h_x^2+\left(\frac{n\pi}{b}\right)^2-k^2\right]^{-1}e^{jh_x(x-x')}\sin\frac{n\pi y}{b}\sin\frac{n\pi y'}{b}\,dh_x \quad (95)$$

Using Cauchy's residue theorem, one can evaluate the spectral integral in a similar manner to that for the transmission line case, yielding

$$G_1(\boldsymbol{\rho},\boldsymbol{\rho}') = \frac{1}{b}\sum_{n=1}^{\infty}\frac{1}{\gamma_x}e^{-\gamma_x|x-x'|}\sin\frac{n\pi y}{b}\sin\frac{n\pi y'}{b} \quad (96)$$

where $\gamma_x = \sqrt{(n\pi/b)^2-k^2}$.

Scalar Integral Equations

Originally, Green's function methods were developed for finding the general solution of a boundary-value problem whose Green's function can be derived. For many practical problems, the Green's function cannot be derived. As a result, one must resort to a numerical method for the solution of the problem. One such numerical method is based on an integral equation derived using the Green's function method.

To demonstrate the formulation of integral equations, consider the problem of a scalar wave produced by a source $f(\mathbf{r})$ in the presence of an arbitrarily shaped object immersed in an infinite medium, as illustrated in Fig. 6. Exterior to the object, the wave function $\phi(\mathbf{r})$ satisfies the inhomogeneous Helmholtz equation in Eq. (45) and the radiation boundary condition in Eq. (46). Since the object has an arbitrary shape, no closed-form Green's function can be found for this problem. However, one can establish an integral equation for this problem using the free-space Green's function given in Eq. (53), which is the solution of Eq. (47) under the condition in Eq. (46).

First, multiply Eq. (45) with G_0, Eq. (47) with ϕ, and integrate the difference of the resultant equations over the entire exterior volume, yielding

$$\int_{V_\infty}[G_0(\mathbf{r},\mathbf{r}')\nabla^2\phi(\mathbf{r})-\phi(\mathbf{r})\nabla^2 G_0(\mathbf{r},\mathbf{r}')]\,dV = -\int_{V_s}G_0(\mathbf{r},\mathbf{r}')f(\mathbf{r})\,dV+\int_{V_\infty}\phi(\mathbf{r})\delta(\mathbf{r}-\mathbf{r}')\,dV \quad (97)$$

where V_∞ denotes the infinite space exterior to the object and V_s denotes the support of $f(\mathbf{r})$. Applying the second scalar Green's theorem (1)

$$\int_V(a\nabla^2 b-b\nabla^2 a)\,dV = \int_S\left(a\frac{\partial b}{\partial n}-b\frac{\partial a}{\partial n}\right)dS \quad (98)$$

where S denotes the surface enclosing V, one obtains

$$\int_{S_o+S_\infty}\left[G_0(\mathbf{r},\mathbf{r}')\frac{\partial\phi(\mathbf{r})}{\partial n}-\phi(\mathbf{r})\frac{\partial G_0(\mathbf{r},\mathbf{r}')}{\partial n}\right]dS - \int_{V_\infty}\phi(\mathbf{r})\delta(\mathbf{r}-\mathbf{r}')\,dV = -\int_{V_s}G_0(\mathbf{r},\mathbf{r}')f(\mathbf{r})\,dV \quad (99)$$

where S_o denotes the surface of the object and S_∞ denotes a spherical surface with a radius approaching infinity. Since both G_0 and ϕ satisfy Eq. (46), the surface integral over S_∞ vanishes. Consequently, one has

$$\int_{S_o}\left[G_0(\mathbf{r},\mathbf{r}')\frac{\partial\phi(\mathbf{r})}{\partial n}-\phi(\mathbf{r})\frac{\partial G_0(\mathbf{r},\mathbf{r}')}{\partial n}\right]dS-\int_{V_\infty}\phi(\mathbf{r})\delta(\mathbf{r}-\mathbf{r}')\,dV = -\int_{V_s}G_0(\mathbf{r},\mathbf{r}')f(\mathbf{r})\,dV \quad (100)$$

where the normal unit vector on S_o points toward the interior of the object. Using Eq. (14), one obtains

$$\int_{S_o}\left[G_0(\mathbf{r},\mathbf{r}')\frac{\partial\phi(\mathbf{r})}{\partial n}-\phi(\mathbf{r})\frac{\partial G_0(\mathbf{r},\mathbf{r}')}{\partial n}\right]dS+\int_{V_s}G_0(\mathbf{r},\mathbf{r}')f(\mathbf{r})\,dV = \begin{cases}\phi(\mathbf{r}') & \text{for } \mathbf{r}' \text{ in } V_\infty\\ 0 & \text{for } \mathbf{r}' \text{ in } V_o\end{cases} \quad (101)$$

where V_o denotes the volume of the object. Exchanging $\mathbf{r}$ and $\mathbf{r}'$ and using the symmetry property of G_0 [i.e., $G_0(\mathbf{r}', \mathbf{r}) = G_0(\mathbf{r}, \mathbf{r}')$], one has

$$\int_{S_o}\left[G_0(\mathbf{r},\mathbf{r}')\frac{\partial\phi(\mathbf{r}')}{\partial n'}-\phi(\mathbf{r})\frac{\partial G_0(\mathbf{r},\mathbf{r}')}{\partial n'}\right]dS' + \int_{V_s}G_0(\mathbf{r},\mathbf{r}')f(\mathbf{r}')\,dV' = \begin{cases}\phi(\mathbf{r}) & \text{for } \mathbf{r} \text{ in } V_\infty\\ 0 & \text{for } \mathbf{r} \text{ in } V_o\end{cases} \quad (102)$$

Equation (102) is an important result, which has several implications. First, notice that when the object is absent, the surface integral vanishes. Hence,

$$\phi(\mathbf{r}) = \int_{V_s}G_0(\mathbf{r},\mathbf{r}')f(\mathbf{r}')\,dV' \quad (103)$$

which is the same as Eq. (48). This may be called the incident field impinging on the object and be denoted as $\phi^{\text{inc}}(\mathbf{r})$. Second, when there is no source in V_∞, Eq. (102) becomes

$$\phi(\mathbf{r}) = \int_{S_o}\left[G_0(\mathbf{r},\mathbf{r}')\frac{\partial\phi(\mathbf{r}')}{\partial n'}-\phi(\mathbf{r}')\frac{\partial G_0(\mathbf{r},\mathbf{r}')}{\partial n'}\right]dS' \quad (104)$$

for $\mathbf{r}$ in V_∞. Since there is no source in V_∞, the field on S_o must be produced by the source inside S_o. This equation indicates

that the field in a source-free region can be calculated based on the knowledge of the potential and its normal derivative on the surface enclosing the region. This is the mathematical representation of the well-known Huygens' principle for a scalar wave.

Equation (102) also provides the foundation to establish an integral equation for ϕ and $\partial\phi/\partial n$ on the surface of the object. If the object is impenetrable with a hard surface where ϕ satisfies the boundary condition

$$\phi(\mathbf{r}) = 0 \qquad \text{for } \mathbf{r} \text{ on } S_o \tag{105}$$

Eq. (102) becomes

$$\phi^{\text{inc}}(\mathbf{r}) + \int_{S_o} G_0(\mathbf{r},\mathbf{r}')\frac{\partial\phi(\mathbf{r}')}{\partial n'}\,dS' = \begin{cases}\phi(\mathbf{r}) & \text{for } \mathbf{r} \text{ in } V_\infty \\ 0 & \text{for } \mathbf{r} \text{ in } V_o\end{cases} \tag{106}$$

Applying this equation on S_o, one obtains

$$\int_{S_o} G_0(\mathbf{r},\mathbf{r}')\frac{\partial\phi(\mathbf{r}')}{\partial n'}\,dS' = -\phi^{\text{inc}}(\mathbf{r}) \qquad \text{for } \mathbf{r} \text{ on } S_o \tag{107}$$

which is the integral equation for $\partial\phi/\partial n$ on S_o.

If the object is impenetrable with a soft surface where ϕ satisfies the boundary condition

$$\frac{\partial\phi(\mathbf{r})}{\partial n} = 0 \qquad \text{for } \mathbf{r} \text{ on } S_o \tag{108}$$

Eq. (102) becomes

$$\phi^{\text{inc}}(\mathbf{r}) - \int_{S_o} \phi(\mathbf{r}')\frac{\partial G_0(\mathbf{r},\mathbf{r}')}{\partial n'}\,dS' = \begin{cases}\phi(\mathbf{r}) & \text{for } \mathbf{r} \text{ in } V_\infty \\ 0 & \text{for } \mathbf{r} \text{ in } V_o\end{cases} \tag{109}$$

Applying this equation on S_o, one obtains

$$\frac{1}{2}\phi(\mathbf{r}) + \fint_{S_o} \phi(\mathbf{r}')\frac{\partial G_0(\mathbf{r},\mathbf{r}')}{\partial n'}\,dS' = \phi^{\text{inc}}(\mathbf{r}) \qquad \text{for } \mathbf{r} \text{ on } S_o \tag{110}$$

where $\fint$ denotes the integral excluding the contribution from the singular point which is known as the principal value integral. This result is obtained as follows: The integral over S_o in Eq. (109) is divided into an integral over a small circular disk with center at $\mathbf{r}$ plus the remaining integral which is represented as a principal value integral $\fint$ in the limit as the area of the isolated disk approaches zero. If $\mathbf{r}$ approaches S_o from the outside, the integral over the vanishingly small disk can be evaluated to give $-\phi(\mathbf{r})/2$. If $\mathbf{r}$ approaches S_o from the inside, the integral gives $\phi(\mathbf{r})/2$. In either case, one obtains Eq. (110), which is the integral equation for ϕ on S_o.

If the object is penetrable and homogeneous, apply Eq. (102) on S_o to obtain

$$\frac{1}{2}\phi(\mathbf{r}) - \fint_{S_o}\left[G_0(\mathbf{r},\mathbf{r}')\frac{\partial\phi(\mathbf{r}')}{\partial n'} - \phi(\mathbf{r}')\frac{\partial G_0(\mathbf{r},\mathbf{r}')}{\partial n'}\right]dS' = \phi^{\text{inc}}(\mathbf{r}) \qquad \text{for } \mathbf{r} \text{ on } S_o \tag{111}$$

To solve for ϕ and $\partial\phi/\partial n$ on S_o, another equation is needed, which can be derived by considering the interior of the object. The wave function inside the object satisfies the Helmholtz equation

$$\nabla^2\phi(\mathbf{r}) + \tilde{k}^2\phi(\mathbf{r}) = 0 \tag{112}$$

where $\tilde{k}$ characterizes the property of the object. Multiplying this equation by the Green's function for unbounded space filled with material characterized by $\tilde{k}$:

$$\tilde{G}_0(\mathbf{r},\mathbf{r}') = \frac{e^{-j\tilde{k}|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \tag{113}$$

and applying a similar derivation as before, one has

$$-\int_{S_o}\left[\tilde{G}_0(\mathbf{r},\mathbf{r}')\frac{\partial\phi(\mathbf{r}')}{\partial n'} - \phi(\mathbf{r}')\frac{\partial\tilde{G}_0(\mathbf{r},\mathbf{r}')}{\partial n'}\right]dS' = \begin{cases}0 & \text{for } \mathbf{r} \text{ in } V_\infty \\ \phi(\mathbf{r}) & \text{for } \mathbf{r} \text{ in } V_o\end{cases} \tag{114}$$

When this is applied on S_o, one obtains the second integral equation

$$\frac{1}{2}\phi(\mathbf{r}) + \fint_{S_o}\left[\tilde{G}_0(\mathbf{r},\mathbf{r}')\frac{\partial\phi(\mathbf{r}')}{\partial n'} - \phi(\mathbf{r}')\frac{\partial\tilde{G}_0(\mathbf{r},\mathbf{r}')}{\partial n'}\right]dS' = 0 \qquad \text{for } \mathbf{r} \text{ on } S_o \tag{115}$$

which can be used together with Eq. (111) for a numerical solution of ϕ and $\partial\phi/\partial n$ on S_o.

If the object is penetrable and inhomogeneous, the wave function still satisfies Eq. (112); however, $\tilde{k}$ now is a function of $\mathbf{r}$. In this case, one can write Eqs. (45) and (112) in one equation

$$\nabla^2\phi(\mathbf{r}) + k^2\phi(\mathbf{r}) = -f(\mathbf{r}) - [\tilde{k}^2(\mathbf{r}) - k^2]\phi(\mathbf{r}) \tag{116}$$

Multiplying this equation by G_0 and integrating over the infinite volume, one obtains

$$\phi(\mathbf{r}) - \int_{V_o}[\tilde{k}^2(\mathbf{r}') - k^2]G_0(\mathbf{r},\mathbf{r}')\phi(\mathbf{r}')\,dV' = \int_{V_s} G_0(\mathbf{r},\mathbf{r}')f(\mathbf{r}')\,dV' \tag{117}$$

This is the integral equation that can be used to solve for ϕ in V_o. Unlike the previous integral equations, this equation involves the volume integral. For this reason, it is often referred to as the volume integral equation, whereas the previous ones are often referred to as the surface integral equations. Integral equations for more complicated objects may involve both volume and surface integrals (10).

DYADIC GREEN'S FUNCTIONS

When both the source and response are vector functions, the corresponding Green's function is a dyad, and hence, the name dyadic Green's function.

Definition of Dyad

A dyad, denoted by $\overline{\mathbf{D}}$, is formed by two vectors

$$\overline{\mathbf{D}} = \mathbf{AB} \tag{118}$$

This entity by itself does not have any physical interpretation as a vector. However, when it acts upon another vector, the result becomes meaningful. The major role of a dyad is that its scalar product with a vector produces another vector of different magnitude and direction. For example, its anterior scalar product with vector $\mathbf{C}$ yields

$$\mathbf{C} \cdot \overline{\mathbf{D}} = (\mathbf{C} \cdot \mathbf{A})\mathbf{B} \tag{119}$$

which is a vector. Its posterior scalar product with vector $\mathbf{C}$ yields

$$\overline{\mathbf{D}} \cdot \mathbf{C} = \mathbf{A}(\mathbf{B} \cdot \mathbf{C}) \tag{120}$$

which is also a vector. Apparently, the resulting vectors in Eqs. (119) and (120) are different. In addition to the two scalar products, there are two vector products. The anterior vector product is defined as

$$\mathbf{C} \times \overline{\mathbf{D}} = (\mathbf{C} \times \mathbf{A})\mathbf{B} \tag{121}$$

and the posterior vector product is defined as

$$\overline{\mathbf{D}} \times \mathbf{C} = \mathbf{A}(\mathbf{B} \times \mathbf{C}) \tag{122}$$

Clearly, these products are dyads.

The dyad defined in Eq. (118) is a special entity, since it contains only six independent components, three in each of the two vectors. A more general dyad, also called a tensor, is defined as

$$\overline{\mathbf{D}} = \mathbf{D}_x\hat{x} + \mathbf{D}_y\hat{y} + \mathbf{D}_z\hat{z} \tag{123}$$

where $\mathbf{D}_x$, $\mathbf{D}_y$, and $\mathbf{D}_z$ are vectors. Therefore, Eq. (123) can be expressed as

$$\begin{aligned}\overline{\mathbf{D}} = {} & D_{xx}\hat{x}\hat{x} + D_{yx}\hat{y}\hat{x} + D_{zx}\hat{z}\hat{x} + D_{xy}\hat{x}\hat{y} + D_{yy}\hat{y}\hat{y} + D_{zy}\hat{z}\hat{y} \\ & + D_{xz}\hat{x}\hat{z} + D_{yz}\hat{y}\hat{z} + D_{zz}\hat{z}\hat{z}\end{aligned} \tag{124}$$

which contains nine independent components.

A special dyad is called the unit dyad or identity dyad, defined as

$$\overline{\mathbf{I}} = \hat{x}\hat{x} + \hat{y}\hat{y} + \hat{z}\hat{z} \tag{125}$$

It is evident that

$$\mathbf{C} \cdot \overline{\mathbf{I}} = \overline{\mathbf{I}} \cdot \mathbf{C} = \mathbf{C} \tag{126}$$

Free-Space Dyadic Green's Functions

Consider the electric and magnetic fields produced by an electric current source $\mathbf{J}(\mathbf{r})$ in an unbounded space. Maxwell's equations for this problem are given in Eqs. (37)–(40), which lead to Eq. (42), reproduced here as

$$\nabla \times \nabla \times \mathbf{E}(\mathbf{r}) - k^2\mathbf{E}(\mathbf{r}) = -j\omega\mu\mathbf{J}(\mathbf{r}) \tag{127}$$

The above is the vector wave equation, which is the analog of the scalar Helmholtz wave equation. It describes electromagnetic wave phenomena that are very pervasive in modern technologies, such as in communications, microwave, computer chips, and so forth.

Just as in the scalar case, one can derive a dyadic Green's function $\overline{G}_{e0}$, whose end result is to relate $\mathbf{E}(\mathbf{r})$ and $\mathbf{J}(\mathbf{r})$ by

$$\mathbf{E}(\mathbf{r}) = -j\omega\mu \int_V \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dV' \tag{128}$$

where V is the support of the current $\mathbf{J}(\mathbf{r})$. Using Eq. (128) in Eq. (127), one obtains

$$\begin{aligned}-j\omega\mu \int_V \nabla \times \nabla \times \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dV' & \\ + j\omega\mu k^2 \int_V \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dV' & \\ = -j\omega\mu\mathbf{J}(\mathbf{r}) = -j\omega\mu \int_V \overline{\mathbf{I}}\delta(\mathbf{r} - \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dV' &\end{aligned} \tag{129}$$

For arbitrary $\mathbf{J}(\mathbf{r})$, the above could be satisfied only if

$$\nabla \times \nabla \times \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') - k^2\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') = \overline{\mathbf{I}}\delta(\mathbf{r} - \mathbf{r}') \tag{130}$$

The $\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}')$ is called the dyadic Green's function of the electric type that relates vector field $\mathbf{E}$ to vector current $\mathbf{J}$.

Taking the curl of Eq. (128) and using Maxwell's equations, one obtains

$$\mathbf{H}(\mathbf{r}) = \int_V \nabla \times \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dV' = \int_V \overline{\mathbf{G}}_{m0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dV' \tag{131}$$

where $\overline{\mathbf{G}}_{m0}(\mathbf{r}, \mathbf{r}') = \nabla \times \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}')$ is called the dyadic Green's function of the magnetic type. It satisfies the equation

$$\nabla \times \nabla \times \overline{\mathbf{G}}_{m0}(\mathbf{r}, \mathbf{r}') - k^2\overline{\mathbf{G}}_{m0}(\mathbf{r}, \mathbf{r}') = \nabla \times [\overline{\mathbf{I}}\delta(\mathbf{r} - \mathbf{r}')] \tag{132}$$

Therefore, the task of finding the dyadic Green's function of the electric type is reduced to the task of solving Eq. (130). Equation (130) can be made less difficult by taking the posterior scalar product with an arbitrary vector $\mathbf{a}$, yielding

$$\nabla \times \nabla \times \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{a} - k^2\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{a} = \mathbf{a}\delta(\mathbf{r} - \mathbf{r}') \tag{133}$$

Recognizing that $\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{a}$ represents a vector, one may use the vector identity $\nabla \times \nabla \times \mathbf{A} = \nabla(\nabla \cdot \mathbf{A}) - \nabla^2\mathbf{A}$ to find

$$\begin{aligned}-\nabla^2\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{a} - k^2\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{a} & \\ = \mathbf{a}\delta(\mathbf{r} - \mathbf{r}') - \nabla[\nabla \cdot \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{a}] &\end{aligned} \tag{134}$$

Taking the divergence of Eq. (133) and making use of the fact that $\nabla \cdot (\nabla \times \mathbf{A}) \equiv 0$, it can be seen that

$$\nabla \cdot \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{a} = -\frac{1}{k^2}\nabla \cdot [\mathbf{a}\delta(\mathbf{r} - \mathbf{r}')] \tag{135}$$

Using this in Eq. (134), one obtains

$$\nabla^2\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{a} + k^2\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{a} = -\left(1 + \frac{\nabla\nabla\cdot}{k^2}\right)[\mathbf{a}\delta(\mathbf{r} - \mathbf{r}')] \tag{136}$$

By making use of the fact that

$$\nabla^2 G_0(\mathbf{r}, \mathbf{r}') + k^2 G_0(\mathbf{r}, \mathbf{r}') = -\delta(\mathbf{r} - \mathbf{r}') \tag{137}$$

and the fact that $1 + \nabla\nabla\cdot/k^2$ is a linear operator that commutes with ∇^2, it can be deduced that

$$\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\cdot\mathbf{a} = \left(1+\frac{\nabla\nabla\cdot}{k^2}\right)[\mathbf{a}G_0(\mathbf{r},\mathbf{r}')] \tag{138}$$

Writing the above as

$$\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\cdot\mathbf{a} = \left(\overline{\mathbf{I}}+\frac{\nabla\nabla}{k^2}\right)G_0(\mathbf{r},\mathbf{r}')\cdot\mathbf{a} \tag{139}$$

and since $\mathbf{a}$ is an arbitrary vector, one deduces that

$$\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}') = \left(\overline{\mathbf{I}}+\frac{\nabla\nabla}{k^2}\right)G_0(\mathbf{r},\mathbf{r}') \tag{140}$$

The free-space dyadic Green's function of the magnetic type can be derived as

$$\overline{\mathbf{G}}_{m0}(\mathbf{r},\mathbf{r}') = \nabla\times\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}') = \nabla\times[\overline{\mathbf{I}}G_0(\mathbf{r},\mathbf{r}')] = \nabla G_0(\mathbf{r},\mathbf{r}')\times\overline{\mathbf{I}} \tag{141}$$

The above is the explicit representation of the dyadic Green's function in terms of the scalar Green's function $G_0(\mathbf{r}, \mathbf{r}')$. It is to be noted that the aforementioned relationship between the dyadic Green's function and the scalar Green's function $G_0(\mathbf{r}, \mathbf{r}')$ is only valid for a homogeneous unbounded space such as a free space. Such a relation does not hold true in a cavity, waveguide, or half-space. For example, the dyadic Green's functions for a half-space (above $z = 0$) are given by (3)

$$\overline{\mathbf{G}}_{e1}(\mathbf{r},\mathbf{r}') = \left(\overline{\mathbf{I}}-\frac{\nabla\nabla'}{k^2}\right)[G_0(\mathbf{r},\mathbf{r}')-G_0(\mathbf{r},\mathbf{r}_i')] + 2\hat{z}\hat{z}G_0(\mathbf{r},\mathbf{r}_i') \tag{142}$$

and

$$\overline{\mathbf{G}}_{m2}(\mathbf{r},\mathbf{r}') = \nabla G_0(\mathbf{r},\mathbf{r}')\times\overline{\mathbf{I}} + \nabla G_0(\mathbf{r},\mathbf{r}_i')\times\overline{\mathbf{I}}_i \tag{143}$$

where $\mathbf{r}_i' = x'\hat{x} + y'\hat{y} - z'\hat{z}$ and $\overline{\mathbf{I}}_i = -\overline{\mathbf{I}} + 2\hat{z}\hat{z}$. It can be verified that $\overline{\mathbf{G}}_{e1}(\mathbf{r}, \mathbf{r}')$ and $\overline{\mathbf{G}}_{m2}(\mathbf{r}, \mathbf{r}')$ satisfy the boundary conditions

$$\hat{z}\times\overline{\mathbf{G}}_{e1}(\mathbf{r},\mathbf{r}') = 0 \qquad \text{for } z = 0 \tag{144}$$

and

$$\hat{z}\times\nabla\times\overline{\mathbf{G}}_{m2}(\mathbf{r},\mathbf{r}') = 0 \qquad \text{for } z = 0 \tag{145}$$

respectively. For this reason, $\overline{\mathbf{G}}_{e1}(\mathbf{r}, \mathbf{r}')$ is called the electric-type dyadic Green's function of the first kind and $\overline{\mathbf{G}}_{m2}(\mathbf{r}, \mathbf{r}')$ is called the magnetic-type dyadic Green's function of the second kind. The classification of dyadic Green's functions is similar to that of scalar Green's functions.

Eigenfunction Expansion

As in the scalar case, the Ohm–Rayleigh method or the method of eigenfunction expansion is a general method to derive dyadic Green's functions (3). For vector problems, the eigenfunctions are vector functions, known as vector wave functions. There are three kinds of vector wave functions (1), defined by

$$\mathbf{L}(\mathbf{r}) = \nabla\psi(\mathbf{r}) \tag{146}$$

$$\mathbf{M}(\mathbf{r}) = \nabla\times[c\psi(\mathbf{r})] \tag{147}$$

$$\mathbf{N}(\mathbf{r}) = \frac{1}{\kappa}\nabla\times\mathbf{M}(\mathbf{r}) \tag{148}$$

where $\mathbf{c}$ is a vector called the pilot vector and ψ satisfies the homogeneous Helmholtz wave equation

$$\nabla^2\psi(\mathbf{r}) + \kappa^2\psi(\mathbf{r}) = 0 \tag{149}$$

It can be shown that $\mathbf{L}$, $\mathbf{M}$, and $\mathbf{N}$ satisfy the vector equations

$$\nabla^2\mathbf{L}(\mathbf{r}) + \kappa^2\mathbf{L}(\mathbf{r}) = 0 \tag{150}$$

$$\nabla\times\nabla\times\mathbf{M}(\mathbf{r}) - \kappa^2\mathbf{M}(\mathbf{r}) = 0 \tag{151}$$

$$\nabla\times\nabla\times\mathbf{N}(\mathbf{r}) - \kappa^2\mathbf{N}(\mathbf{r}) = 0 \tag{152}$$

and $\mathbf{M}$ can be expressed in terms of $\mathbf{N}$ as

$$\mathbf{M}(\mathbf{r}) = \frac{1}{\kappa}\nabla\times\mathbf{N}(\mathbf{r}) \tag{153}$$

Since $\nabla\times\mathbf{L}(\mathbf{r}) = \nabla\times\nabla\psi(\mathbf{r}) = 0$, $\mathbf{L}$ is known as the irrotational vector wave function. Since $\nabla\cdot\mathbf{M}(\mathbf{r}) = 0$ and $\nabla\cdot\mathbf{N}(\mathbf{r}) = 0$, $\mathbf{M}$ and $\mathbf{N}$ are known as the solenoidal vector wave functions.

For a rectangular waveguide illustrated in Fig. 7, ψ is given by

$$\psi_{\substack{e\\o}mn}(h,\mathbf{r}) = \begin{Bmatrix}\cos k_x x\cos k_y y\\ \sin k_x x\sin k_y y\end{Bmatrix} e^{-jhz} \tag{154}$$

where $k_x = m\pi/a$ and $k_y = n\pi/b$. The vector wave functions $\mathbf{L}$, $\mathbf{M}$, and $\mathbf{N}$ are given by

$$\mathbf{L}_{\substack{e\\o}mn}(h,\mathbf{r}) = \nabla\psi_{\substack{e\\o}mn}(h,\mathbf{r}) \tag{155}$$

$$\mathbf{M}_{\substack{e\\o}mn}(h,\mathbf{r}) = \nabla\times[\hat{z}\psi_{\substack{e\\o}mn}(h,\mathbf{r})] \tag{156}$$

$$\mathbf{N}_{\substack{e\\o}mn}(h,\mathbf{r}) = \frac{1}{\kappa}\nabla\times\nabla\times[\hat{z}\psi_{\substack{e\\o}mn}(h,\mathbf{r})] \tag{157}$$

where the pilot vector is $\mathbf{c} = \hat{z}$. This causes $\mathbf{M}$ to be transverse to $\hat{z}$.

The vector wave functions are always orthogonal to each other. For those in a rectangular waveguide, it can be shown

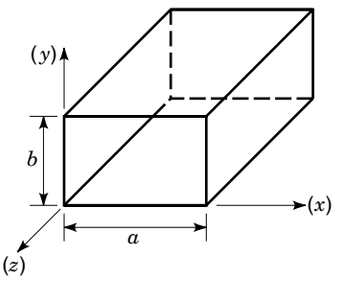


Figure 7. A rectangular waveguide.

that

$$\int_V \mathbf{U}_{\substack{e\\o}mn}(h, \mathbf{r}) \cdot \mathbf{V}_{\substack{e\\o}m'n'}(-h', \mathbf{r})\, dV = 0 \tag{158}$$

where $\mathbf{U}$, $\mathbf{V} = \mathbf{L}$, $\mathbf{M}$, $\mathbf{N}$, except when $\mathbf{U}_{\substack{e\\o}mn}(h, \mathbf{r}) = \mathbf{V}_{\substack{e\\o}mn}(h, \mathbf{r})$. They form a complete set and, therefore, can be employed to expand any vector functions.

The electric-type dyadic Green's function of the first kind satisfies the equation

$$\nabla \times \nabla \times \overline{\mathbf{G}}_{e1}(\mathbf{r}, \mathbf{r}') - k^2 \overline{\mathbf{G}}_{e1}(\mathbf{r}, \mathbf{r}') = \overline{\mathbf{I}}\delta(\mathbf{r} - \mathbf{r}') \tag{159}$$

and the boundary condition

$$\hat{n} \times \overline{\mathbf{G}}_{e1}(\mathbf{r}, \mathbf{r}') = 0 \qquad \text{on the waveguide walls} \tag{160}$$

It is clear that only $\mathbf{L}_{omn}$, $\mathbf{M}_{emn}$, and $\mathbf{N}_{omn}$ satisfy Eq. (160) and, therefore, can be used to expand $\overline{\mathbf{G}}_{e1}$:

$$\begin{aligned}\overline{\mathbf{G}}_{e1}(\mathbf{r}, \mathbf{r}') = \int_{-\infty}^{\infty} \sum_{m,n} [&\mathbf{L}_{omn}(h, \mathbf{r})\mathbf{A}_{omn}(h) + \mathbf{M}_{emn}(h, \mathbf{r})\mathbf{B}_{emn}(h) \\ &+ \mathbf{N}_{omn}(h, \mathbf{r})\mathbf{C}_{omn}(h)]\, dh\end{aligned} \tag{161}$$

Substituting this expansion into Eq. (159), one obtains

$$\begin{aligned}\int_{-\infty}^{\infty} \sum_{m,n} \{&-k^2 \mathbf{L}_{omn}(h, \mathbf{r})\mathbf{A}_{omn}(h) + (\kappa^2 - \kappa^2)[\mathbf{M}_{emn}(h, \mathbf{r})\mathbf{B}_{emn}(h) \\ &+ \mathbf{N}_{omn}(h, \mathbf{r})\mathbf{C}_{omn}(h)]\}\, dh = \mathbf{I}\delta(\mathbf{r} - \mathbf{r}')\end{aligned} \tag{162}$$

Taking the anterior scalar product of Eq. (162) with $\mathbf{L}_{om'n'}(-h', \mathbf{r})$, $\mathbf{M}_{em'n'}(-h', \mathbf{r})$, and $\mathbf{N}_{om'n'}(-h', \mathbf{r})$, respectively, integrating over the entire volume of the waveguide, and applying the orthogonal relation in Eq. (158), one can find

$$\mathbf{A}_{omn}(h) = -\frac{k_{cmn}^2}{k^2\kappa^2} C_{mn} \mathbf{L}_{omn}(-h, \mathbf{r}') \tag{163}$$

$$\mathbf{B}_{emn}(h) = \frac{1}{\kappa^2 - k^2} C_{mn} \mathbf{M}_{emn}(-h, \mathbf{r}') \tag{164}$$

$$\mathbf{C}_{emn}(h) = \frac{1}{\kappa^2 - k^2} C_{mn} \mathbf{N}_{omn}(-h, \mathbf{r}') \tag{165}$$

where $k_{cmn}^2 = k_x^2 + k_y^2$ and $C_{mn} = (2 - \delta_0)/(\pi ab k_{cmn}^2)$ with $\delta_0 = 1$ when $m = 0$ or $n = 0$ and $\delta_0 = 0$, where both m and n are nonzero. Therefore,

$$\begin{aligned}\overline{\mathbf{G}}_{e1}(\mathbf{r}, \mathbf{r}') = \int_{-\infty}^{\infty} \sum_{m,n} C_{mn} \Bigg\{ &-\frac{k_{cmn}^2}{k^2\kappa^2} \mathbf{L}_{omn}(h, \mathbf{r}) \mathbf{L}_{omn}(-h, \mathbf{r}') \\ &+ \frac{1}{\kappa^2 - k^2} [\mathbf{M}_{emn}(h, \mathbf{r}) \mathbf{M}_{emn}(-h, \mathbf{r}') \\ &+ \mathbf{N}_{omn}(h, \mathbf{r}) \mathbf{N}_{omn}(-h, \mathbf{r}')] \Bigg\}\, dh\end{aligned} \tag{166}$$

Through some mathematical manipulations and the application of Cauchy's residue theorem (3), one can simplify Eq. (166) as

$$\begin{aligned}\overline{\mathbf{G}}_{e1}(\mathbf{r}, \mathbf{r}') = &-\frac{1}{k^2}\hat{z}\hat{z}\delta(\mathbf{r} - \mathbf{r}') - \frac{j}{ab} \sum_{m,n} \frac{2 - \delta_0}{k_{cmn}^2 k_{gmn}} \\ &[\mathbf{M}_{emn}(\pm k_{gmn}, \mathbf{r}) \mathbf{M}_{emn}(\pm k_{gmn}, \mathbf{r}) \mathbf{M}_{emn}(\mp k_{gmn}, \mathbf{r}') \\ &+ \mathbf{N}_{omn}(\pm k_{gmn}, \mathbf{r}) \mathbf{N}_{omn}(\mp k_{gmn}, \mathbf{r}')] \quad z \gtrless z'\end{aligned} \tag{167}$$

where $k_{gmn} = \sqrt{k^2 - k_{cmn}^2}$.

In addition to the method described above, Tai (3) proposed the method of $\overline{\mathbf{G}}_m$, in which $\overline{\mathbf{G}}_m$ is derived first and $\overline{\mathbf{G}}_e$ is then derived from $\nabla \times \overline{\mathbf{G}}_m = \overline{\mathbf{I}}\delta(\mathbf{r} - \mathbf{r}') + k^2 \overline{\mathbf{G}}_e$. Since $\overline{\mathbf{G}}_m$ is completely solenoidal, its expansion requires only $\mathbf{M}$ and $\mathbf{N}$ and, therefore, the derivation becomes simpler.

The Ohm–Rayleigh method can be used to derive a variety of dyadic Green's functions. Table 1 lists the problems for which the dyadic Green's functions have been derived.

Table 1. Problems with Available Dyadic Green's Functions

Geometry of Problem	References
Parallel-plate waveguide	(3)
Rectangular waveguide	(3), (11)
Rectangular waveguide with two dielectrics	(3), (12)
Cylindrical waveguide	(3), (13), (14)
Coaxial waveguide	(3), (15)
Rectangular cavity	(3), (16)
Cylindrical cavity	(3), (13), (17)
Spherical cavity	(3), (18)
Circular conducting cylinder	(3)
Circular dielectric cylinder	(3)
Circular coated cylinder	(3)
Elliptical conducting cylinder	(3)
Conducting wedge and half-sheet	(3)
Conducting sphere and cone	(3)
Homogeneous and inhomogeneous spheres	(3)
Planar layered medium	(3), (5), (6)
Planar anisotropic layered medium	(19)
Conductor-backed layered medium	(20), (21), (22)
Cylindrically layered medium	(6), (23)
Spherically layered medium	(3), (6), (24)
Moving medium	(3), (25), (26)

Vector Integral Equations

Consider the problem of the electric and magnetic fields produced by an electric current source $\mathbf{J}(\mathbf{r})$ in the presence of an arbitrarily shaped object immersed in an infinite homogeneous medium (see Fig. 6). Exterior to the object, the electric field satisfies the vector wave equation in Eq. (127) and the radiation condition at infinity is given by

$$r[\nabla \times \mathbf{E}(\mathbf{r}) + jk\hat{r} \times \mathbf{E}(\mathbf{r})] = 0 \qquad \text{for } r \to \infty \tag{168}$$

Multiplying Eq. (127) by $\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}')$, Eq. (130) by $\mathbf{E}(\mathbf{r})$, and integrating the difference of the resultant equations over the exterior region, one obtains

$$\begin{aligned}\int_{V_\infty} \{&[\nabla \times \nabla \times \mathbf{E}(\mathbf{r})] \cdot \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') - \mathbf{E}(\mathbf{r}) \cdot [\nabla \times \nabla \times \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}')]\}\, dV \\ &= -j\omega\mu \int_{V_s} \mathbf{J}(\mathbf{r}) \cdot \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}')\, dV - \int_{V_\infty} \mathbf{E}(\mathbf{r}) \cdot \overline{\mathbf{I}}\delta(\mathbf{r} - \mathbf{r}')\, dV\end{aligned} \tag{169}$$

where V_∞ denotes the infinite space exterior to the object and V_s denotes the support of $\mathbf{J}(\mathbf{r})$. Applying the vector-dyadic Green's second identity (27)

$$\int_V \left[(\nabla\times\nabla\times\mathbf{A})\cdot\overline{\mathbf{D}} - \mathbf{A}\cdot(\nabla\times\nabla\times\overline{\mathbf{D}})\right] dV = \int_S \left[(\hat{n}\times\mathbf{A})\cdot(\nabla\times\overline{\mathbf{D}}) + (\hat{n}\times\nabla\times\mathbf{A})\cdot\overline{\mathbf{D}}\right] dS \quad (170)$$

where V is a volume enclosed by S, one has

$$\int_{S_o+S_\infty} \left\{[\hat{n}\times\mathbf{E}(\mathbf{r})]\cdot[\nabla\times\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')] + \hat{n}\times\nabla\times\mathbf{E}(\mathbf{r})]\cdot\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\right\} dS = -j\omega\mu\int_{V_s}\mathbf{J}(\mathbf{r})\cdot\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\,dV - \int_{V_\infty}\mathbf{E}(\mathbf{r})\delta(\mathbf{r}-\mathbf{r}')\,dV \quad (171)$$

where S_o denotes the surface of the object, S_∞ denotes a large spherical surface whose radius approaches infinity, and $\hat{n}$ is the normal unit vector pointing away from V_∞. Since both $\mathbf{E}(\mathbf{r})$ and $\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')$ satisfy the radiation condition, the surface integral over S_∞ vanishes. As a result,

$$-\int_{S_o}\left\{[\hat{n}\times\mathbf{E}(\mathbf{r})]\cdot[\nabla\times\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')] + [\hat{n}\times\nabla\times\mathbf{E}(\mathbf{r})]\cdot\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\right\} dS - j\omega\mu\int_{V_s}\mathbf{J}(\mathbf{r})\cdot\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\,dV = \begin{cases}\mathbf{E}(\mathbf{r}') & \text{for } \mathbf{r}' \text{ in } V_\infty\\ 0 & \text{for } \mathbf{r}' \text{ in } V_o\end{cases} \quad (172)$$

which can also be written as

$$-\int_{S_o}\left\{[\nabla\times\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')]\cdot[\hat{n}'\times\mathbf{E}(\mathbf{r}')] - j\omega\mu\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\cdot[\hat{n}'\times\mathbf{H}(\mathbf{r}')]\right\} dS' - j\omega\mu\int_{V_s}\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')\,dV' = \begin{cases}\mathbf{E}(\mathbf{r}) & \text{for } \mathbf{r} \text{ in } V_\infty\\ 0 & \text{for } \mathbf{r} \text{ in } V_o\end{cases} \quad (173)$$

where V_o denotes the volume of the object.

Similar to Eq. (102) in the scalar case, Eq. (173) is an important result, which has several implications. First, notice that when the object is absent, the surface integral vanishes. Hence,

$$\mathbf{E}(\mathbf{r}) = -j\omega\mu\int_{V_s}\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')\,dV' \quad (174)$$

which is the same as Eq. (128). The above can be regarded as the incident field and denoted as $\mathbf{E}^{\text{inc}}(\mathbf{r})$. Second, when there is no source in V_∞, Eq. (173) becomes

$$\mathbf{E}(\mathbf{r}) = -\int_{S_o}\left\{[\nabla\times\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')]\cdot[\hat{n}'\times\mathbf{E}(\mathbf{r}')] - j\omega\mu\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\cdot[\hat{n}'\times\mathbf{H}(\mathbf{r}')]\right\} dS' \quad (175)$$

for $\mathbf{r}$ in V_∞. Since there is no source in V_∞, the field on S_o must be produced by the source inside S_o. This equation indicates that the field in a source-free region can be calculated based on the knowledge of the tangential electric and magnetic fields on the surface enclosing the region. This is the mathematical representation of the well-known Huygens' principle for a vector field.

Equation (173) also provides the foundation to establish an integral equation for $\hat{n}\times\mathbf{E}$ and $\hat{n}\times\mathbf{H}$ on the surface of the object. If the object is a perfect conductor, $\hat{n}\times\mathbf{E}(\mathbf{r}) = 0$ for $\mathbf{r}$ on S_o. Consequently, Eq. (173) becomes

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{\text{inc}}(\mathbf{r}) + j\omega\mu\int_{S_o}\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\cdot[\hat{n}'\times\mathbf{H}(\mathbf{r}')]\,dS' \quad (176)$$

for $\mathbf{r}$ in V_∞. Substituting this into $\hat{n}\times\mathbf{E}(\mathbf{r}) = 0$ for $\mathbf{r}$ on S_o, we obtain an integral equation, which can be solved for $\hat{n}\times\mathbf{H}(\mathbf{r})$.

If the object is a homogeneous body, one can derive another integral representation for the field inside S_o using the unbounded-space dyadic Green's function for the interior medium. When this and Eq. (173) are applied at S_o, one obtains two integral equations, which can be solved for $\hat{n}\times\mathbf{E}(\mathbf{r})$ and $\hat{n}\times\mathbf{H}(\mathbf{r})$.

If the object is an inhomogeneous dielectric body, the electric field satisfies the vector wave equation

$$\nabla\times\nabla\times\mathbf{E}(\mathbf{r}) - \tilde{k}^2(\mathbf{r})\mathbf{E}(\mathbf{r}) = -j\omega\mu\mathbf{J}(\mathbf{r}) \quad (177)$$

This can be written as

$$\nabla\times\nabla\times\mathbf{E}(\mathbf{r}) - k^2\mathbf{E}(\mathbf{r}) = -j\omega\mu\mathbf{J}(\mathbf{r}) + [\tilde{k}^2(\mathbf{r}) - k^2]\mathbf{E}(\mathbf{r}) \quad (178)$$

Multiplying Eq. (177) by $\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')$ and integrating the resultant equation over the entire space, one obtains

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}^{\text{inc}}(\mathbf{r}) + \int_{V_o}\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\cdot[\tilde{k}^2(\mathbf{r}') - k^2]\mathbf{E}(\mathbf{r}')\,dV' \quad (179)$$

This is the mathematical representation of the volume equivalence principle. It provides a volume integral equation which can be solved for $\mathbf{E}(\mathbf{r})$.

We note that the formulation described in this section can be repeated for the magnetic field in a similar manner. As a result, different integral equations exist for the same problem, which provide different approaches to the solution of the problem.

Singularity of the Dyadic Green's Function

As shown in Eq. (128), the electric field produced by the current $\mathbf{J}$ in an unbounded space can be written as

$$\mathbf{E}(\mathbf{r}) = -j\omega\mu\int_V\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')\,dV' \quad (180)$$

where $\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')$ is defined by Eq. (140). Many electromagneticists have tried to fathom the meaning of Eq. (180). Strictly speaking, the integral does not converge because of the $1/|\mathbf{r}-\mathbf{r}'|$ singularity in $G_0(\mathbf{r},\mathbf{r}')$. After being operated upon by the double ∇ operator in Eq. (140), $\overline{\mathbf{G}}_{e0}(\mathbf{r},\mathbf{r}')$ contains terms of the form $1/|\mathbf{r}-\mathbf{r}'|^3$, rendering the integral in Eq. (180) ill-defined. A remedy to this is to rewrite Eq. (180) as

$$\mathbf{E}(\mathbf{r}) = -j\omega\mu\left(\overline{\mathbf{I}} + \frac{\nabla\nabla}{k^2}\right)\cdot\int_V G_0(\mathbf{r},\mathbf{r}')\mathbf{J}(\mathbf{r}')\,dV' \quad (181)$$

This equation is well-defined for all $\mathbf{r}$ and $\mathbf{r}'$, but lacks the compactness of Eq. (180).

Equation (180) can be made meaningful in a generalized function sense. To this end, one defines $\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}')$ as a generalized function

$$\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') = PV\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') - \frac{\overline{\mathbf{L}}\delta(\mathbf{r} - \mathbf{r}')}{k^2} \tag{182}$$

where PV implies the invokement of a principal volume integral whose value depends on the shape of the principal volume chosen. For the sake of uniqueness, $\overline{\mathbf{L}}$ also depends on the shape of the principal volume. A principal volume integral is defined as

$$\begin{aligned}\int_V PV\overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dV' &= PV \int_V \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dV' \\ &= \lim_{V_\delta \to 0} \int_{V - V_\delta} \overline{\mathbf{G}}_{e0}(\mathbf{r}, \mathbf{r}') \cdot \mathbf{J}(\mathbf{r}')\, dV'\end{aligned} \tag{183}$$

where V_δ is called the exclusion volume. Unfortunately, even though the integral above converges, its value is nonunique in the sense that it depends on the shape of V_δ. The nonuniqueness in the first term of Eq. (182) is rectified by the choice of a shape-dependent $\overline{\mathbf{L}}$. The value of $\overline{\mathbf{L}}$ in Eq. (182) for various exclusion volumes is given by (28–33)

$$\overline{\mathbf{L}} = \frac{\overline{\mathbf{I}}}{3} \qquad \text{for spheres and cubes} \tag{184}$$

$$\overline{\mathbf{L}} = \hat{z}\hat{z} \qquad \text{for disks perpendicular to the } z\text{-axis} \tag{185}$$

$$\overline{\mathbf{L}} = \frac{\hat{x}\hat{x} + \hat{y}\hat{y}}{2} \qquad \text{for needles parallel to the } z\text{-axis} \tag{186}$$

To understand how the above $\overline{\mathbf{L}}$s are derived, one can start with the classically legitimate Eq. (181) and split the integral into two terms with the definition of a principal volume integral:

$$\begin{aligned}\mathbf{E}(\mathbf{r}) = &- j\omega\mu \lim_{V_\delta \to 0} \left(\overline{\mathbf{I}} + \frac{\nabla\nabla}{k^2}\right) \cdot \int_{V - V_\delta} G_0(\mathbf{r}, \mathbf{r}')\mathbf{J}(\mathbf{r}')\, dV' \\ &- j\omega\mu \lim_{V_\delta \to 0} \left(\overline{\mathbf{I}} + \frac{\nabla\nabla}{k^2}\right) \cdot \int_{V_\delta} G_0(\mathbf{r}, \mathbf{r}')\mathbf{J}(\mathbf{r}')\, dV'\end{aligned} \tag{187}$$

In the first term, $V_\delta > 0$ and hence, it is legitimate to exchange the order of differentiation and integration so that it becomes the first term of Eq. (182). In the second term in Eq. (187), the term that does not allow the exchange of the order of integration and differentiation is the term involving the $\nabla\nabla$ operator. Focusing on it more carefully, one has

$$\begin{aligned}&\lim_{V_\delta \to 0} \nabla\nabla \cdot \int_{V_\delta} G_0(\mathbf{r}, \mathbf{r}')\mathbf{J}(\mathbf{r}')\, dV' \\ &= \lim_{V_\delta \to 0} \nabla \left[\int_{V_\delta} G_0(\mathbf{r}, \mathbf{r}')\nabla' \cdot \mathbf{J}(\mathbf{r}')\, dV' - \int_{S_\delta} \hat{n}' \cdot \mathbf{J}(\mathbf{r}')G_0(\mathbf{r}, \mathbf{r}')\, dS'\right]\end{aligned} \tag{188}$$

To arrive at the above, one has made use of the fact that $\nabla G_0(\mathbf{r}, \mathbf{r}') = -\nabla' G_0(\mathbf{r}, \mathbf{r}')$, and $[\nabla' G_0(\mathbf{r}, \mathbf{r}')] \cdot \mathbf{J}(\mathbf{r}') = \nabla' \cdot [G_0(\mathbf{r}, \mathbf{r}')\mathbf{J}(\mathbf{r}')] - G_0(\mathbf{r}, \mathbf{r}')\nabla' \cdot \mathbf{J}(\mathbf{r}')$. Using Gauss's theorem on the term involving $\nabla' \cdot [G_0(\mathbf{r}, \mathbf{r}')\mathbf{J}(\mathbf{r}')]$ finally gives rise to Eq. (188). The first integral on the right-hand side of Eq. (188) vanishes since if $\mathbf{J}(\mathbf{r}')$ is regular, $\nabla' \cdot \mathbf{J}(\mathbf{r}') = \rho(\mathbf{r}')/j\omega$ is also regular and the integral is finally proportional to V_δ. In the second integral, S_δ is the surface bounding V_δ. Hence, $\hat{n}' \cdot \mathbf{J}(\mathbf{r}')$ is the surface charge on S_δ due to the sudden truncation of $\mathbf{J}(\mathbf{r}')$ within the volume V_δ. This integral gives the potential observed within V_δ due to this surface charge, and it is nonzero even when $V_\delta \to 0$. The gradient (outside the brackets) in turn yields the field generated by this surface charge. In other words, surface charges of opposite polarities on the wall of an infinitesimally small volume always generate a finite field within the small volume. This fact is also intimately related to the scale invariant nature of the Laplace equation which is Maxwell's equations at low frequency.

BIBLIOGRAPHY

1. J. A. Stratton, *Electromagnetic Theory,* New York: McGraw-Hill, 1941.
2. P. M. Morse and H. Feshbach, *Methods of Theoretical Physics,* New York: McGraw-Hill, 1953.
3. C. T. Tai, *Dyadic Green Functions in Electromagnetic Theory,* 2nd ed., New York: IEEE Press, 1994.
4. R. E. Collin, *Field Theory of Guided Waves,* 2nd ed., New York: IEEE Press, 1991.
5. J. A. Kong, *Electromagnetic Wave Theory,* 2nd ed., New York: Wiley, 1990.
6. W. C. Chew, *Waves and Fields in Inhomogeneous Media,* New York: IEEE Press, 1995.
7. I. M. Gel'fand and G. E. Shilov, *Generalized Functions,* New York: Academic Press, 1964.
8. J. D. Jackson, *Classical Electrodynamics,* 2nd ed., New York: Wiley, 1975.
9. E. T. Copson, *Theory of Functions of a Complex Variable,* London: Oxford University Press, 1948.
10. J. M. Jin, V. V. Liepa, and C. T. Tai, A volume-surface integral equation for electro-magnetic scattering by inhomogeneous cylinders, *J. Electromagn. Waves Appl.,* **2**: 573–588, 1988.
11. C. T. Tai, On the eigenfunction expansion of dyadic Green's functions, *Proc. IEEE,* **61**: 480–481, 1973.
12. C. T. Tai, Dyadic Green's functions for a rectangular waveguide filled with two dielectrics, *J. Electromagn. Waves Appl.,* **2**: 245–253, 1988.
13. M. Kisliuk, The dyadic Green's functions for cylindrical waveguides and cavities, *IEEE Trans. Microw. Theory Tech.,* **28**: 894–898, 1980.
14. V. Daniel, New expressions of dyadic Green's functions in uniform waveguides with perfectly conducting walls, *IEEE Trans. Antennas Propag.,* **30**: 487–499, 1982.
15. C. T. Tai, Dyadic Green's functions for a coaxial line, *IEEE Trans. Antennas Propag.,* **31**: 355–358, 1983.
16. C. T. Tai and P. Rozenfeld, Different representations of dyadic Green's functions for a rectangular cavity, *IEEE Trans. Microw. Theor. Tech.,* **24**: 597–601, 1976.
17. S. Zhang and J. M. Jin, Derivation of dyadic Green's functions for cylindrical cavities by image method, *Acta Electronica Sinica,* **12** (5): 21–26, 1984.
18. R. E. Collin, Dyadic Green's function expansion in spherical coordinates, *Electromagn.,* **6**: 183–207, 1986.
19. J. K. Lee and J. A. Kong, Dyadic Green's functions for layered anisotropic medium, *Electromagn.,* **3**: 111–130, 1983.

20. J. S. Bagby and D. P. Nyquist, Dyadic Green's functions for integrated electronic and optical circuits, *IEEE Trans. Microw. Theor. Tech.,* **35**: 206–210, 1987.
21. L. Vegni, R. Cicchetti, and P. Capcec, Spectral Dyadic Green's function formulation for planar integrated structures, *IEEE Trans. Antennas Propag.,* **36**: 1057–1065, 1988.
22. P. Bernardi and R. Cicchetti, Dyadic Green's functions for conductor-backed layered structures excited by arbitrary tridimensional sources, *IEEE Trans. Microw. Theor. Tech.,* **42**: 1474–1483, 1994.
23. Z. Xiang and Y. Lu, Electromagnetic dyadic Green's function in cylindrically multilayered media, *IEEE Trans. Microw. Theor. Tech.,* **44**: 614–621, 1996.
24. L. W. Li et al., Electromagnetic dyadic Green's function in spherically multilayered media, *IEEE Trans. Microw. Theor. Tech.,* **42**: 2302–2309, 1994.
25. C. T. Tai, The dyadic Green's function for a moving isotropic medium, *IEEE Trans. Antennas Propag.,* **13**: 322–323, 1965.
26. C. Stubenrauch and C. T. Tai, Dyadic Green's function for cylindrical waveguide with moving medium, *Appl. Sci.,* **25**: 281–289, 1971.
27. C. T. Tai, *Generalized Vector and Dyadic Analysis,* 2nd ed., New York: IEEE Press, 1997.
28. J. Van Bladel, Some remarks on Green's dyadic for infinite space, *IEEE Trans. Antennas Propag.,* **9**: 563–566, 1961.
29. K. M. Chen, A simple physical picture of tensor Green's function in source region, *Proc. IEEE,* **65**: 1202–1204, 1977.
30. A. W. Johnson, A. Q. Howard, and D. G. Dudley, On the irrotational component of the electric Green's dyadic, *Radio Sci.,* **14**, 961–967, 1979.
31. A. D. Yaghjian, Electric dyadic Green's functions in the source region, *Proc. IEEE,* **68**: 248–263, 1980.
32. S. W. Lee et al., Singularity in Green's function and its numerical evaluation, *IEEE Trans. Antennas Propag.,* **22**: 311–317, 1980.
33. W. C. Chew, Some observations on the spatial and eigenfunction representations of dyadic Green's function, *IEEE Trans. Antennas Propag.,* **37**: 1322–1327, 1989.

JIAN-MING JIN
WENG CHO CHEW
University of Illinois at
Urbana–Champaign

GROUND EFFECTS, RADIOWAVE PROPAGATION.

See RADIOWAVE PROPAGATION GROUND EFFECTS.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Hadamard Transforms
Standard Article
Jon W. Mark[1] and
M. Barazande-Pour[2]
[1]University of Waterloo, Waterloo, ON, Canada
[2]University of Waterloo, Waterloo, ON, Canada
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2422
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (386K)

Abstract

The sections in this article are

Hadamard Transforms

The Modified Hadamard-Structured DCT (MHDCT)

Applications of Hadamard Matrices

Acknowledgments

| | |

Copyright © 1999-2008 All Rights Reserved.

HADAMARD TRANSFORMS

Arithmetic operations in $GF(2)$ [$GF(2)$ is the Galois or finite field with two elements: 0 and 1] are computationally simpler than in the real domain. This is one of the salient features of the Hadamard transform. The entries of the Hadamard matrices are ± 1, which are related to the (0, 1) field elements in $GF(2)$ by a simple linear transformation. The computational simplicity makes the Hadamard transform suitable for such applications as error correction coding, image compression and processing, signal representation, and others.

We begin with a discussion of the properties of the Hadamard transform and the construction of the associated Hadamard matrices. The key to applying the Hadamard transform to practical problems is the identification of the basis functions and algorithms that perform fast implementations. This article discusses the ramifications of fast implementation algorithms and their application in error correction coding, image compression and processing, and signal representation.

HADAMARD TRANSFORMS

The first work on the Hadamard transform was done by Sylvester, (1), who in 1867 proposed a recursive method for the construction of Hadamard matrices of order $N = 2^k$, for $k = 0, 1, 2, \ldots$. Later, in 1898, Scarpis proved the existence of Hadamard matrices of order $p + 1$ for $p = 3 \pmod 4$ and $p + 3$ for $p = 1 \pmod 4$. In 1893, Hadamard proved that the bound on the determinant of matrices of order N ($M^N N^{N/2}$, where $A = [a_{i,j}]$ and $|a_{i,j}| \leq M$ for all $1 \leq i, j \leq N$) which is met only for Hadamard matrices (2,3,4,5).

The Hadamard Matrix

For any real square, $N \times N$, nonsingular matrix $A = [a_{ij}]$, the Hadamard inequality states that

$$\det A \leq \left[\prod_{i=1}^{N}\left(\sum_{j=1}^{N} a_{ij}^2\right)\right]^{1/2}$$

A matrix H that satisfies the bound with equality is called a *Hadamard matrix.* Multiplying a row or a column by -1 does not destroy the Hadamard property. Similarly, permuting rows and columns does not destroy this property. A matrix H in which the elements of the first row and column are all $+1$'s is called a normalized *Hadamard matrix.*

Definition 1. The *Hadamard matrix,* $H_N = [h_{ij}]_{N\times N}$, of order N is defined as an $N \times N$ square matrix in which (1) all elements are ± 1, and (2) $\sum_k h_{ik}h_{jk} = 0$, $i \neq j$, that is, any two distinct rows are orthogonal. This condition requires that the order N be at least even for a Hadamard matrix to exist.

Existence of the Hadamard Matrix. Except for the all $+1$'s row (or all -1's row), any row of an N-dimensional Hadamard matrix must have exactly $N/2$ $+1$'s and $N/2$ -1's. Moreover, orthogonality requires that, for any two distinct rows, i and j, not involving the all $+1$'s (or -1's) row, there must be $N/4$ columns, where elements of rows i and j are both $+1$'s, both -1's, $+1$'s and -1's, and -1's and $+1$'s. Therefore a Hadamard matrix cannot exist for $N > 2$ which is not a multiple of 4.

Theorem 1. The order of any Hadamard matrix greater than 2 must be divisible by 4.

Proof: Suppose H_N is a normalized Hadamard matrix of order N, $N \geq 3$, and consider the first 3 rows of this matrix. Rows 2 through N must have an equal number of $+1$'s and -1's. Permute the columns of H_N so that the first $N/2$ elements of the second row are $+1$'s and the remaining $N/2$ elements are -1's. Suppose there are α elements of $+1$ in the first $N/2$ elements of the third row. Permute the columns so that the first α elements of the third row are $+1$'s and the elements in the third row in column $N/2 + 1$ to $N/2 + (N - \alpha/2)/2$ are $+1$'s. For the second and third rows to be orthogonal, it requires that

$$2\left[\alpha - \left(\frac{N}{2} - \alpha\right)\right] = 0$$

which implies that $N = 4\alpha$. It follows that the order of any Hadamard matrix of order greater than 2 is divisible by 4.

It remains an open question as to a whether Hadamard matrix of order $4N$, N any positive number, exists. There is no method for constructing Hadamard matrices of order $4N$ for all integer N. In the next subsection we present several methods for the construction of Hadamard matrices with the order of special sequences.

Construction of the Hadamard Matrix

There are a number of approaches to constructing Hadamard matrices. These can be divided into two general categories: those which are based on the construction of S matrices and those which are constructed directly.

The Hadamard matrix, H_N, is said to be in a normal form if the first row and column of the matrix contain only $+1$'s. Aside from the trivial cases of $N = 1$, 2, these conditions can be satisfied only if $N > 2$ is an integer multiple of 4.

Construction Methods Using *S* Matrices

Definition 2. An S matrix of order $N - 1$ is derived from a Hadamard matrix of order N by deleting the first row and column of the Hadamard matrix (row and column with only $+1$ elements) and replacing $+1$'s by 0's and -1's by 1's. An S matrix is called cyclic if each row is obtained by cyclically shifting the previous row one place to the left.

The Quadratic Residue Construction Method. Let $a_1, a_2, \ldots, a_{N-1}$ be the remainders of the numbers 1, 4, 9, . . ., $((N - 1)/2)^2$, N odd and >2, divided by N. The a_i's are called *quadratic residues* modulo N. Then the first row of the S matrix

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

is constructed as

$$s_0, s_1, \ldots, s_{N-1}$$

where

$$s_0, s_{\alpha_1}, s_{\alpha_2}, \ldots, s_{\alpha_{(N-1)/2}} \quad \text{are} \quad 1's$$

and all other s_j's are 0's. All other rows of S are constructed by cyclically shifting the previous row by one place to the left. This construction produces S matrices of order $N = 4m + 3$, where $m = 0, 1, 2, 3, \ldots$.

Maximal Length Shift Register Construction Method. In this construction the first row of the S matrix is selected to be a maximal length shift register sequence of length $N = 2^m - 1$, where $m = 1, 2, 3, \ldots$. Construction of maximal length sequences up to degree $m = 20$ can be found in Refs. 6 and 7.

The Twin Prime Construction Method. Let p and $q = p + 2$ be two prime numbers. Define two functions $f(i)$ and $g(i)$ as

$$f(i) = \begin{cases} +1 & \text{if the remainder of } i/p \text{ is a quadratic residue modulo } p \\ 0 & \text{if } p \text{ divides } i \\ -1 & \text{otherwise} \end{cases}$$

and

$$g(i) = \begin{cases} +1 & \text{if the remainder of } i/q \text{ is a quadratic residue modulo } q \\ 0 & \text{if } q \text{ divides } i \\ -1 & \text{otherwise} \end{cases}$$

There are $k = (p - 1)(q - 1)/2$ numbers a_i, $i = 1, \ldots, k$ and $1 \le a_i \le pq - 1$ for which $f(i) = g(i)$. Also, let

$$a_{k+j} = (j - 1)q \quad \text{for} \quad 1 < j < p$$

Then the first row of the S_N matrix would be

$$s_0, s_1, s_2, \ldots, s_{pq-1}$$

where

$$s_{a_i} = 0 \quad \text{for} \quad 1 \le i \le k + p$$

and

$$s_j = 1 \quad \text{for other values of j}$$

Sylvester-Hadamard Matrix. The Sylvester-type Hadamard matrices are constructed recursively as

$$H_1 = [1]$$

and

$$H_{2^n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix} \tag{1}$$

$$= H_2 \otimes H_{2^n-1} \tag{2}$$

where $\otimes$ is the Kronecker product of matrices.

If H_{N_1} and H_{N_2} are Hadamard matrices of orders N_1 and N_2 respectively, then the Kronecker product $H_{N_1} \otimes H_{N_2}$ is easily shown to be a Hadamard matrix of order N_1N_2.

Definition 3. The Kronecker product of two matrices $A_{m\times n}$ and $B_{p\times q}$ is defined as the matrix $C_{mp\times nq} = A_{m\times n} \otimes B_{p\times q}$, and is given by

$$C = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & & & \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix} \tag{3}$$

The Kronecker product is also referred to as the direct product or tensor product of matrices.

Since the permutation of the rows and columns of a Hadamard matrix does not affect the definition in Eq. (2), the Hadamard matrices H^1 and H^2 are said to be equivalent if

$$H^2 = P_r^T H^1 P_c \tag{4}$$

where P_r and P_c are permutation matrices for rows and columns, respectively.

It can be shown that the Sylvester-type Hadamard matrices are equivalent to the Hadamard matrices obtained from S matrices through maximal length shift register sequences. The (i, j)th element of the Sylvester-type Hadamard matrix can be obtained as $(-1)^{i,j}$, where i,j is the number of 1's that the binary representation of i and j have in common.

Basis Functions

The application of an orthogonal transformation depends on the basis functions and the algorithms for implementing the transformation. For example, the discrete Fourier transform is used for frequency domain analysis and filtering operation (8), the discrete cosine transform for data compression (9), the slant transformation for image coding (10), and the Hadamard and Haar transform for dyadic-invariant signal processing (11,12). The inner product of the input signal with the basis functions of the transform represents a measure of similarity between the input signal and its corresponding basis function. Figure 1 shows the basis function of the Hadamard transform of length $N = 16$.

Walsh Functions

Walsh functions are rectangular waveforms orthonormal on the interval [0, 1). They form a complete orthonormal set over this interval and can be expressed as products of Rademacher functions (13). Figure 2 shows the orthogonal waveforms of a Walsh function of order $N = 8$. Uniform sampling of the Walsh functions results in the Hadamard (or Walsh–Hadamard) matrices of corresponding order (14).

Higher-Dimensional Hadamard Matrices

Reference 15 proposes a method for construction of higher-dimensional Hadamard matrices. An N-dimensional Hadamard matrix $[h_{ijk\ldots n}]$ is a matrix with elements ± 1 such that

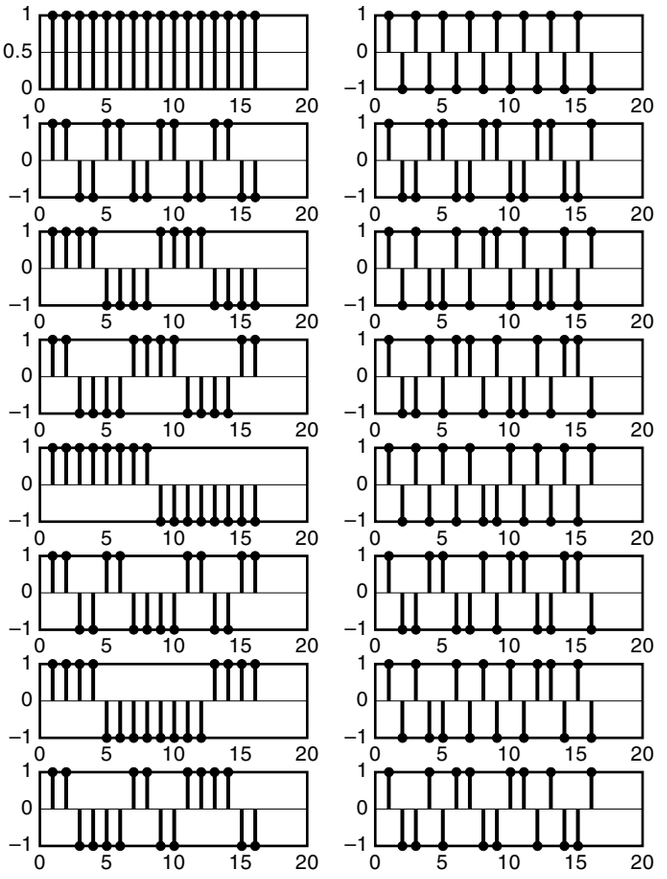


Figure 1. Hadamard basis functions ($N = 16$).

$$\sum_p \sum_q \sum_r \cdots \sum_y h_{pqr\ldots ya} h_{pqr\ldots yb} = m^{(N-1)} \delta_{ab}$$

where m is the order of the matrix. A 3-dimensional Hadamard matrix is one in which all parallel 2-dimensional layers in all normal axes are orthogonal. An N-dimensional Hadamard matrix is called proper if any 2-dimensional layer of the matrix is a 2-dimensional Hadamard matrix. These higher dimensional Hadamard matrices may find applications in error correction codes where their hierarchy of orthogonalities permit a variety of checking procedures. They might also be used in security codes based on their similarity to random binary matrices.

There is no general theory for the construction of high-dimensional Hadamard matrices of any order as there is for 2-dimensional Hadamard matrices. But 3-dimensional matrices of order 2^t can be generated from $t - 1$ successive direct (Kronecker) products of 3-dimensional Hadamard matrices of order 2.

Hadamard Transformation

Consider an N-dimensional source column vector $\boldsymbol{X}$. The Hadamard transform (HT) $\boldsymbol{Y}$ is given by

$$\boldsymbol{Y} = H_N \boldsymbol{X}$$

where H_N is the Hadamard matrix of order $N = 2^n$. Since $H_N H_N^T = NI$, where I is an $N \times N$ identify matrix,

$$H_N^{-1} = \frac{1}{N} H_N^T = H_N$$

where the superscript T denotes the matrix transpose. From definition, the determinant $|H_N|$ of the Hadamard matrix is $|H_N| = \pm N^{N/2}$. By rearranging the rows of the Hadamard matrix, we can obtain the Walsh (or sequency ordered) matrix, the Hadamard (or naturally ordered) matrix, and the Paley (or dyadically ordered) matrix. In the Walsh matrix, the rows of the Hadamard matrix are ordered according to their sequencies. The kth row of the Hadamard matrix is the jth row of the Walsh matrix, where j is the Gray code to binary conversion of k after it has been bit reversed. By premultiplying the naturally ordered Hadamard matrix with the bit-reversed order matrix, we obtain the Paley (or dyadically ordered) transform matrix.

Two-Dimensional Hadamard Transform. The 2-dimensional Hadamard transform (HT) of an array $[X(m, n)]$ of size $N \times N$ is defined as

$$[Y(u, v)] = [H_N(u, v)][X(m, n)][H_N(u, v)] \tag{5}$$

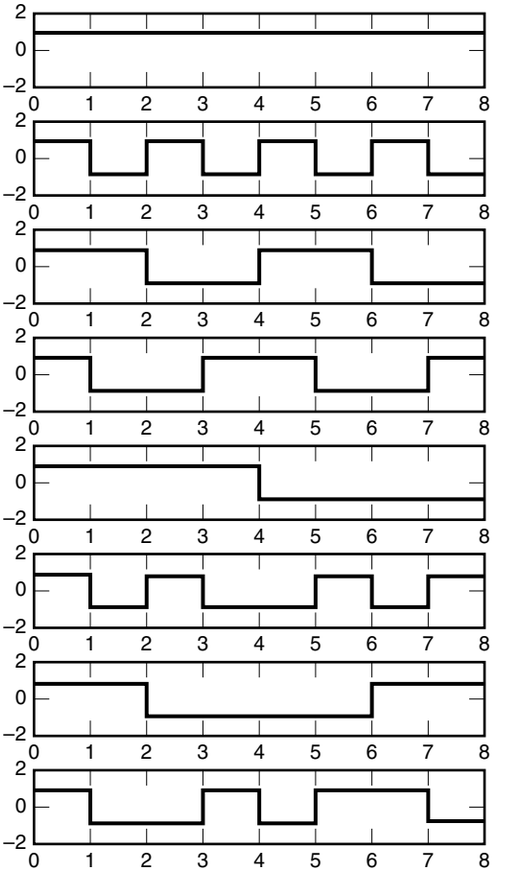


Figure 2. Walsh functions used to construct Hadamard basis ($N = 8$).

where $H_N = [H_N(u, v)]$ is a Hadamard transform of order $N = 2^n$. Pre- and postmultiplying the transformed array $[Y(u, v)]$ by the Hadamard matrix $[H_N(u, v)]$ gives

$$[X(m,n)] = \frac{1}{N^2}[H_N(u,v)][Y(u,v)][H_N(u,v)]$$

The coefficients of the 2-dimensional Hadamard transform of a matrix can be considered as the orthogonal projection of the matrix onto a set of 2-dimensional basis functions. Each transformed coefficient represents a measure of similarity between the input matrix and the corresponding basis function. The 2-dimensional basis functions of the Hadamard transform are shown in Fig. 3.

Fast Algorithms and Implementation

In Ref. 16 an efficient method is proposed to calculate the discrete Fourier transform (DFT) and Walsh–Hadamard transform (WHT) of a vector, one from the other, using a transformation matrix T between Fourier and Hadamard transformations. The method is based on the factorization of the transform matrix T into sparse matrices.

The Least Mean Square (LMS) algorithm for calculation of the forward and inverse orthogonal transformations is described in Ref. 17. Although this method requires twice as much computation as simply using the transform matrix directly, it is useful for parallel computation applications and for VLSI implementations (18).

Reference 19 proposes an algorithm of a simple systolic array processor for the Hadamard transform. It is based on the Hadamard Coefficient Generator (HCG). The HCG makes the signs of the Hadamard matrix elements required to execute the matrix multiplication.

By using the recursive structure of the Sylvester-type Hadamard matrices, efficient algorithms can be developed for Hadamard transformation (20). In general it can easily be shown that the number of 2-point Hadamard transforms required to implement an N-point Hadamard transform is $N/2 \log_2 N$, which is equivalent to $N \log_2 N$ multiply/add operations. By writing

$$\begin{aligned} H_N &= H_2 \otimes H_{N/2} = H_2 \otimes H_2 \otimes H_{N/4} \\ &= H_4 \otimes H_{N/4} \end{aligned}$$

an algorithm to compute an H_4 with only seven multiply/add operations was described in (20). The number of multiply/add operations for computing an N-point Hadamard transform can be reduced to $\frac{7}{8}N \log_2 N$, when N is an even power of 2, and $\frac{7}{8}N \log_2 N + N/8$, when N is an odd power of 2.

In Ref. 21 a fast algorithm was developed for the sequency-ordered form of the Hadamard transform. The machine architecture obtained for this algorithm is similar to the one derived for the machine-oriented algorithm of the fast Fourier transform (FFT) by Corinthios (22), except that the multipliers are deleted, the add/subtract operator sequencing varies from one iteration to another, and the ideal shuffling is performed before each iteration. The algorithms for the Hadamard transform in its natural order and dyadic order were also derived. It was shown that all three algorithms can be performed with a single machine structure by including a simple binary controller.

The 2-dimensional Hadamard transform, together with a selected set of basis functions and fast computational algorithms, may be used to encode 2-dimensional images. As shown later, the Hadamard matrix with ± 1 entries lacks dynamic range for good image representation at the decoder. On the other hand, the discrete cosine transform (DCT) with cosinusoidal entries provides greater dynamic range and hence better image representation at the decoder. However, the computational complexity of the DCT is very high compared to the Hadamard transform. Nevertheless, the salient features of the HT and DCT may be combined to yield an acceptable performance-complexity tradeoff, as described in the next section.

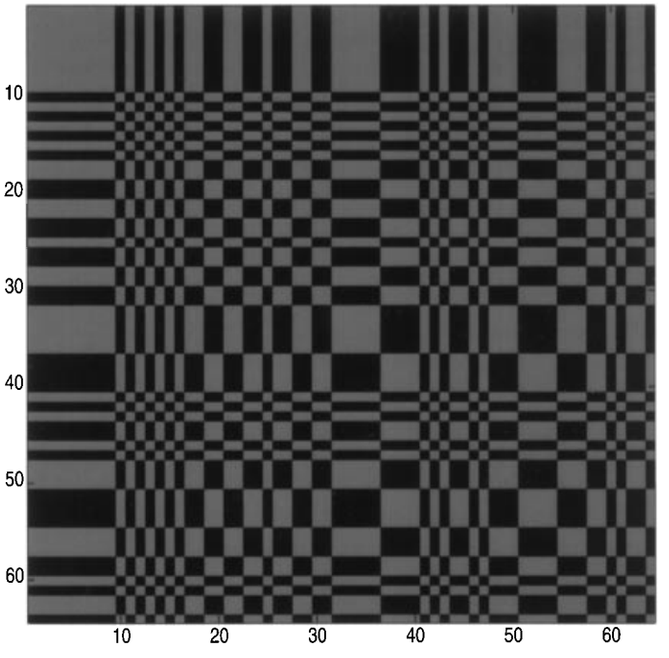


Figure 3. Two-dimensional Hadamard basis functions ($N = 8$).

THE MODIFIED HADAMARD-STRUCTURED DCT (MHDCT)

The salient features of a good image scheme are: (1) good reproduction quality, (2) high compression ratio, and (3) fast computation. The DCT, which forms an integral part of the JPEG and MPEG standards, satisfies the first two features, but is computationally complex. The entries of the DCT matrix are cosine functions, so that all arithmetic operations, such as multiplications and additions, have to be performed in the real domain. On the other hand, the entries of the Hadamard matrix are ± 1, so that arithmetic operations are much simpler. The excellent performance of DCT is attributable to the fact that the transform coefficients are uncorrelated in DCT, but not in HT. Thus, the use of HT for image coding does not yield sufficiently good representation quality. As a performance-complexity tradeoff, it may be feasible to combine the good features of DCT and HT. This was the approach taken in Ref. 23 in the construction of the Hadamard-structured DCT (HDCT). The effectiveness of HDCT as an image coding scheme depends to a large degree on the choice of basis functions.

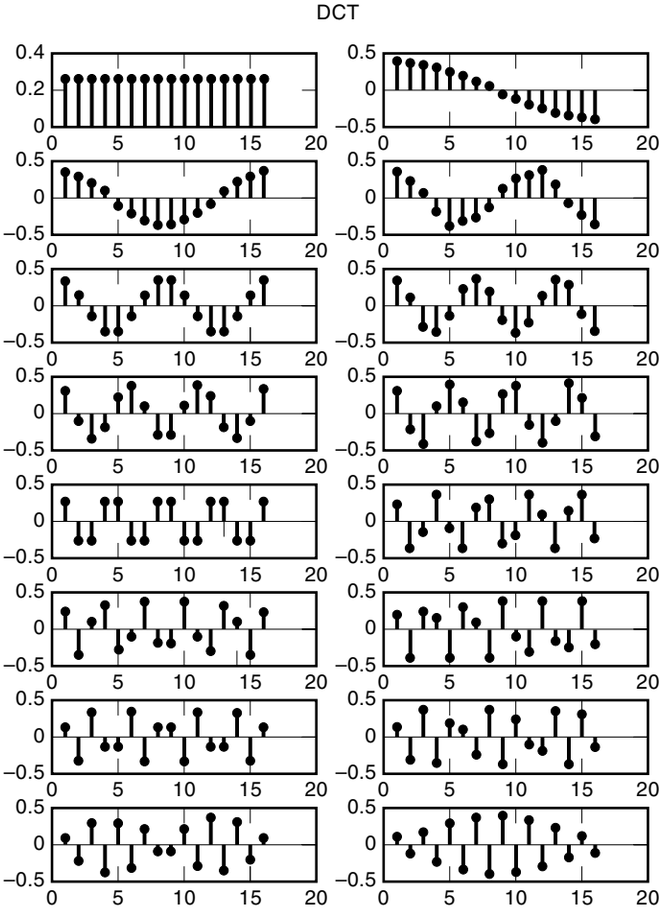


Figure 4. DCT basis vectors ($N = 16$).

Kou and Mark (23) have proposed an HDCT for speech coding. The basis functions used in Ref. 23 lack symmetry to be efficient for image coding. The modified HDCT (MHDCT) presented in Ref. 24 has basis vectors with symmetric and antisymmetric properties suitable for image coding. Examples of the basis vectors for $N = 16$ for DCT and MHDCT are shown in Figs. 4 and 5, respectively. It is noted that the basis vectors of DCT and MHDCT exhibit even and odd symmetry.

The MHDCT transformation matrix of order N is defined by the following recursive structure:

$$T_1 = [1]$$

$$T_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

and

$$T_{2^k} = \frac{1}{\sqrt{2}}\begin{bmatrix} T_{2^{k-1}} & T_{2^{k-1}} \\ C_{2^{k-1}} & -C_{2^{k-1}} \end{bmatrix} \quad \text{for} \quad k = 2, 3, 4, \ldots \tag{6}$$

where C_{2^k} is $2^k \times 2^k$ normalized DCT matrix with mnth entry given by

$$c_{mn} = \eta_n \cos\frac{(2m+1)\,n\pi}{2N} \quad \text{for} \quad m, n = 0, 1, \ldots, N-1 \tag{7}$$

where

$$\eta_n = \begin{cases} \dfrac{1}{\sqrt{N}}, & n = 0 \\ \sqrt{\dfrac{2}{N}}, & n \neq 0 \end{cases}$$

Thus the MHDCT matrix has a mixture of ± 1 and cosine entries as opposed to all cosine entries in DCT.

The sequency of a basis vector is defined as the number of sign changes. Figures 4 and 5 show the basis vectors of DCT and MHDCT, respectively. It is observed that the basis functions of MHDCT take all possible sequencies from 0 to 15. Like the DCT, the MHDCT is an orthogonal transformation, that is, $T_N^{-1} = T_N^t$. Hence we have the transform pair between a source vector $\boldsymbol{X}$ and the transform vector $\boldsymbol{Y}$:

$$\boldsymbol{Y} = T_N \boldsymbol{X} \tag{8}$$

and

$$\boldsymbol{X} = T_N^t \boldsymbol{Y} \tag{9}$$

The MHDCT has the property that the lower order transformation matrix can be obtained from the higher order ones using the following theorem.

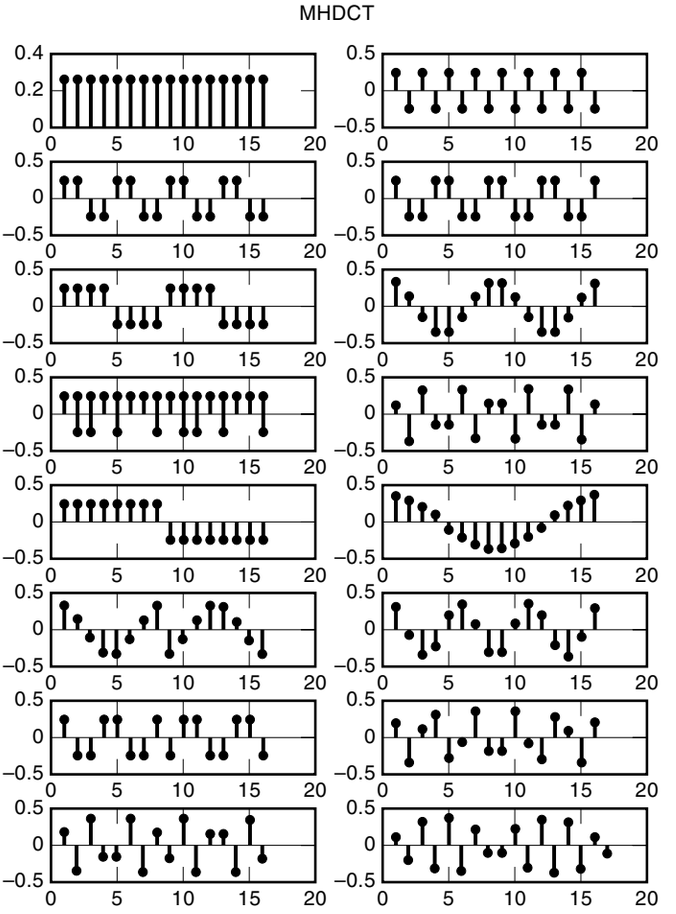


Figure 5. MHDCT basis vectors ($N = 16$).

Theorem 2. *For $k \geq 3$, T_{2^k} can be expressed as*

$$T_{2^k} = \frac{1}{\sqrt{2^k}}\left[C_2 \oplus \sum_{i=1}^{k-1} C_{2^i}\right]\left[\prod_{i=1}^{k-1}\left((H_2 \otimes I_{2^{k-i}}) \oplus I_{2^k - 2^{k-i+1}}\right)\right] \tag{10}$$

where $\otimes$ is a Kronecker product operation and $\sum$ or $\oplus$ is a direct sum operation.

Proof. Let H^2 be a 2×2 Hadamard matrix. From the definition of MHDCT and the recursion, Eq. (6), for $k \geq 2$ we have

$$\begin{aligned} T_{2^k} &= \frac{1}{\sqrt{2}}\begin{bmatrix} T_{2^{k-1}} & T_{2^{k-1}} \\ C_{2^{k-1}} & -C_{2^{k-1}} \end{bmatrix} \\ &= \frac{1}{\sqrt{2}}\begin{bmatrix} T_{2^{k-1}} & 0 \\ 0 & C_{2^{k-1}} \end{bmatrix} \cdot \begin{bmatrix} I_{2^{k-1}} & I_{2^{k-1}} \\ I_{2^{k-1}} & -I_{2^{k-1}} \end{bmatrix} \\ &= \frac{1}{\sqrt{2}}\left[T_{2^{k-1}} \oplus C_{2^{k-1}}\right]\left[H_2 \otimes I_{2^{k-1}}\right] \end{aligned} \tag{11}$$

$T_{2^{k-1}}$ in the first term on the right-hand side of Eq. (11) can further be expanded as

$$T_{2^{k-1}} = \frac{1}{\sqrt{2}}\left[\left(T_{2^{k-2}} \oplus C_{2^{k-2}}\right)\left(H_2 \otimes I_{2^{k-2}}\right)\right] \tag{12}$$

Substituting Eq. (12) in Eq. (11) yields

$$T_{2^k} = \frac{1}{\sqrt{2}}\left[\left(T_{2^{k-2}} \oplus C_{2^{k-2}}\right)\left(H_2 \otimes I_{2^{k-2}}\right) \oplus C_{2^{k-1}}\right]\left[H_2 \otimes I_{2^{k-1}}\right] \tag{13}$$

By using the matrix operation (25),

$$AD \oplus B = (A \oplus B)(D \oplus I) \tag{14}$$

in Eq. (13), we get

$$\begin{aligned} T_{2^k} = \frac{1}{\sqrt{2^2}}&\left[T_{2^{k-2}} \oplus C_{2^{k-2}} \oplus C_{2^{k-1}}\right]\left[H_2 \otimes I_{2^{k-2}} \oplus I_{2^k - 2^{k-1}}\right] \\ &\cdot \left[H_2 \otimes I_{2^{k-1}}\right] \end{aligned} \tag{15}$$

As in Eq. (12), we can expand $T_{2^{k-2}}$ as

$$T_{2^{k-2}} = \frac{1}{\sqrt{2}}\left[T_{2^{k-3}} \oplus C_{2^{k-3}}\right]\left[H_2 \otimes I_{2^{k-3}}\right] \tag{16}$$

Substituting $T_{2^{k-2}}$ from Eq. (16) into Eq. (15) yields

$$\begin{aligned} T_{2^k} = \frac{1}{\sqrt{2^3}}&\left[\left(T_{2^{k-3}} \oplus C_{2^{k-3}}\right)\left(H_2 \otimes I_{2^{k-3}}\right) \oplus C_{2^{k-2}} \oplus C_{2^{k-1}}\right] \\ &\left[H_2 \otimes I_{2^{k-2}} \oplus I_{2^k - 2^{k-1}}\right] \cdot \left[H_2 \otimes I_{2^{k-1}}\right] \end{aligned} \tag{17}$$

Again by using the matrix operation Eq. (14) and Eq. (17) we get

$$\begin{aligned} T_{2^k} = \frac{1}{\sqrt{2^3}}&\left[T_{2^{k-3}} \oplus C_{2^{k-3}} \oplus C_{2^{k-2}} \oplus C_{2^{k-1}}\right] \\ &\cdot \left[H_2 \otimes I_{2^{k-3}} \oplus I_{2^k - 2^{k-2}}\right)\left[H_2 \otimes I_{2^{k-2}} \oplus I_{2^k - 2^{k-1}}\right] \\ &\cdot \left[H_2 \otimes I_{2^{k-1}}\right] \end{aligned} \tag{18}$$

As we continue this iterative procedure and recognize that $T_2 = C_2$, Theorem 2 follows.

Fast MHDCT Algorithm

Theorem 2 also offers a way to implement the MHDCT. In this algorithm, the input signal is hierarchically Hadamard transformed and then the result is DCT transformed using different sizes. The structure of this algorithm for $N = 8$ is given in Fig. 6. It consists of a 2-stage Hadamard structured transform followed by a windowed discrete-cosine transform.

Complexity of the MHDCT

As with the transform methods, the complexity of the MHDCT is defined as the number of multiplications and additions (or subtraction) required to implement the transformation. Let $A_{DCT}(2^i)$ and $M_{DCT}(2^i)$ denote the number of additions and multiplications of a 2^i-point DCT transform, respectively. Then by using Theorem 2, the number of additions and multiplications for a 2^k-dimensional fast MHDCT, $A_{MHDCT}(2^k)$ and $M_{MHDCT}(2^k)$, will be

$$A_{MHDCT}(2^k) = \sum_{i=1}^{k-1} 2 \times 2^{k-1} + A_{DCT}(2) + \sum_{j=1}^{k-1} A_{DCT}(2^j) \tag{19}$$

and

$$M_{MHDCT}(2^k) = M_{DCT}(2) + \sum_{j=1}^{k-1} M_{DCT}(2^j) \tag{20}$$

The total number of multiplications and additions depends on the DCT algorithm used in the MHDCT implementation. Tables 1 and 2 compare the complexity of MHDCT with DCT using the algorithms of Chen et al. (26) and Lee (27).

The results in Tables 1 and 2 show clearly the computational saving of the MHDCT over the DCT. The number of additions and multiplications required to implement MHDCT are remarkably less than those required for DCT. However, this computational saving is only one factor, and we shall compare the performance of MHDCT with those of other transformations.

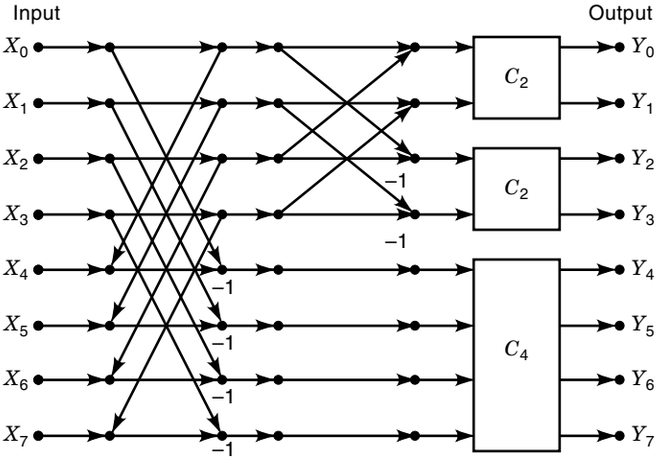


Figure 6. A block diagram for fast computation of MHDCT for $N = 8$.

Table 1. Comparison of Complexity Between DCT and MHDCT (Using Chen's Algorithm)

	Additions		Multiplications	
N	DCT	MHDCT	DCT	MHDCT
8	26	24	16	10
16	74	66	44	26
32	194	172	116	70
64	482	430	292	186
128	1154	1040	708	478
256	2690	2450	1668	1186
512	6146	5652	3844	2854
1024	13826	12822	8708	6698

In the following subsections, different measures are used to evaluate the performance of different transformations and use them to compare the performance of MHDCT with HT and DCT.

Decorrelation Efficiency and Coding Performance

In the previous section we showed that the MHDCT is computationally simpler than the DCT. Now we will compare the performance of MHDCT with those of HT and DCT in terms of decorrelation efficiency and transformation gain. The decorrelation efficiency provides a basis for comparing different orthogonal transforms against each other. With regard to transformation gain, pulse code modulation (PCM) is used as a benchmark for comparing the performance of different coding techniques.

Decorrelation Efficiency. This section presents simulation results of the decorrelation efficiency η_c of the DCT, HT, and MHDCT transforms.

Let R_X and R_Y be the correlation matrix of the source and transformed processes, respectively. Let

$$\lambda_x = \sum_{i=0}^{N-1} \sum_{\substack{j=1 \\ i \neq j}}^{N-1} |R_X(i, j)|$$

and

$$\lambda_y = \sum_{i=0}^{N-1} \sum_{\substack{j=1 \\ i \neq j}}^{N-1} |R_Y(i, j)|$$

Table 2. Comparison of Complexity Between DCT and MHDCT (Using Lee's Algorithm)

	Additions		Multiplications	
N	DCT	MHDCT	DCT	MHDCT
8	29	25	12	8
16	81	70	32	20
32	209	183	80	52
64	513	456	192	132
128	1217	1097	448	324
256	2817	2570	1024	772
512	6401	5899	2304	1796
1024	14337	13324	5120	4100

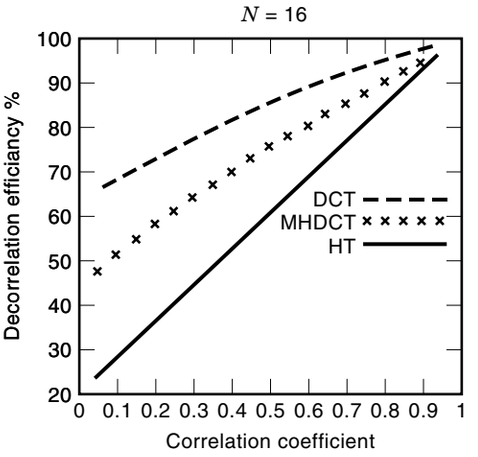


Figure 7. Comparison of the decorrelation efficiency of the HT, MHDCT, and DCT ($N = 16$).

The decorrelation efficiency is defined as

$$\eta_c = 1 - \frac{\lambda_y}{\lambda_x} \tag{21}$$

For completely decorrelated spectral coefficients, $\lambda_y = 0$ and $\eta_c = 1$.

Figures 7 and 8 show the decorrelation efficiency, η_c, of different transforms as a function of the correlation coefficient of the source process, with the transform size as a parameter. It is observed that the performance of MHDCT is between that of HT and DCT, and by increasing the transformation size, the performance of MHDCT approaches that of DCT.

Transformation Gain. The transformation gain G_{TC} of a transform coding (TC) system is defined as the ratio of the measured reconstruction error of PCM to that of the transform coding at the same information bit rate. On the assump-

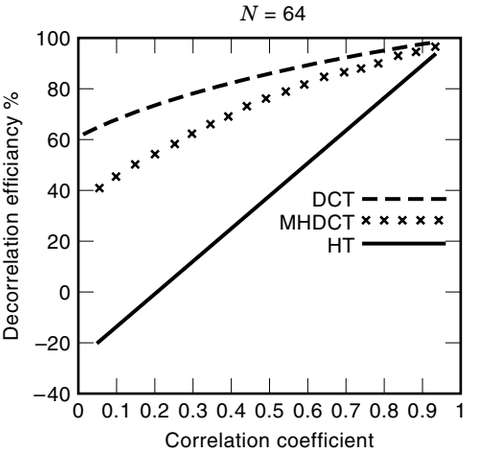


Figure 8. Comparison of the decorrelation efficiency of the HT, MHDCT, and DCT ($N = 64$).

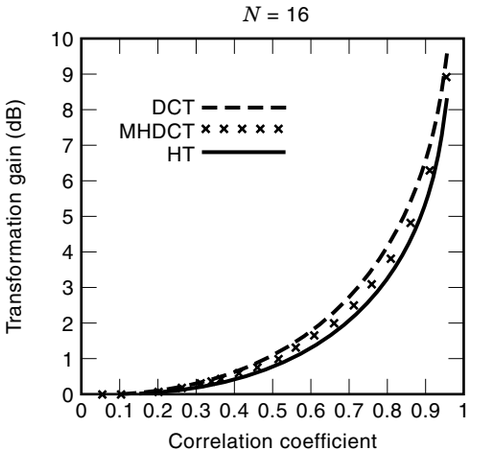


Figure 9. Comparison of the transformation gain of different transforms ($N = 16$).

tion that the error processes have zero mean, G_{TC} is given by

$$G_{TC} = \frac{\sigma_e^2(PCM)}{\sigma_e^2(TC)} = \frac{\frac{1}{N}\sum_{j=0}^{N-1}\sigma_j^2}{\left(\prod_{j=0}^{N-1}\sigma_j^2\right)^{1/N}} \tag{22}$$

Figures 9 and 10 present the transformation gain of DCT, MHDCT, and HT transform codes as a function of the correlation coefficient, for two different transformation size N.

It is observed that the transformation gain of the MHDCT is quite close to that of DCT, and it increases as the transformation size or the correlation coefficient increases.

APPLICATIONS OF HADAMARD MATRICES

Error Correction Coding

Due to the unwanted effects of noise, distortion, and interference, the output of a storage medium or a digital communication channel differs from its input. The theory of error correction coding is concerned with the protection of digital signals against the errors that occur during transmission or storage. In this section a special group of error correcting codes, called Hadamard codes, are discussed. A Hadamard code of rate N/k, where $k = \log_2(2N)$ has a minimum distance of $N/2$. This code is capable of correcting $d_{\min}/2 = N/4$ random errors.

Hadamard Codes. An error correcting code with $2N$ codewords can be constructed from a Hadamard matrix of order N as follows:

1. Change all $+1$'s to 0's and all -1's to 1's.
2. Select each row and its complement of the Hadamard matrix as a codeword.

$$H_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

The complement of H_4 is

$$\overline{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

The rows of H_4 and $\overline{H}_4$ form a linear binary code of block length $N = 4$ having $2N = 8$ codewords. The code is also called a first order Reed-Muller error correcting code, an implementation of which is shown in Fig. 11. The minimum dis-

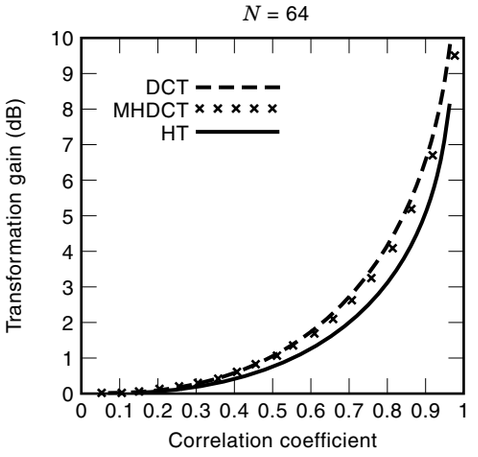


Figure 10. Comparison of the transformation gain of different transforms ($N = 64$).

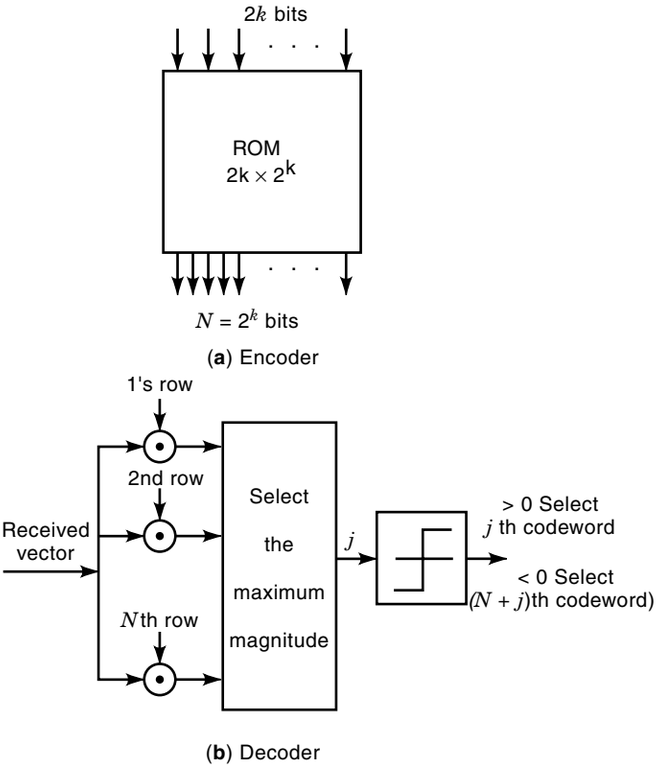


Figure 11. The encoder and decoder of a first-order Reed–Muller code.

tance of the code is

$$d_{\min} = N/2 = 2$$

In general a Hadamard code of size $2N$ codewords is obtained by selecting the N rows and their complement. By selecting $M = 2^k \leq 2N$ of these codewords, we obtain a Hadamard code $H(N, k)$, where each codeword conveys k information bits. The resulting code has constant weight equal to $N/2$ and minimum distance $d_{\min} = N/2$.

Using the above procedure, one can construct a Hadamard error correcting code (n, k, d) with codeword length $N = 2^m$, input block length $k = \log_2 2N = m + 1$, and Hamming distance $d = N/2 = 2^{m-1}$, where m is a positive integer.

The Hadamard error correcting codes with $N = 2^m$ are called *linear*. The *nonlinear* Hadamard codes are those with order $n = p + 1$ for a multiple of 4, and any order $n = p^m + 1$ if the quadratic residues in $GF(p^m)$ are used. They are also called Paley-type Hadamard codes.

Decoding of Hadamard Codes. Hadamard codes are decoded using the following procedure:

1. First, in a transform matrix of order N change all 0's to $+1$'s and all 1's to -1's.
2. Premultiply this transform matrix by the received vector and locate the largest magnitude coefficient of the transformed vector. Assume it is the jth coefficient.
3. If the largest coefficient is positive, decide on the jth codeword; otherwise decide on the $(j + N)$th codeword.

Logical Hadamard Transform and Nonlinear Block Codes. The logical Hadamard transform proposed by Searle (28) is a modification of the arithmetic Hadamard transform for binary inputs. In the logical Hadamard transform, both input and output blocks are binary. The procedure for obtaining the logical Hadamard transform is to take the output of an arithmetic Hadamard transform and threshold each element at zero. The only condition for recovering the input signal is that the first element of the input vector should be 1. Banta (29) has used the logical Hadamard transform to obtain a nonlinear block code with block length $N = 2^n - 1$, data length $K = 2^r - 1$, $r < n - 1$, error correction $t = 2^{n-r} - 1$, and rate $R = K/N \approx 1/t$.

A series of explicit low-rate binary linear block codes which have relatively low covering radius and can be rapidly recoded is described in Ref. 30. These codes can be derived from higher dimensional analogues of the Gale–Berlekamp switching game.

MHDCT as an Image Coding Scheme

The MHDCT can be used to transform 2-dimensional image signals. The image array is divided into blocks of size $N \times N$. Each block is then transformed using the MHDCT and the transform coefficients are adaptively quantized and sent to the receiver (24).

Two-Dimensional MHDCT Transform. Let T_N be the one-dimensional MHDCT transformation matrix that operates independently on the rows and the columns of the 2-dimensional $N \times N$ image block $[X(m, n)]$. Then the 2-dimensional MHDCT transform $[Y(u, v)]$ of the block image $[X(m, n)]$ is defined as

$$[Y(u, v)] = [T_N(u, m)][X(m, n)][T_N(v, n)]^t \tag{23}$$

where T_N is the one-dimensional MHDCT transform defined by Eq. (6). By the orthogonality property of the MHDCT, the inverse transform can be derived as

$$[X(m, n)] = [T_N(u, m)]^t[Y(u, v)][T_N(v, n)] \tag{24}$$

The 2-dimensional transformation of the images can be considered as an orthogonal projection of the image onto the set of basis pictures. The input image can be reconstructed by a linear combination of the basis pictures, with coefficients being the 2-dimensional transform coefficients. The basis pictures of MHDCT and DCT for $N = 8$ are shown in Figs. 12 and 13, respectively. The efficiency of a transformation for encoding a particular image depends on the shape of both the image and the basis pictures. The basis pictures should be able to represent different patterns of pixel intensities within the image.

For a 2-dimensional image, the N^2 values of $X(m, m)$ are the elements of a subimage of size $N \times N$. In image coding, the typical arrays used are of sizes $N = 4$, 8, 16, or 32. This partitioning into subimages is particularly efficient in cases where the correlations are localized to the neighboring pixels, and where the structural details tend to cluster. Partitioning of an image into subimages reduces the complexity of the transformation. The coding method in Ref. 24 uses 8×8 blocks. This block size yields a good tradeoff between complexity and performance of the transformation. By using the Kolmogorov-Smirnov (KS) test (31), the distribution of the ac coefficients of the MHDCT was found to be Laplacian.

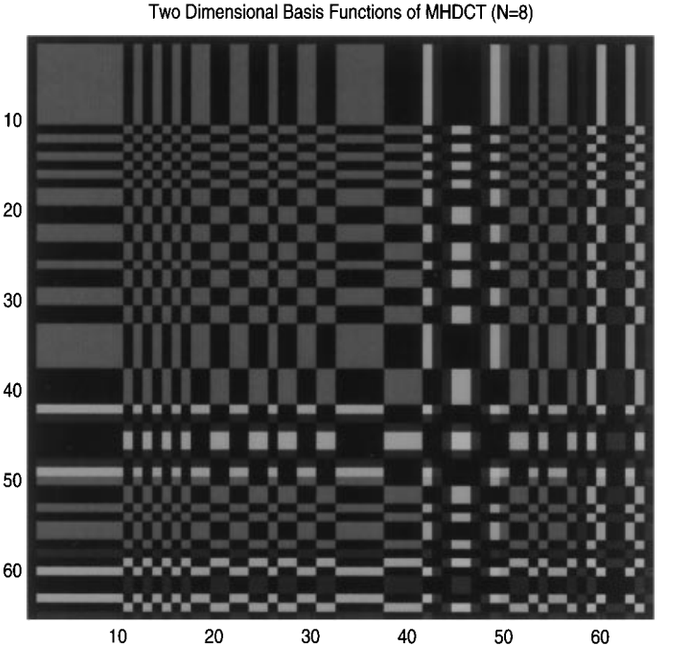


Figure 12. Two-dimensional MHDCT basis pictures ($N = 8$).

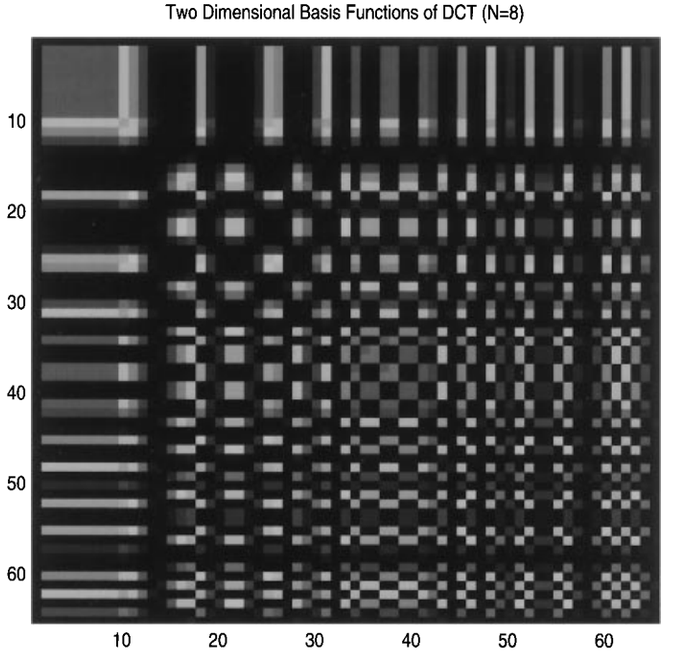


Figure 13. Two-dimensional DCT basis pictures ($N = 8$).

Adaptive MHDCT Transform Coding of Images. The transform coefficients of 8×8 blocks of the images are quantized and transmitted to the receiver through the communication channel. To make efficient use of the available bandwidth with minimum distortion, an adaptive method as in Ref. 32 can be used. The blocks of the image in the transform domain are classified according to their ac energy level. To demonstrate the effectiveness of the coding scheme, choose four classes of blocks, and choose the decision boundaries for the classification such that the number of blocks in each class is the same. The coding of the image is performed on a block-by-block basis. Then the process is made adaptive by assigning more bits to the higher energy blocks. Also, within a block a larger number of bits is allocated to the coefficients in the block with higher variance. The sum of the squared values of ac coefficients in each block of the image in the transform domain is defined as the energy level of the block. The classification map, bit allocation matrices, and the MHDCT coefficients are transmitted to the receiver through the communication channel. In the receiver, image reconstruction is accomplished by inverting the compression operation. Figure 14 shows a block diagram of the adaptive MHDCT coder.

In practice, it is necessary to make two passes on the image data. The first pass generates the subblock classification map and also assigns the bit allocation matrices to different classes. The second pass quantizes the subblock transform coefficients using the bit allocation matrices. We have used the optimal Lloyd-Max (33,34) quantizers designed for Laplacian sources in our coding system. The quantizer can also be designed using the Lloyd-Max algorithm for a suitable training data.

Bit Allocation. A crucial part of transform coding is an efficient bit allocation algorithm that provides the possibility of quantizing some transform coefficients more finely than others. Minimization of the mean-squared reconstruction error can be used as the criterion to derive an optimum bit allocation algorithm. In our case, the bit allocation matrix for each block is constructed after determining the variances of the transform coefficients, as given by

$$N_{B_k}(u,v) = \frac{1}{2}\log_2[\sigma_k^2(u,v)] - \log_2[D] \qquad \forall (u,v) \neq (0,0) \tag{25}$$

where $\sigma_k^2(u, v)$ is the variance of the transform sample and D is a parameter. The value of D is first initialized and then recursively calculated to meet the desired total number of bits.

Experimental Results for Adaptive Encoding of Images. We have used the adaptive MHDCT coding method to compress the 512×512 Lena image with intensity value uniformly quantized to 256 levels (8 bits per pixel). The results of our experiments are summarized in Table 3. The peak signal-to-noise ratio defined in Eq. (26) is used for objective comparison of images.

$$SNR = 10\log_{10}\left(\frac{255^2}{\frac{1}{N^2}\sum_{m=0}^{m=N-1}\sum_{n=0}^{n=N-1}[X(m,n) - \hat{X}(m,n)]^2}\right) \tag{26}$$

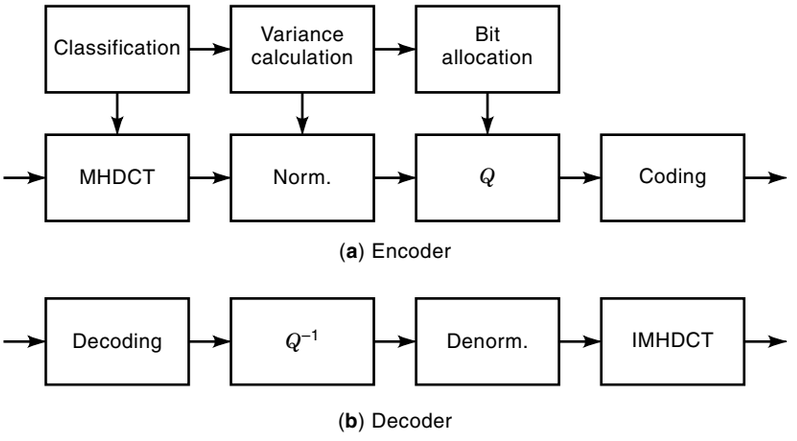


Figure 14. Adaptive MHDCT coding system.

Table 3. Comparison of SNR for Lena Image Using Different Transforms

Bit Rate (bpp)	Hadamard SNR	MHDCT SNR	DCT SNR	Compression Ratio
0.25	28.29	29.08	30.91	32.0
0.50	29.67	30.44	31.77	16.0
0.75	30.81	32.05	32.75	10.6
1	31.91	33.41	33.68	8.0

where the image size is $N \times N$ and $X(m, n)$ and $\hat{X}(m, n)$ are the original and the reconstructed images, respectively.

It is shown that the performance of MHDCT is better than HT and close to that of DCT with less complexity. Figures 15 and 16 provide a visual comparison between the performances of DCT and MHDCT in adaptive coding of the Lena image. The difference in quality of the two pictures is not noticeable. From this figure it is observed that the performance of MHDCT is quite close to the performance of DCT and that the difference in the SNR is very small. No entropy coding has been used in our experiments and using a lossless entropy code will significantly improve the performance of the coding system.

Signal Representation

The Hadamard matrix may be used to design orthogonal and biorthogonal M-ary sequences. To form the signal set, we might use the Hadamard matrix construction. The Hadamard matrix of order 4 is

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

These four rows plus their complements form an 8-ary biorthogonal set of linear binary code of block length $N = 4$ having $2N = 8$ codewords. The minimum distance is $d_{\min} = N/2 = 2$. The selected row can be sent as a rectangular pulse train of duration

$$T_s = T_b \log_2 M = 3T_b \quad (M = 8)$$

where T_b is the bit duration.

The Hadamard matrix of order 8 is constructed as

$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix} = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix}$$

The eight rows can be used as the signal patterns for the 8-ary orthogonal set. The minimum distance of the code is $d_{\min} = N/2$. The first element of each row is a $+1$, which means that this signal element yields no distinguishing feature to the signal set. Therefore this signal element can be dropped with no loss in performance to lower the entropy per bit to $\frac{7}{8}$ of the former value, while maintaining $d_{\min}$ fixed and thus achieving the same error probability. Although the rows of the Hadamard matrices are mutually orthogonal, for spectral purposes, these are not good for random binary sequences.



Figure 15. MHDCT coding of Lena image, compression ratio = 8.10.



Figure 16. DCT coding of Lena image, compression ratio = 8.10.

Feature Extractions and Pattern Recognition. Features such as shape, motion, pressure details and timing, and transformation methods such as Fourier and Hadamard have been used in handwritten signature recognition with various degrees of success. In Ref. 35 a fast Fourier transform is used to transform normalized signatures into the frequency domain. Fifteen harmonics having the largest magnitude normalized by their corresponding variances were selected and used in a stepwise discriminant analysis.

An approach to the problem of signature verification is described in Ref. 36. This paper considers the signature as a 2-dimensional image and uses the Hadamard transformation as a means of data reduction and feature extraction. The signature image is a 2-dimensional array of 1's and 0's corresponding to light and dark areas on the original image. This method achieves 91% of correct recognition, 11% valid signature rejection, and 41% forgery acceptance.

For handwriter identification, feature extraction was performed by decomposition of the quantized pressure pattern into a set of orthogonal functions. In view of the rectangular nature of the time domain waveforms, Hadamard transform is a logical natural choice (37).

In Ref. 38 the Hadamard transform was used to design a vector classifier for a Predictive Classified Vector Quantizer (PCVQ). The performance of Hadamard transform vector classifier was compared to a spatial vector classifier. The good performance of the Hadamard transform classifier is the unique property of the Hadamard transform, which groups the frequency components within the image vector into distinct coefficients.

The Hadamard transform is used in Ref. 39 to represent image signals in the transformed domain. Compared to the Fourier transform, the Hadamard transform offers an order of magnitude speed increase. Transmitting the Hadamard transform coefficients of an image instead of the spatial representation of the image provides a potential tolerance to channel errors and the possibility of reduced bandwidth requirement. Linear and Gaussian-optimized quantizers are used to quantize the Hadamard transform coefficients. Results with the linear quantizer are poor because of the large quantization errors at high sequences (equivalent to frequencies in Fourier transform).

The coding efficiency of the differential PCM (DPCM) with a 2-dimensional predictor is compared to that of a 2-dimensional Hadamard transform code (HTC) in intrafield coding of the NTSC composite signal in Ref. 40. It is shown that the coding efficiency of the HTC is far lower than that of the DPCM in the case of a signal having high-power level carrier chrominance signal, such as a color-bar signal. In general it was shown that

1. For signals with large values of horizontal and vertical correlation ratios (close to 1), DPCM outperforms HTC, while for smaller values of correlation ratio, the performance of HTC is much better.
2. In the case of high compression ratio (2 bit/pel), HTC shows higher coding efficiency than DPCM.

Special Purpose HT Applications

Spread Spectrum. The basic idea in spread spectrum is to distribute a relatively low-dimensional data signal into a higher dimensional signal. A jammer with finite energy has to either distribute its energy on all dimensions, thereby inducing a small interference on each dimension or put its total energy on a small subspace leaving the remainder of the space interference free. In the time domain, the distribution of the signal is achieved by multiplying the data signal by a member of an orthogonal set.

Orthogonal sequences can be used as spreading signals in spread-spectrum multiple-access systems. They have zero correlation when they are time synchronized. But in some applications, like multipath fading environments, multiple delays introduce nonzero cross correlation between the otherwise orthogonal signals. One solution to this can be concatenation of a (pseudo-noise) PN sequence with the orthogonal coding to increase the randomness of the orthogonal sequences.

Orthogonal coding was used to spread the information signal in Ref. 41. Each signal is coded with the same orthogonal or biorthogonal code, followed by a modulo-2 addition of a unique signature sequence. With block orthogonal coding, $\log_2 N$ information bits are encoded into an N-bit codeword. An N-bit signature sequence is then modulo-2 added to the codeword before transmission. Thus, orthogonal coding provides the spreading of the information signal, not the signature sequence. From the coding point of view, each signal is assigned a code set, or coset, which is formed by modulo-2 adding the signature sequence to each of N (orthogonal) or $2N$ (bi-orthogonal) codes. Thus the system employs a supercode consisting of codes of orthogonal codes.

A wideband, direct-sequence, code-division multiple access (CDMA) was proposed in Ref. 42. The wideband CDMA system uses PN and Walsh–Hadamard codes for spreading the signal in order to achieve the minimal interference between traffic and control (pilot, sync, and paging) channels. The spreading is done by a combination code, which is generated by PN and orthogonal codes from Walsh–Hadamard sequences to minimize mutual interference between traffic and control signals.

Reference 43 proposes an optimal set of signature sequences for use in a CDMA system where orthogonal or biorthogonal Walsh—Hadamard coding is used to spread the signal. This paper shows that in the special case of a synchronous system with no multipath echoes and use of WH code as the spreading sequence, the product of any two different signature sequences should be a bent sequence of length $N = 2^n$. A sequence with a constant magnitude WH transform is called a bent sequence.

Filter Design in the Hadamard Transform Domain. Adaptive filters have many applications in interference cancellation, linear prediction, spectral estimation, system modeling, and channel equalization in communication systems. The filter parameters can be computed in the time or transform domain. Because of some computational efficiencies observed in the transform domain (44), this subsection discusses the application of Hadamard transform for filter design.

Reference 45 proposes a fast implementation of the LMS error adaptive transversal filter. The fast Walsh–Hadamard transform (FWHT) technique is adopted in this implementation. The error vector is obtained by subtracting the WH transform of the desired output and the filter output. The input vector is also WH transformed before entering the filter. Finally, the output of the filter is inverse WH transformed to obtain the representation in the time domain. This filter provides a significant reduction in computation over both the conventional time domain and the frequency domain adaptive filters. For data blocks of size of N, the proposed filter only requires $2N$ adaptations compared to those of $2N^2$ and $2N + 3N/2 \log_2 N$ for time domain and FFT filters, respectively.

A block implementation of 2-dimensional finite-impulse response (FIR) digital filters using the matrix decomposition approach is described in Ref. 46. The coefficient matrix of the

block realization is decomposed via the Walsh–Hadamard transform without involving any intermediate calculations.

The application of the recursive Walsh–Hadamard transformation to FIR and infinite impulse response (IIR) filtering was investigated in Ref. 47. It was shown that by using a common recursive transform, the usual frequency domain FIR filtering problem was converted into a Walsh sequence-domain filtering problem. A hardware implementation of the filter was also proposed.

Equalizers. Equalizers are used to mitigate the effect of intersymbol interference (ISI) in transmission of digital signals through band-limited communication channels. Different algorithms in the time domain, including the symbol rate linear transversal filter equalizer and the fractionally spaced equilizer (FSE), are proposed for equalizer design. To achieve rapid convergence of the equalizer coefficients, the equalizers are designed in the frequency domain.

Reference 48 considers adaptive equalization for digital data transmitted over discrete linear channels exhibiting intersymbol interference in addition to additive noise. LMS equalization is developed in the discrete sequence (Walsh or Hadamard) domain using a gradient projection method. An adaptive LMS adaptation algorithm in the Hadamard domain is developed, in which the input data sequence is divided into blocks. Each block is Hadamard transformed, passed through an LMS equalizer, and then converted into the time domain again. The performance of time domain and Hadamard transformed domain are comparable, but the latter provides a much faster convergence.

A technique for implementing an echo canceller for full-duplex data transmission was presented in Ref. 49. This article considers the effect of nonlinear distortion in the echo path or in the echo replica. The Hadamard transform was used to add or delete some taps in the equalizer design.

Spectroscopy. Spectroscopy is a branch of physics that studies the production and measurement of the spectra. Conventional spectrometers sort the electromagnetic radiation into rays of different wavelengths and measure the intensity of each ray separately. Hadamard transform optics is a technique in spectroscopy that measures the spectrum of a beam of light using multiplexing. The basic idea is that instead of measuring the intensity of each wavelength separately, the spectral components are multiplexed and the total intensity of each group is measured. This reduces the measurement noise and results in a more accurate measurement of the spectra. Hadamard transform is used for the multiplexing. The same technique can be used for imagers in reconstructing an image or picture.

The basic Hadamard transform instrument consists of an optical separator, an encoding mask, a detector, and a processor. The separator may be a lens that produces a focused image at the mask or a prism that spears different frequency components of the beam and focuses them at different locations on the mask. Different parts of the mask pass the light to the detector, or absorb it or refect it towards a reference detector. If we record the difference between readings of the main detector and the reference detector, the intensity of this element of the beam is multiplied by $+1$, 0, or -1, respectively. Sometimes masks are only made up of two types of elements, open and closed slots that pass or obstruct the light when the reference detector is removed. The best mask for minimizing the measurement error is the Hadamard mask for the first configuration and the S-matrix mask for the second one (50).

Encryption. Hadamard transform was used in Ref. 51 to encrypt analog speech signals. In the analog speech encryption, speech samples are first converted into a transform domain like DCT, DFT, or discrete Hadamard transform (DHT). The encryption is achieved by permuting the transform coefficients. The encrypted transform samples are then converted back into the time domain and transmitted. The application of the analog speech encryption is in both narrowband and wideband systems (speech transmission over a bandlimited telephone channel and speech storage and retrieval). As a comparison for using different transforms, the DCT, DFT, and (Discrete Prolate Spherical transform) DPST can be used in narrowband systems. The KLT (Karhunen–Loeve transform) and DHT are more suitable for wideband systems. Based on subjective and objective measures (such as LPC, cepstral, SNR distance measures), DCT turned out to be the best transform with respect to both residual intelligibility of the encrypted speech and the recovered speech quality. The DFT produced results that are inferior to the DCT. The DCT implementation would also offer speed advantages over FFT.

ACKNOWLEDGMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant No. A7779.

BIBLIOGRAPHY

1. J. Sylvester, Thoughts on inverse orthogonal matrices, simultaneous sign-succesions, and tesselated properties in two or more colors with application to Newton's rule, ornamental tile-work and the theory of numbers, *Philos. Magazine,* **Series 4**: 461–475, 1967.
2. J. Sylvester, *Mathematical Recreation and Essays,* New York: Macmillan, 1947, pp. 108–111.
3. J. Hadamard, Resolution d'une question relative aux determinants, *Bull. Sci. Math. (2),* **17**: I, 240–246, 1893.
4. M. Vitterli, Tree structure for orthogonal transforms and application to the Hadamard transform, *Sig. Proc.,* **5**: 473–484, 1983.
5. S. G. Wilson and M. Lakshman, Autocorrelation and power spectrum of Hadamard signalling, *Proc. IEEE,* **135** (3): 258–261, 1968.
6. E. R. Berlekamp, *Algebric Coding Theory,* New York: McGraw-Hill, 1968.
7. F. J. MacWillimas and N. J. A. Sloane, *The Theory of Error Correcting Codes,* Amsterdam, The Netherlands: North-Holland, 1977.
8. A. V. Oppenheim and R. W. Schaffer, *Digital Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1975.
9. N. S. Jayant and P. Knoll, *Digital Coding of Waveforms,* Englewood Cliffs, NJ: Prentice-Hall, 1984.
10. W. K. Pratt, W. H. Chen, and L. R. Welch, Slant transform image coding, *IEEE Trans. Commun.,* **22**: 1075–1093, 1974.
11. K. G. Beauchamp, *Applications of Walsh and Related Functions,* New York: Academic Press, 1984.

12. J. Pearl, Optimal dyadic methods of time-invariant systems, *IEEE Trans. Comput.,* **24**: 598–603, 1975.
13. D. F. Elliot and K. R. Rao, *Fast Transforms; Algorithms, Analysis, Applications,* New York: Academic Press, 1982, p. 301.
14. J. L. Walsh, A closed set of orthogonal functions, *Amer. J. Math.,* **55**: 5–24, 1923.
15. P. J. Shlichta, Higher dimensional Hadamard matrices, *IEEE Trans. Inf. Theory,* **25**: 566–572, 1979.
16. S. Boussakta and A. Holt, Fast algorithms for calculation of both Walsh–Hadamard and Fourier transforms, *Electron. Lett.,* **25**: 1352–1354, 1989.
17. S. S. Wang, LMS algorithm and discrete orthogonal transforms, *IEEE Trans. Circuits Syst.,* **38**: 949–951, 1991.
18. B. Widrow et al., Fundamental relations between the LMS algorithm and the DFT, *IEEE Trans. Circuits Syst.,* **34**: 614–820, 1987.
19. M. H. Lee and Y. Yasuda, Simple systolic array algorithm for Hadamard transform, *Electron. Lett.,* **26**: 1478–1480, 1990.
20. D. Coppersmith, E. Feig, and E. Linzer, Hadamard transforms on multiply/add transforms, *IEEE Trans. Sig. Process.,* **42**: 969–970, 1994.
21. Y. A. Geaddah and M. J. G. Corinthios, Natural dyadic and sequency-order algorithms and processors for the Walsh–Hadamard transform, *IEEE Trans. Comput.,* **C-26**: 435–442, 1977.
22. M. J. Corinthios, A time-series analyzer, *Proc. Symp. Comput. Process. Commun.,* April 8–10, 1969, pp. 47–60.
23. W. Kou and J. W. Mark, A new look at DCT transform, *IEEE Trans. Acoust. Speech Signal Process.,* **37**: 1899–1907, 1989.
24. M. Barazande-Pour and J. W. Mark, Adaptive MHDCT coding of images, *Proc. 1994 1st Int. Conf. Image Process.,* 1994, pp. 90–94.
25. R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis,* New York: Cambridge University Press, 1991.
26. W. H. Chen, C. Smith, and S. C. Fralick, A fast computational algorithm for the discrete cosine transform, *IEEE Trans. Commun.,* **COM-25**: 1004–1009, 1977.
27. B. G. Lee, A new algorithm to compute the discrete cosine transform, *IEEE Trans. Acoust. Speech Signal Process.,* **ASSP-32**: 1243–1245, 1984.
28. N. H. Searle, A logical Walsh-Fourier transform, *n Applications of Walsh functions, 1970 Proc.-Symp. Workshop,* Naval Research Laboratory, Washington, DC, 1970, pp. 95–98.
29. E. D. Banta, A class of nonlinear block codes using the logical Hadamard transform to achieve virtually identical encoding and decoding, *IEEE Trans. Inf. Theory,* **24**: 761–763, 1978.
30. J. Pach and J. Spencer, Explicit codes with low covering radius, *IEEE Trans. Inf. Theory,* **34** (5): 1281–1285, 1988.
31. S. D. Silvey, *Statistical Inference,* London: Chapman Hall, 1975.
32. Wen-Hsiun Chen and C. H. Smith, Adaptive coding of monochrome and color images, *IEEE Trans. Commun.,* **COM-25**: 1285–1292, 1977.
33. J. Max, Quantizing for minimum distortion, *IEEE Trans. Inf. Theory,* **6**: 7–12, 1960.
34. S. P. Lloyd, Least squares quantization in PCM, *IEEE Trans. Inf. Theory,* **IT-28**: 127–135, 1982.
35. C. F. Lam and D. Kamins, Signature recognition through spectral analysis, *Pattern Recognition,* **22**: 39–44, 1989.
36. W. F. Nemeck and W. C. Lin, Experimental investigation of automatic signature verification, *IEEE Trans. Syst. Man Cybern.,* **4**: 121–126, 1974.
37. K. P. Zimmerman and M. J. Varady, Handwriter identification from one bit quantized pressure patterns, *Pattern Recognition,* **18**: 63–72, 1985.
38. K. N. Ngan and H. C. Koh, Predictive classified vector quantization, *IEEE Trans. Image Process.,* **1**: 269–280, 1992.
39. W. K. Pratt, J. Kane, and H. C. Andrews, Hadamard transform image coding, *Proc. IEEE,* **57**: 58–68, 1969.
40. H. Murakaami, Y. Yatori, and H. Yamamoto, Comparison between DPCM and Hadamard transform coding in the composite coding of the NTSC color TV signal, *IEEE Trans. Commun.,* **30**: 469–479, 1982.
41. G. E. Bottomley, Signature sequence selection in a CDMA system with orthogonal coding, *IEEE Trans. Veh. Technol.,* **42**: 62–68, 1993.
42. F. Atsushi et al., Wideband CDMA system for personal communication systems, *IEEE Commun. Mag.,* **34**: 116–123, 1996.
43. P. K. Enge and D. V. Sarwate, Spread spectrum multiple access performance of orthogonal codes: linear receivers, *IEEE Trans. Commun.,* **COM-35**: 1309–1318, 1987.
44. M. Dentino, J. McCool, and B. Widrow, Adaptive filtering in the frequency domain, *Proc. IEEE,* **66**: 1658–1659, 1978.
45. R. N. Boules, Adaptive filtering using the fast Walsh–Hadamard transformation, *IEEE Trans. Electromagn. Compat.,* **31**: 125–128, 1989.
46. B. Mertzios and A. Venetsanopoulos, Fast block implementation of 2-dimensional FIR digital filters via the Walsh–Hadamard decomposition, *Int. J. Electron.,* **68**: 991–1004, 1990.
47. G. Peceli and B. Feher, Digital filters based on recursive Walsh–Hadamard transformation, *IEEE Trans. Circuits Syst.,* **37**: 150–152, 1990.
48. M. Maqusi and O. Natour, Adaptive equalization in the discrete-time discrete frequency and Hadamard domains, *Int. J. Electron.,* **72**: 197–212, 1992.
49. O. Agazzi, D. G. Messerschmitt, and D. A. Hodges, Nonlinear echo cancellation of data signals, *IEEE Trans. Commun.,* **30**: 2421–2433, 1982.
50. M. Harwit and N. J. A. Sloane, *Hadamard Transform Optics,* New York: Academic Press, 1979.
51. S. Sridharan, E. Dawson, and B. Goldburg, Speech encryption in the transform domain, *Electron. Lett.,* **26**: 655–657, 1990.

JON W. MARK
M. BARAZANDE-POUR
University of Waterloo

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Hankel Transforms
Standard Article
M. Rahman[1]
[1]Daimler-Chrysler Corporation
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2423
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (361K)

Abstract

The sections in this article are

The Hankel Transform

Some Elementary Properties of Hankel Transforms

The Hankel Transforms of Derivatives of a Function

Relation Between Fourier and Hankel Transforms

Parseval's Relation For Hankel Transforms

The Hankel Operator

The Erdelyi–Kober Operators of Fractional Integration

Beltrami-Type Relations

Dual Integral Equations Involving Hankel Transforms

Triple Integral Equations Involving Hankel Transforms

Quadruple Integral Equations Involving Hankel Transforms

Miscellaneous

Compendium of Basic Formulas

Suggested Further Reading

Acknowledgments

| | |

Copyright © 1999-2008 All Rights Reserved.

HANKEL TRANSFORMS

The object of this article is to introduce integral transform of a particular type, called the *Hankel transform,* and to illustrate the *use* of this method by means of examples. The treatment is that of a review article and as such is not meant to be exhaustive, its aim being to give a concatenated account of known results rather than present new ones. The emphasis throughout is on those results that are of frequent occurrence in boundary-value problems of mathematical physics, but some indication is also given for possible theoretical investigations.

Proofs are either omitted entirely or only the key steps are outlined. Readers interested in rigorous proofs of some of the statements in this article are referred to the books by Sneddon (1,2), Davies (3), Andrews and Shivamoggi (4), and Zayed (5).

The organization of the article is as follows: In the first section, we illustrate the motivation behind introducing the Hankel transform and then give a precise definition of the Hankel transform and its inversion. The next two sections are devoted to the derivation of some basic properties of Hankel transforms. In the following section, we explore the connection between Fourier and Hankel transforms. Parseval's relation for Hankel transforms is then deduced. We next introduce the modified operator of Hankel transforms. An overview of Erdelyi–Kober operators and their generalization by Sneddon and Cooke is given. We then derive Beltrami-type relations and give a brief account of their generalization by Sneddon. An extensive account is given of the applications of Erdelyi–Kober and Cooke operators to dual, triple, and quadruple integral equations involving Hankel transforms. A number of issues that arise in connection with applications of Hankel transforms to many physical problems is then addressed. For the convenience of the readers, a compendium is given in the last section of the basic theorems and formulas of Hankel transforms that are of frequent occurrence in applications.

THE HANKEL TRANSFORM

The Hankel transform arises naturally as a result of using the method of separation of variables to boundary value problems of mathematical physics in cylindrical coordinates, for example, boundary-value problems for the Laplace and Helmholtz equations involving half-spaces and regions bounded by parallel planes. In general, application of this technique is relevant to problems leading to the integration of equations of the type

$$\frac{\partial^2\phi}{\partial r^2}+\frac{1}{r}\frac{\partial\phi}{\partial r}-\frac{v^2}{r^2}\phi+L\phi=f(r,\ldots)$$

where L is a linear operator that does not contain r, and $f(r, \ldots)$ is a prescribed function.

To illustrate this, let us consider the axisymmetric solution $\phi(r, z)$ of Laplace's equation:

$$\frac{\partial^2\phi}{\partial r^2}+\frac{1}{r}\frac{\partial\phi}{\partial r}+\frac{\partial^2\phi}{\partial z^2}=0 \tag{1}$$

in the half-space $r > 0$, $z > 0$, which satisfies the boundary condition

$$\phi(r,0)=f(r) \tag{2}$$

where $f(r)$ is a prescribed function of r.

In addition, the solution of the problem must satisfy the regularity conditions so that the field decays as $R \to \infty$, where $R=\sqrt{r^2+z^2}$.

Assuming that the solution can be represented in the separated-variable form,

$$\phi(r,z)=\phi_1(r)\phi_2(z)$$

we find that Eq. (1) reduces to

$$\frac{1}{\phi_1}\frac{d^2\phi_1}{dr^2}+\frac{1}{\phi_1 r}\frac{d\phi_1}{dr}=\frac{-1}{\phi_2}\frac{d^2\phi_2}{dz^2} \tag{3}$$

Since the left-hand side of Eq. (3) depends only on r while the right-hand side only on z, we conclude that they must be equal to a constant, say, $\lambda = -s^2$, where s is a real quantity. Thus, we obtain two ordinary differential equations

$$\begin{aligned}\frac{d^2\phi_1}{dr^2}+\frac{1}{r}\frac{d\phi_1}{dr}+s^2\phi_1&=0\\ \frac{d\phi_2}{dz^2}-s^2\phi_2&=0\end{aligned} \tag{4}$$

The first of these equations is that of Bessel [see Watson (6)], whose solution bounded at the origin is

$$\phi_1(r)=A_1(s)J_0(sr)$$

where $A_1(s)$ is an arbitrary function of s and $J_0(sr)$ is the zeroth-order Bessel function of the first kind.

On the other hand, the solution of the second relation of Eq. (4) ensuring a decaying field is given by

$$\phi_2(z)=A_2(s)e^{-sz}$$

Therefore, the solution of Eq. (1) is

$$\phi(r,z)=A(s)J_0(sr)e^{-sz} \tag{5}$$

where $A(s)$ is an arbitrary function of s. Readers can easily verify that the other cases, viz., $\lambda = 0$ and $\lambda = s^2$ (s is a real quantity), must be ignored, since they do not ensure a decaying field as $R \to \infty$.

The solution of Eq. (5) has the property that, if $s > 0$, $\phi(r, z) \to 0$ as $R \to \infty$. By simple superposition, we can therefore construct the solution of the form

$$\phi(r,z)=\int_0^\infty sA(s)J_0(sr)e^{-sz}ds \tag{6}$$

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

The condition of Eq. (2) will be satisfied if

$$f(r) = \int_0^\infty sA(s)J_0(sr)\,ds \tag{7}$$

yielding an equation for determining the unknown function $A(s)$. It will be shown later that $A(s)$ is given by the formula

$$A(s) = \int_0^\infty rf(r)J_0(sr)\,dr \tag{8}$$

which upon substitution into Eq. (6) then formally gives the solution of our problem.

The formulas in Eqs. (7) and (8) define a transformation pair called the *Hankel transform of order zero.* We now give a formal definition of the Hankel transform of an arbitrary order of a function.

Given a real function $f(r)$ defined in the interval $(0, \infty)$, suppose that

1. $f(r)$ is piecewise continuous and of bounded variation in every finite subinterval $[a,b]$, where $0 < a < b < \infty$
2. the integral

$$\int_0^\infty \sqrt{r}|f(r)|\,dr < \infty$$

Then, *the Hankel transform of the νth order* of the function $f(r)$ satisfying the preceding conditions is defined as

$$\tilde{f}_\nu(s) = \int_0^\infty rf(r)J_\nu(sr)\,dr \tag{9}$$

which we shall write as

$$\tilde{f}_\nu(s) = \mathscr{H}_\nu[f(r); r \to s] \tag{10}$$

Sometimes, for the sake of brevity, we shall write this notation as $\mathscr{H}\nu[f(r);s]$, $\mathscr{H}\nu[f(r)]$, or simply $\mathscr{H}\nu f(r)$.

Readers should note that since the kernel of the Hankel transform is the Bessel function, the theory of Hankel transforms relies heavily on the theory of the Bessel functions. Perhaps, for this reason, in some literature, this transform is called Bessel transformation or Fourier–Bessel transformation.

The Hankel inversion theorem states that if the function $f(r)$ satisfies the preceding conditions, then

$$\int_0^\infty s\tilde{f}_\nu(s)J_\nu(sr)\,ds = f(r) \tag{11}$$

If the function has a jump discontinuity at a point, then the right-hand side of Eq. (11) should be replaced by the sum

$$\tfrac{1}{2}[f(r+0) + f(r-0)]$$

We shall not give a proof of the Hankel inversion theorem here. Interested readers are referred to the book by Sneddon (2).

It follows from Eqs. (10) and (11) that

$$f(r) = \int_0^\infty sJ_\nu(sr)\,ds \int_0^\infty r_0 f(r_0)J_\nu(sr_0)\,dr_0 \qquad 0 < r < \infty, \qquad \nu > -\tfrac{1}{2} \tag{12}$$

Equation (12) is called *Hankel's integral theorem.*

Evidently, Eq. (11) can be written as

$$f(r) = \mathscr{H}_\nu^{-1}[\tilde{f}_\nu(s); s \to r]$$

which, in the notation of Eq. (10), is equivalent to

$$f(r) = \mathscr{H}_\nu[\tilde{f}_\nu(s); s \to r]$$

whence establishing the rule $\mathscr{H}_\nu = \mathscr{H}_\nu^{-1}$. Thus, we see that if $\nu > -\frac{1}{2}$, there is a symmetrical relationship between a function and its Hankel transform of order ν, in the sense that if $\tilde{f}\nu(s)$ is the Hankel transform of order ν of a function $f(r)$, then $f(r)$is the Hankel transform of order ν of $\tilde{f}\nu(s)$.

Extensive tables have been constructed of the Hankel direct and inverse transforms of functions usually encountered in applications [for instance, see Erdelyi et al. (7)].

As in the case of other types of integral transforms, the use of Hankel transform has many advantages, for example, it is applicable to both homogeneous and inhomogeneous problems, it simplifies calculations and singles out the purely computational part of the solution, and it allows us to construct an operational calculus for a given kernel by using tables of direct and inverse transforms.

An extensive account of applications of the Hankel transform as well as other integral transforms to problems in mathematical physics was given by Sneddon (1,2,8) and Lebedev, Skalskaya, and Ufliand (9). Perhaps, it is Sneddon who may quite justifiably be regarded as the most ardent proponent of using the method of integral transforms—in particular, Hankel transform—to various boundary-value problems of mathematical physics.

SOME ELEMENTARY PROPERTIES OF HANKEL TRANSFORMS

Property 1

$$\mathscr{H}_{-m}[f(r); r \to s] = (-1)^m\mathscr{H}_m[f(r); r \to s] \qquad (m = \pm1, \pm2, \ldots, \pm n, \ldots)$$

Proof of this property follows from the fact that [Watson (6)]

$$J_{-m}(sr) = (-1)^m J_m(sr)$$

Property 2

$$\mathscr{H}_\nu[f(ar); r \to s] = a^{-2}\mathscr{H}_\nu\left[f(r); r \to \frac{s}{a}\right]$$

Proof. By definition, we have

$$\mathscr{H}_\nu[f(ar); r \to s] = \int_0^\infty rf(ar)J_\nu(sr)\,dr \tag{13}$$

By making a change of variable $ar = \rho$, we reduce the integral in Eq. (13) to the form

$$\begin{aligned}\mathscr{H}_\nu[f(ar); r \to s] &= a^{-2}\int_0^\infty \rho f(\rho) J_\nu(sa^{-1}\rho)\, d\rho \\ &= a^{-2}\mathscr{H}_\nu\left[f(r); r \to \frac{s}{a}\right]\end{aligned}$$

Property 3

$$\mathscr{H}_\nu[r^{-1}f(r); r \to s] = \frac{s}{2\nu}[\tilde{f}_{\nu-1}(s) + \tilde{f}_{\nu+1}(s)] \qquad (\nu \neq 0)$$

Proof. From the recurrence relation for the Bessel functions [Watson (6)]

$$j_{\nu-1}(x) - \frac{2\nu}{x}J_\nu(x) + J_{\nu+1}(x) = 0$$

we deduce

$$\begin{aligned}\mathscr{H}_\nu[r^{-1}f(r); r \to s] &= \int_0^\infty f(r) J_\nu(sr)\, dr \\ &= \frac{s}{2\nu}\left(\int_0^\infty rf(r) J_{\nu-1}(sr)\, dr \right. \\ &\quad \left. + \int_0^\infty rf(r) J_{\nu+1}(sr)\, dr\right) \\ &= \frac{s}{2\nu}[\tilde{f}_{\nu-1}(s) + \tilde{f}_{\nu+1}(s)]\end{aligned}$$

Property 4 The shift formula for the Hankel transforms is

$$\mathscr{H}_n[f(r-a)H(r-a); r \to s] = \sum_{m=-\infty}^{\infty} \alpha_m \tilde{f}_m(s)$$

where

$$\begin{aligned}\alpha_m = J_{n-m}(sa) &+ \tfrac{1}{2}as[(m+1)^{-1}J_{n-m-1}(sa) \\ &+ (m-1)^{-1}J_{n-m+1}(sa)]\end{aligned}$$

Proof of this property is given in the book by Sneddon (2).

It should be mentioned here that it is not possible to obtain a simple shift formula for the Hankel transforms. This is primarily because the addition formula for the Bessel functions, that is, the Neumann–Lommel addition formula [Watson (6)]

$$J_n(x+y) = \sum_{m=-\infty}^{\infty} J_m(x) J_{n-m}(y)$$

is much more complicated than the addition formula for the exponential functions e^x and e^{ix} for the Laplace and Fourier transforms.

THE HANKEL TRANSFORMS OF DERIVATIVES OF A FUNCTION

In applications of Hankel transforms to physical problems, it is necessary to have expressions for the Hankel transforms of the derivatives of a function or a combination of them, through the Hankel transforms of the function itself. Using the definition of Hankel's transform and the formula for integrating by parts, we obtain

$$\begin{aligned}\mathscr{H}_\nu\left[\frac{df}{dr}; s\right] &= \int_0^\infty r\frac{df}{dr} J_\nu(sr)\, dr \\ &= [rf(r)J_\nu(sr)]_0^\infty - \int_0^\infty \frac{\partial}{\partial r}[rJ_\nu(sr)]f(r)\, dr\end{aligned} \tag{14}$$

The first term on the right vanishes provided that the function $f(r)$ is such that

$$\lim_{r\to 0} r^{\nu+1}f(r) = 0, \qquad \lim_{r\to\infty} \sqrt{r}f(r) = 0$$

It follows from the arguments leading to the proof of the Hankel inversion theorem [see Sneddon (2)] that the second of these conditions holds for any $f(r)$ whose Hankel transform exists. Therefore, the first term on the right in Eq. (14) vanishes if

$$f(r) = o(r^{-\nu-1}), \qquad r \to 0$$

where o is the Landau's symbol of order.

From the theory of Bessel functions [Watson (6), Erdelyi et al. (10)], we have

$$\begin{aligned}\frac{\partial}{\partial r}[rJ_\nu(sr)] &= J_\nu(sr) + rJ'_\nu(sr) \\ J'_\nu(sr) &= srJ_{\nu-1}(sr) - \nu J_\nu(sr)\end{aligned}$$

so that Eq. (14) now takes the following form:

$$\mathscr{H}_\nu\left[\frac{df}{dr}; r\right] = (\nu-1)\int_0^\infty f(r)J_\nu(sr)\, dr - s\int_0^\infty rf(r)J_{\nu-1}(sr)\, dr \tag{15}$$

However, the integral on the right is the $(\nu-1)$th-order Hankel transform of $f(r)$, that is,

$$\int_0^\infty rf(r)J_{\nu-1}(sr)\, dr = \mathscr{H}_{\nu-1}[f(r); r \to s]$$

Thus, Eq. (15) takes the form

$$\mathscr{H}_\nu\left[\frac{df}{dr}; r \to s\right] = (\nu-1)\int_0^\infty f(r)J_\nu(sr)\, dr - s\mathscr{H}_{\nu-1}[f(r); r \to s] \tag{16}$$

The first term on the right is obviously the νth-order Hankel transform of the function $r^{-1}f(r)$. However, our objective is to express everything in terms of the Hankel transform of the function $f(r)$. This can be achieved by utilizing the following relation [Erdelyi et al. (10)]:

$$J_\nu(sr) = \frac{1}{2\nu}[J_{\nu-1}(sr) + J_{\nu+1}(sr)] \tag{17}$$

Inserting Eq. (17) into Eq. (16), after some arrangements, we finally obtain the following important relationship:

$$\begin{aligned}\mathscr{H}_\nu\left[\frac{df}{dr}; r \to s\right] = &-s\frac{\nu+1}{2\nu}\mathscr{H}_{\nu-1}[f(r); r \to s] \\ &+ s\frac{\nu-1}{2\nu}\mathscr{H}_{\nu+1}[f(r); r \to s]\end{aligned} \tag{18}$$

Expressions for Hankel transforms of the higher derivatives of the function $f(r)$ may be deduced by repeated application of the formula in Eq. (18). For instance,

$$\mathscr{H}_\nu\left[\frac{d^2f}{dr^2}; r\to s\right] = \frac{s^2(\nu+1)}{4(\nu-1)}\mathscr{H}_{\nu-2}[f(r)] - \frac{s^2(\nu^2-3)}{2(\nu^2-1)}\mathscr{H}_\nu[f(r)] + \frac{s^2(\nu-1)}{4(\nu+1)}\mathscr{H}_{\nu+2}[f(r)] \tag{19}$$

In applications of Hankel transforms to many physical problems, it becomes necessary to have available the formula for Hankel transform of the differential operator:

$$\mathscr{B}_\nu = \frac{d^2}{dr^2} + \frac{1}{r}\frac{d}{dr} - \frac{\nu^2}{r^2}$$

Integrating by parts and assuming that $df/dr = o(r^{-1})$, we find

$$\int_0^\infty r\frac{d^2f}{dr^2}J_\nu(sr)\,dr = -\int_0^\infty \frac{df}{dr}\frac{d}{dr}[rJ_\nu(sr)]\,dr$$

so that

$$\int_0^\infty r\left(\frac{d^2f}{dr^2} + \frac{1}{r}\frac{df}{dr}\right)J_\nu(sr)\,dr = -s\int_0^\infty \frac{df}{dr}rJ'_\nu(sr)\,dr = s\int_0^\infty f(r)\frac{d}{dr}[rJ'_\nu(sr)]\,dr \tag{20}$$

Equation (20) was derived on the assumption that the function $rf(r)\to 0$ as $r\to 0$ or ∞.

We know from the theory of Bessel functions [Watson (6), Erdelyi et al. (10)] that the function $J_\nu(sr)$ satisfies the differential equation

$$\frac{d}{dr}[rJ'_\nu(sr)] = -\left(s^2 - \frac{\nu^2}{r^2}\right)rJ_\nu(sr) \tag{21}$$

Upon substitution of Eq. (21) into Eq. (20), we obtain the following formula:

$$\int_0^\infty r\left(\frac{d^2f}{dr^2} + \frac{1}{r}\frac{df}{dr} - \frac{\nu^2}{r^2}f\right)J_\nu(sr)\,dr = -s^2\int_0^\infty rf(r)J_\nu(sr)\,dr = -s^2\mathscr{H}_\nu[f(r); r\to s] \tag{22}$$

An immediate consequence of Eq. (22) is the formula

$$\int_0^\infty r\left(\frac{d^2f}{dr^2} + \frac{1}{r}\frac{df}{dr}\right)J_0(sr)\,dr = -s^2\mathscr{H}_0[f(r); r\to s] \tag{23}$$

To illustrate the use of the properties of Hankel transforms, let us consider the classic problem of determining the potential at any point in the field induced by an electrified disk of radius a, whose potential is raised to ϕ_0 (ϕ_0 is a constant). The problem is known as *Weber's problem*. A discussion of this problem can be found in the books by Jeans (11) and Smythe (12). The problem reduces to that of solving Laplace's equation in Eq. (1) with the boundary conditions

$$\begin{aligned}&\phi(r,0) = \phi_0, \qquad 0\le r<a\\ &\left.\frac{\partial\phi}{\partial z}\right|_{z=0} = 0, \qquad r>a\end{aligned} \tag{24}$$

The second boundary condition in Eq. (24) expresses the symmetry of the field with respect to the plane of the disk, that is, the plane $z = 0$.

To solve the problem, we use the zeroth-order Hankel transform of the function $\phi(r, z)$, that is,

$$\phi(r,z) = \mathscr{H}_0[\tilde{\phi}(s,z); s\to r] \tag{25}$$

Applying the transformation in Eq. (25) to Eq. (1) and making use of the relation of Eq. (23), we obtain the following ordinary differential equation

$$\frac{d^2\tilde{\phi}}{dz^2} - s^2\tilde{\phi} = 0$$

whose solution is

$$\tilde{\phi}(s,z) = A(s)e^{-sz} + B(s)e^{sz} \tag{26}$$

where $A(s)$ and $B(s)$ are some unknown functions of s.

Because of symmetry, it is sufficient to consider the half-space $z\ge 0$ only. Then, since the field must vanish at infinity (regularity conditions), we must set $B = 0$, so that Eq. (26) reduces to

$$\tilde{\phi}(s,z) = A(s)e^{-sz}$$

Therefore, our formal solution of the problem takes the form

$$\phi(r,z) = \mathscr{H}_0[A(s)e^{-sz}; s\to r] \tag{27}$$

Utilizing the boundary conditions in Eq. (24), we get the following equations to determine the unknown function $A(s)$:

$$\begin{aligned}&\mathscr{H}_0[A(s); s\to r] = \phi_0, \qquad 0\le r<a\\ &\mathscr{H}_0[sA(s); s\to r] = 0, \qquad r>a\end{aligned}$$

or writing in integral form

$$\begin{aligned}&\int_0^\infty sA(s)J_0(sr)\,ds = \phi_0, \qquad 0\le r<a\\ &\int_0^\infty s^2A(s)J_0(sr)\,ds = 0, \qquad r>a\end{aligned} \tag{28}$$

Equations of the type in Eq. (28) are called *dual integral equations*. A systematic treatment of this kind of equations will be discussed later. Here, we give a rather heuristic solution. Gradshteyn and Ryzhik (13) provide the following integrals:

$$\begin{aligned}&\int_0^\infty \frac{\sin s}{s}J_0(sr)\,ds = \frac{\pi}{2}, \qquad 0\le r<a\\ &\int_0^\infty (\sin s)J_0(sr)\,ds = 0, \qquad r>a\end{aligned} \tag{29}$$

A comparison of Eqs. (28) with Eqs. (29) shows that the solution for $A(s)$ is

$$A(s) = \frac{2\phi_0}{\pi}\frac{\sin s}{s} \tag{30}$$

Putting Eq. (30) into Eq. (27), we obtain the solution of our problem as

$$\phi(r,z) = \frac{2\phi_0}{\pi}\int_0^\infty \frac{\sin s}{s} J_0(sr)e^{-sz}\,ds \tag{31}$$

The uniqueness of Eq. (31) follows from the physical contents of the problem.

RELATION BETWEEN FOURIER AND HANKEL TRANSFORMS

In this section, the relationship between Hankel and Fourier transforms of a function of two variables is explored. Specifically, we shall see that there exists a close relationship between the double Fourier transform of a function of two variables of a particular type and its Hankel transform.

Consider a function $f(x_1, x_2)$ that is a function of $r = x_1^2 + x_2^2$ only. The double Fourier transform $F(\alpha_1, \alpha_2)$ is

$$F(\alpha_1,\alpha_2) = \frac{1}{2\pi}\int_{-\infty}^\infty\int_{-\infty}^\infty f(\sqrt{x_1^2+x_2^2})e^{i(\alpha_1x_1+\alpha_2x_2)}\,dx_1\,dx_2 \tag{32}$$

If we make the substitutions into Eq. (32)

$$x_1 = r\cos\theta, \qquad x_2 = r\sin\theta, \qquad \alpha_1 = s\cos\varphi, \qquad \alpha_2 = s\sin\varphi$$

then, since

$$dx_1\,dx_2 = r\,dr\,d\theta, \quad \alpha_1x_1+\alpha_2x_2 = rs\cos(\theta-\varphi)$$

the double integral in Eq. (32) reduces to

$$F(\alpha_1,\alpha_2) = \frac{1}{2\pi}\int_0^\infty rf(r)\,dr\int_0^{2\pi} e^{irs\cos(\theta-\varphi)}\,d\theta \tag{33}$$

Since the inner integral on the right is 2π-periodic, it does not depend on φ, that is

$$\int_0^{2\pi} e^{irs\cos(\theta-\varphi)}\,d\theta = \int_0^{2\pi} e^{irs\cos\theta}\,d\theta$$

which is equal to $2\pi J_0(rs)$ [Watson (6), Erdelyi et al. (10)], where $s = \sqrt{\alpha_1^2+\alpha_2^2}$. We therefore see that the function $F(\alpha_1, \alpha_2)$ is a function of s only and may be written as

$$F(s) = \int_0^\infty rf(r)J_0(sr)\,dr = \mathscr{H}_0[f(r); r\to s] \tag{34}$$

which, of course, is the zeroth-order Hankel transform of $f(r)$. On the other hand, by the Fourier inversion theorem, we have

$$f(x_1,x_2) = \frac{1}{2\pi}\int_{-\infty}^\infty\int_{-\infty}^\infty F(\alpha_1,\alpha_2)e^{-i(\alpha_1x_1+\alpha_2x_2)}\,d\alpha_1\,d\alpha_2$$

Using the same substitution as before, the preceding expression can be reduced to the following formula:

$$f(r) = \int_0^\infty sF(s)J_0(sr)\,ds = \mathscr{H}_0[F(s); s\to r] \tag{35}$$

Formulas in Eqs. (34) and (35) obviously express the Hankel inversion theorem in the special case where $\nu = 0$.

The preceding results can be easily generalized in case of n-dimensional Fourier transforms. If the function $f(x_1, x_2, \ldots, x_n)$ is a function only of $r = \sqrt{x_1^2+x_2^2+\cdots+x_n^2}$, then its Fourier transform $F(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is a function of s only where $s = \sqrt{\alpha_1^2+\alpha_2^2+\cdots+\alpha_n^2}$. More specifically, the following relationship holds:

$$s^{(n-1)/2}F(s) = \int_0^\infty r[r^{(n-1)/2}f(r)]J_{(n-1)/2}(sr)\,dr \tag{36}$$

For proof, the readers are referred to the book by Sneddon (1).

It therefore follows from Eq. (36) that $s^{(n-1)/2}F(s)$ is the Hankel transform of order $(n-1)/2$ of the function $r^{(n-1)/2}f(r)$.

Similarly, by n-dimensional Fourier inversion theorem, it can be shown that

$$r^{(n-1)/2}f(r) = \int_0^\infty s[s^{(n-1)/2}F(s)]J_{(n-1)/2}(sr)\,ds \tag{37}$$

If we write

$$\phi(r) = r^{(n-1)/2}f(r), \qquad \tilde{\phi}_\nu(s) = s^{(n-1)/2}F(s), \qquad \nu = \frac{n-1}{2}$$

then Eqs. (36) and (37) take the following form:

$$\tilde{\phi}_\nu(s) = \int_0^\infty r\phi(r)J_\nu(sr)\,dr = \mathscr{H}_\nu[\phi(r); r\to s]$$
$$\phi(r) = \int_0^\infty s\tilde{\phi}_\nu(s)J_\nu(sr)\,ds = \mathscr{H}_\nu[\tilde{\phi}_\nu(s); s\to r]$$

The above formulas obviously define the νth-order Hankel transformation pair for the function $\phi(r)$.

PARSEVAL'S RELATION FOR HANKEL TRANSFORMS

Suppose that

$$\tilde{f}_\nu(s) = \mathscr{H}_\nu[f(r); r\to s], \qquad \tilde{g}_\nu(s) = \mathscr{H}_\nu[g(r); r\to s]$$

Then, putting formally, we obtain the equation

$$\begin{aligned}\int_0^\infty s\tilde{f}_\nu(s)\tilde{g}_\nu(s)\,ds &= \int_0^\infty s\tilde{f}_\nu(s)\,ds\int_0^\infty xg(x)J_\nu(sx)\,dx \\ &= \int_0^\infty xg(x)\,dx\int_0^\infty s\tilde{f}_\nu(s)J_\nu(sx)\,ds\end{aligned} \tag{38}$$

in which the inner integral, by Hankel's inversion theorem, is obviously equal to $f(r)$.

Equation (38) then yields the following formula:

$$\int_0^\infty s\tilde{f}_\nu(s)\tilde{g}_\nu(s)\,ds = \int_0^\infty xf(x)g(x)\,dx \tag{39}$$

The expression in Eq. (39) is evidently the Parseval relation for the Hankel transform. As in the case of other integral transforms, such as Fourier, Laplace, Mellin, and Kantorovich-Lebedev transforms, Parseval's relation is a very useful tool in many theoretical and practical investigations.

It should be noted here that a general Parseval relation involving Hankel transforms of two functions of different orders does not exist. This is primarily because the Neumann-Rahman formula (6,14) for the product of two first-kind Bessel functions of different orders,

$$J_{m+n}(sr)J_n(sr_0) = \frac{1}{\pi}\int_0^\pi \left\{\cos(n\varphi)T_m\left(\frac{r-r_0\cos\varphi}{R}\right) + \frac{r_0\sin n\varphi\sin\varphi}{R} \times U_{m-1}\left(\frac{r-r_0\cos\varphi}{R}\right)\right\} J_m(R)\,d\varphi, \qquad U_{-1}(\cdots)=0 \quad (40)$$

where $R = \sqrt{r^2 + r_0^2 - 2rr_0\cos\varphi}$, $T_m(\cdots)$ and $U_{m-1}(\cdots)$ are the Chebyshev polynomials of the first and second kinds, respectively, is much more complicated than the simplest rule for the product of two exponential functions (kernels of Laplace and Fourier transforms) of different powers.

As an example of application of Parseval's relation in Eq. (39), let us evaluate the integral

$$\mathscr{H}_\nu[x^{-2}J_\nu(ax); x\to s], \qquad \nu > -\tfrac{1}{2}$$

Taking $f(x) = x^\nu H(a-x)$ $(a>0)$ and $g(x) = x^\nu H(b-x)$ $(b>0)$, where $\mathscr{H}(\cdots)$ is the step function, we have

$$\tilde{f}_\nu(s) = \int_0^a x^{\nu+1}J_\nu(sx)\,dx; \qquad \tilde{g}_\nu(s) = \int_0^b x^{\nu+1}J_\nu(sx)\,dx$$

These integrals are easily evaluated [Gradshteyn and Ryzhik (13)] as

$$\tilde{f}_\nu(s) = \frac{a^{\nu+1}}{s}J_{\nu+1}(sa), \qquad \tilde{g}_\nu(s) = \frac{b^{\nu+1}}{s}J_{\nu+1}(sb)$$

Now, using Parseval's relation in Eq. (39), we obtain

$$(ab)^{\nu+1}\int_0^\infty s^{-1}J_{\nu+1}(sa)J_{\nu+1}(sb)\,ds = \int_0^{\min(a,b)} x^{2\nu+1}\,dx$$

Assuming that $0<a<b$, we find that (13)

$$\int_0^\infty s^{-1}J_{\nu+1}(sa)J_{\nu+1}(sb)\,ds = \frac{1}{2(\nu+1)}\left(\frac{a}{b}\right)^{\nu+1} \qquad 0<a<b, \qquad \nu>-\tfrac{1}{2}$$

It therefore follows from the preceding equation that

$$\mathscr{H}_\nu[x^{-2}J_{\nu+1}(ax); x\to s] = \begin{cases} \dfrac{1}{2\nu}\left(\dfrac{s}{a}\right)^\nu, & 0<s<a \\ \dfrac{1}{2\nu}\left(\dfrac{a}{s}\right)^\nu, & s>a \end{cases}$$

where $\nu > \frac{1}{2}$.

THE HANKEL OPERATOR

In many theoretical investigations, it is more convenient to use a modified operator of Hankel transform, $S_{\eta,\alpha}$, instead of the operator $\mathscr{H}_\nu$. This modified Hankel operator is defined by the formula

$$S_{\eta,\alpha}[f(t); x] = 2^\alpha x^{-\alpha}\mathscr{H}_{2\eta+\alpha}[t^{-\alpha}f(t); t\to x] \quad (41)$$

so that

$$S_{\eta,\alpha}[f(t); x] = 2^\alpha x^{-\alpha}\int_0^\infty t^{1-\alpha}f(t)J_{2\eta+\alpha}(xt)\,dt \quad (42)$$

If we write

$$\tilde{f}_{\eta,\alpha}(x) = S_{\eta,\alpha}[f(t); x] \quad (43)$$

then from Eq. (41), we obtain

$$\mathscr{H}_{2\eta+\alpha}[t^{-\alpha}f(t); x] = 2^{-\alpha}x^\alpha\tilde{f}_{\eta,\alpha}(x) \quad (44)$$

Applying Hankel's inversion, we deduce from Eq. (43) that

$$f(t) = 2^{-\alpha}t^\alpha\mathscr{H}_{2\eta+\alpha}[x^\alpha\tilde{f}_{\eta,\alpha}(x); t]$$

or writing out the above expression in full, we obtain

$$f(t) = S_{\eta+\alpha,-\alpha}[\tilde{f}_{\eta,\alpha}(x); t]$$

thus establishing the rule

$$S^{-1}_{\eta,\alpha} = S_{\eta+\alpha,-\alpha} \quad (45)$$

In applications, the following relationship is useful:

$$S_{\eta,\alpha}f(x) = 2^{-\lambda}x^\lambda S_{\eta\lambda/2,\alpha+\lambda}[x^\lambda f(x)]$$

the validity of which can be easily proved by writing out both sides of the equation using the definition in Eq. (42).

THE ERDELYI–KOBER OPERATORS OF FRACTIONAL INTEGRATION

In this section, we present a brief exposition of the so-called Erdelyi–Kober operators of fractional integrations (15–17) and their generalization due to Sneddon and Erdelyi (8,18) and Cooke (19,20). We next illustrate applications of these operators to the solution of dual, triple and quadruple integral equations involving Hankel transforms, that arise in many boundary value problems of mathematical physics, especially electrostatics and electromagnetic scattering. The description here closely follows Sneddon (21).

In a series of papers (15–17), Erdelyi and Kober investigated the properties of the fractional integral

$$\frac{x^{-\eta-\alpha+1}}{\Gamma(\alpha)}\int_0^x (x-t)^{\alpha-1}t^{\eta-1}f(t)\,dt \qquad (\alpha>0, \qquad \eta>0)$$

which is a generalization of Riemann's integral

$$\frac{1}{\Gamma(\alpha)}\int_0^x (x-t)^{\alpha-1}f(t)\,dt$$

and Weyl's integral

$$\frac{x^n}{\Gamma(\alpha)}\int_x^\infty (t-x)^{\alpha-1}t^{-\alpha-\eta}f(t)\,dt \qquad (\alpha>0, \qquad \eta>0)$$

Definitions and Basic Results

If $\alpha>0$, $\eta>-\frac{1}{2}$, we define the operator $I_{\eta,\alpha}$ by the equation

$$I_{\eta,\alpha}f(x)=\frac{2x^{-2\alpha-2\eta}}{\Gamma(\alpha)}\int_0^x (x^2-u^2)^{\alpha-1}u^{2\eta+1}f(u)\,du$$

$I_{\eta,0}$ is the identity operator, and if $\alpha<0$, we define $I_{\eta,\alpha}$ by the relation

$$I_{\eta,\alpha}f(x)=x^{-2\eta-2\alpha-1}D_x^n\,x^{2\eta+2\alpha+2n+1}I_{\eta,\alpha+n}f(x)$$

where n is a positive integer such that $0<\alpha+n<1$ and D_x is the differential operator

$$D_x=\frac{d}{dx}\,x^{-1}$$

Similarly, if $\alpha>0$, $\eta>-\frac{1}{2}$, we define the operator $K_{\eta,\alpha}$ by the equation

$$K_{\eta,\alpha}f(x)=\frac{2x^{2\eta}}{\Gamma(\alpha)}\int_x^\infty (u^2-x^2)^{\alpha-1}u^{-2\alpha-2\eta+1}f(u)\,du$$

$K_{\eta,0}$ is the identity operator, and if $\alpha<0$, we define $K_{\eta,\alpha}$ by the equation

$$K_{\eta,\alpha}f(x)=(-1)^n x^{2\eta-1}D_x^n\,x^{2n-2+1}K_{\eta-n,\alpha+n}f(x)$$

Operators $I_{\eta,\alpha}$ and $K_{\eta,\alpha}$ are called Erdelyi–Kober operators.

We next establish some properties of these operators. If we assume that $\alpha>0$, $\beta>0$, we have

$$I_{\eta,\alpha}I_{\eta+\alpha,\beta}f(x)=\frac{2x^{-2\eta-2\alpha}}{\Gamma(\alpha)}\int_0^x (x^2-u^2)^{\alpha-1}u^{2\eta+1}du\,\frac{2u^{-2\eta-2\alpha-2\beta}}{\Gamma(\beta)}\times\int_0^u (u^2-t^2)^{B-1}t^{2\eta+2\alpha+1}f(t)\,dt$$

Interchanging the order of integration and using the result (13)

$$2\int_t^x (x^2-u^2)^{\alpha-1}(u^2-t^2)^{\beta-1}u^{-2\alpha-2\beta+1}\,du=\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}\,t^{-2\alpha}x^{-2\beta}(x^2-t^2)^{\alpha+\beta-1}$$

we obtain

$$I_{\eta,\alpha}I_{\eta+\alpha,\beta}f(x)=\frac{2x^{-2\eta-2\alpha-\beta}}{\Gamma(\alpha+\beta)}\int_0^x t^{2\eta+1}(x^2-t^2)^{\alpha+\beta-1}f(t)\,dt$$

The expression on the right is equal to $I_{\eta,\alpha+\beta}$, which follows from its definition, thus establishing the rule

$$I_{\eta,\alpha}I_{\eta+\alpha,\beta}=I_{\eta,\alpha+\beta} \tag{46}$$

Similarly, it can be shown that

$$K_{\eta,\alpha}K_{\eta+\alpha,\beta}=K_{\eta,\alpha+\beta} \tag{47}$$

The preceding relations are valid for $\alpha>0$, $\beta>0$, but it is a simple exercise to show that they are also valid for negative values of α and β. Also, it can be shown from the theory of integral equations of Abel type (8) that the inverse of the Erdelyi–Kober operators are given by the formulas:

$$I_{\eta,\alpha}^{-1}=I_{\eta+\alpha,-\alpha}, \qquad K_{\eta,\alpha}^{-1}=K_{\eta+\alpha,-\alpha} \tag{48}$$

The following formulas hold, whose validity can be proved very easily:

$$I_{\eta,\alpha}\{x^{2\beta}f(x)\}=x^{2\beta}I_{\eta+\beta,\alpha}f(x)$$
$$K_{\eta,\alpha}\{x^{2\beta}f(x)\}=x^{2\beta}K_{\eta+\beta,\alpha}f(x)$$

The following relationships hold between the Erdelyi-Kober and Hankel operators:

$$\begin{aligned} I_{\eta+\alpha,\beta}S_{\eta,\alpha}&=S_{\eta,\alpha+\beta}, & K_{\eta,\alpha}S_{\eta+\alpha,\beta}&=S_{\eta,\alpha+\beta}\\ S_{\eta+\alpha,\beta}S_{\eta,\alpha}&=I_{\eta,\alpha+\beta} & S_{\eta,\alpha}S_{\eta+\alpha,\beta}&=K_{\eta,\alpha+\beta}\\ S_{\eta+\alpha,\beta}I_{\eta,\alpha}&=S_{\eta,\alpha+\beta}, & S_{\eta,\alpha}K_{\eta+\alpha,\beta}&=S_{\eta,\alpha+\beta}\end{aligned} \tag{49}$$

The proofs of these identities are based on the properties of Bessel functions and are given in the book by Davies (3).

The Cooke Operators

Cooke (19,20) has defined the operators

$$\binom{b}{a}I_{\eta,\alpha}$$

and

$$\binom{d}{c}K_{\eta,\alpha}$$

by the formulas

$$\binom{b}{a}I_{\eta,\alpha}f(x)=\begin{cases}\dfrac{2x^{-2\alpha-2\eta}}{\Gamma(\alpha)}\displaystyle\int_a^b (x^2-u^2)^{\alpha-1}u^{2\eta+1}f(u)\,du, & \alpha>0\\ f(x), & \alpha=0\\ \dfrac{x^{-2\alpha-2\eta-1}}{\Gamma(1+\alpha)}\dfrac{d}{dx}\displaystyle\int_a^b (x^2-u^2)^{\alpha}u^{2\eta+1}f(u)\,du, & -1<\alpha<0\end{cases} \tag{50}$$

for $0<a<b<\infty$,

$$\binom{d}{c}K_{\eta,\alpha}f(x)=\begin{cases}\dfrac{2x^{2\eta}}{\Gamma(\alpha)}\displaystyle\int_c^d (u^2-x^2)^{\alpha-1}u^{-2\alpha-2\eta+1}f(u)\,du, & \alpha>0\\ f(x), & \alpha=0\\ \dfrac{-x^{2\eta-1}}{\Gamma(1+\alpha)}\dfrac{d}{dx}\displaystyle\int_c^d (u^2-x^2)^{\alpha}u^{-2\alpha-2\eta+1}f(u)\,du, & -1<\alpha<0\end{cases} \tag{51}$$

for $0 < x < c < d$. It will be observed that these operators are related to the Erdelyi–Kober operators by the relations

$$\begin{pmatrix} x \\ 0 \end{pmatrix} I_{\eta,\alpha} = I_{\eta,\alpha}, \qquad \begin{pmatrix} \infty \\ x \end{pmatrix} K_{\eta,\alpha} = I_{\eta,\alpha}$$

Cooke (19,20) also defined the operators L and M by the equations

$$\begin{aligned} \begin{pmatrix} x, & b \\ c, & a \end{pmatrix} L_{\eta,\alpha} f(x) &= \begin{pmatrix} x \\ c \end{pmatrix} I^{-1}_{\eta,\alpha} \begin{pmatrix} b \\ a \end{pmatrix} I_{\eta,\alpha} f(x) \\ \begin{pmatrix} d, & b \\ x, & a \end{pmatrix} M_{\eta,\alpha} f(x) &= \begin{pmatrix} d \\ x \end{pmatrix} K^{-1}_{\eta,\alpha} \begin{pmatrix} b \\ a \end{pmatrix} K_{\eta,\alpha} f(x) \end{aligned} \tag{52}$$

and showed that if $a < b < c < x$,

$$\begin{pmatrix} x, & b \\ c, & a \end{pmatrix} L_{\eta,\alpha} f(x) = \frac{2\sin(\pi\alpha)}{\pi} x^{-2\eta}(x^2 - c^2)^{-\alpha} \int_a^b \frac{(c^2 - t^2)^\alpha t^{2+1}}{x^2 - t^2} f(t)\, dt \tag{53}$$

and that if $x < d < a < b$

$$\begin{pmatrix} d, & b \\ x, & a \end{pmatrix} M_{\eta,\alpha} f(x) = \frac{2\sin(\pi\alpha)}{\pi} x^{2\eta+2\alpha}(d^2 - x^2)^{-\alpha} \int_a^b \frac{(t^2 - d^2)^\alpha t^{-2\alpha-2\eta+1}}{t^2 - x^2} f(t)\, dt \tag{54}$$

BELTRAMI-TYPE RELATIONS

A classic problem of electrostatics concerns that of determining the potential of the electrostatic field due to a circular disk whose potential is prescribed. One way to solve this problem is to determine the charge density q on the disk and then to calculate the potential at any field point $\boldsymbol{r}$ by evaluating the integral

$$\int_S \frac{q(\boldsymbol{R}')}{|\boldsymbol{R} - \boldsymbol{R}'|}\, dS'$$

over the surface of the disk. In the case of axisymmetry, that is, when the prescribed potential $\phi(r)$ is a function of r only, Beltrami (22) showed that the density of the surface charge is given by the formula

$$q(r) = \frac{-1}{\pi r}\frac{d}{dr}\int_r^\infty \frac{x\, dx}{\sqrt{x^2 - r^2}}\frac{d}{dx}\int_0^x \frac{y\phi(y)\, dy}{\sqrt{x^2 - y^2}}, \qquad 0 \le r \le a \tag{55}$$

where a is the radius of the disk.

Sneddon (23) showed that Beltrami's relation in Eq. (55) is a special case of a general relation between Hankel transforms. In particular, he showed that the expression

$$\mathscr{H}_\mu[s^\delta \mathscr{H}_\nu f(s); r]$$

can be expressed as a double integral involving $f(r)$, which is a generalization of the integral occurring on the right hand side of Beltrami's relation in Eq. (55). By assigning particular values of the parameters μ, δ, and ν, we can deduce relations that are of interest in the investigations into axisymmetric boundary-value problems of potential theory.

If we apply the operator $K_{\eta-\gamma,\gamma}$ to both sides of the first equation of Eqs. (49) and make use of the second relation of Eqs. (49), we obtain

$$K_{\eta-\gamma,\gamma} I_{\eta+\alpha,\beta} S_{\eta,\alpha} = S_{\eta-\gamma,\alpha+\beta+\gamma} \tag{56}$$

Equation (56) can be written in terms of Hankel transforms as follows:

$$\begin{aligned} &\mathscr{H}_{2\eta+\alpha+\beta-\gamma}[t^{-\alpha-\beta-\gamma} f(t); r] \\ &\qquad = \left(\frac{r}{2}\right)^{\alpha+\beta+\gamma} K_{\eta-\gamma,\gamma} I_{\eta+\alpha,\beta} 2^\alpha x^{-\alpha} \mathscr{H}_{2\eta+\alpha}[t^{-\alpha} f(t); x] \end{aligned} \tag{57}$$

For $\alpha = 0$, $\beta = (\mu - \nu - \delta)/2$, $\eta = \nu/2$, Eq. (57) simplifies significantly

$$\mathscr{H}_\mu[s^\delta \tilde{f}(s); r] = 2^\delta r^\delta K_{(\mu+\delta)/2,(\nu-\mu-\delta)/2} I_{\nu/2,(\mu-\nu-\delta)/2} f(r) \tag{58}$$

Some special cases of formulas in Eq. (58) are of particular interest. If we set $\mu = \nu$, we obtain

$$\mathscr{H}_\mu[s^\delta \tilde{f}(s); r] = 2^\delta r^{-\delta} K_{(\mu+\delta)/2,-\delta/2} I_{\nu/2,-\delta/2} f(r) \tag{59}$$

Special cases of particular interest are given by assigning $\delta = \pm 1$ to Eq. (59); we then obtain

$$\begin{aligned} &\mathscr{H}_\nu[s\tilde{f}_\nu(s); r] = \frac{-2}{\pi} r^{\nu-1}\frac{d}{dr}\int_r^\infty \frac{x^{1-2\nu}}{\sqrt{x^2 - r^2}}\frac{d}{dx}\int_0^x \frac{y^{\nu+1} f(y)\, dy}{\sqrt{x^2 - y^2}} \qquad (\nu \ge 0) \\ &\mathscr{H}_\nu[s^{-1}\tilde{f}(s); r] = \frac{2}{\pi} r^\nu \int_r^\infty \frac{x^{-2\nu}}{\sqrt{x^2 - r^2}}\int_0^x \frac{y^{\nu+1} f(y)\, dy}{\sqrt{x^2 - y^2}} \qquad (\nu \ge 0) \end{aligned} \tag{60}$$

On the other hand, if we put $\mu = \nu + 1$ in Eq. (59), we obtain the relation

$$\mathscr{H}_{\nu+1}[s^\delta \tilde{f}_\nu(s); r] = 2^\delta r^{-\delta} K_{(\nu+\delta+1)/2,(-1-\delta)/2} I_{\nu/2,(1-\delta)/2} f(r)$$

The special case $\delta = 1$ corresponds to the well-known formula

$$\mathscr{H}_{\nu+1}[s\tilde{f}_\nu(s); r] = -r^\nu \frac{d}{dr}[r^{-\nu} f(r)] \tag{61}$$

Expressions corresponding to the particular values 0 and -1 of δ are, respectively,

$$\begin{aligned} &\mathscr{H}_{\nu+1}[\tilde{f}_\nu(s); r] = \frac{-2}{\pi} r^\nu \frac{d}{dr}\int_r^\infty \frac{x^{-2\nu}\, dx}{\sqrt{x^2 - r^2}}\int_0^x \frac{y^{\nu+1} f(y)\, dy}{\sqrt{x^2 - y^2}} \qquad (\nu \ge 0) \\ &\mathscr{H}_{\nu+1}[s^{-1}\tilde{f}_\nu(s); r] = r^{-\nu-1}\int_0^r u^{\nu+1} f(u)\, du \qquad (\nu \ge 0) \end{aligned} \tag{62}$$

Finally, if we set $\mu = \nu - 1$ in Eq. (59), we obtain the relation

$$\mathscr{H}_{\nu-1}[s^\delta \tilde{f}_\nu(s); r] = 2^\delta r^{-\delta} K_{(\nu-1+\delta)/2,(\nu-1-\delta)/2} f(r) \tag{63}$$

The most frequently occurring special cases of the formula in Eq. (63) are

$$\mathscr{H}_{\nu-1}[s\tilde{f}_\nu(s);r] = r^{-\nu}\frac{d}{dr}[r^\nu f(r)] \qquad (\nu \geq 1)$$

$$\mathscr{H}_{\nu-1}[\tilde{f}_\nu(s);r] = \frac{2}{\pi}r^{\nu-1}\int_r^\infty \frac{x^{1-2\nu}\,dx}{\sqrt{x^2-r^2}}\frac{d}{dx}\int_0^x \frac{y^{\nu+1}f(y)\,dy}{\sqrt{x^2-y^2}} \qquad (\nu \geq 1)$$

$$\mathscr{H}_{\nu-1}[s^{-1}\tilde{f}_\nu(s);r] = r^{\nu-1}\int_r^\infty x^{1-\nu}f(x)\,dx \qquad (\nu \geq 1) \tag{64}$$

Beltrami's Relation for an Electrified Disk

As an application of Beltrami-type relations just derived, let us consider the problem of an electrified disk of radius a lying in the plane $z = 0$ with its center at the origin of the coordinate system. Let the surface charge density be $q(r)$. Then in the half-space $z \geq 0$ the potential of the electrostatic field will be $\phi_+(r,z)$ and in the half-space $z \leq 0$, it will be $\phi_-(r,z)$, where

$$\phi_\pm(r,z) = \mathscr{H}_0[\tilde{\phi}_0(s)e^{\pm sz};r]$$

where

$$\tilde{\phi}_0(s) = \mathscr{H}_0[\phi(r,0);s]$$

The charge density on the plane $z = 0$ is given by the equation

$$q(r) = \frac{-1}{4\pi}\left(\frac{\partial\phi_+}{\partial z} - \frac{\partial\phi}{\partial z}\right)_{z=0}$$

and it immediately follows from equation that

$$q(r) = \frac{1}{2\pi}\mathscr{H}_0[s\tilde{\phi}_0(s);r] \tag{65}$$

From the first equation of Eqs. (60) then we deduce Beltrami's relation in Eq. (55). On the other hand, we could write Eq. (65) in the form

$$\phi(r,0) = 2\pi\mathscr{H}_0[s^{-1}\tilde{q}_0(s);r]$$

and then using the second relation of Eq. (60) deduce the equation

$$\phi(r,0) = 4\int_r^\infty \frac{dx}{\sqrt{x^2-r^2}}\int_0^{\min(a,x)} \frac{yq(y)\,dy}{\sqrt{x^2-y^2}}$$

Interchanging the order of integration, the last equation can be written as

$$\phi(r,0) = \int_0^a \sigma(y)K(r,y)\,dy$$

where

$$K(r,y) = 4y\int_{\min(r,y)}^\infty \frac{du}{\sqrt{(u^2-r^2)(u^2-y^2)}}$$

DUAL INTEGRAL EQUATIONS INVOLVING HANKEL TRANSFORMS

In the applications of the theory of Hankel transforms to the solution of boundary-value problems of mathematical physics, it often happens that the problem may be reduced to the solution of a pair of simultaneous equations of the form

$$f(x) = S_{\mu/2-\alpha,2\alpha}[1+k(x)]\psi(x); \quad g(x) = S_{\nu/2-\beta,2\beta}\psi(x) \tag{66}$$

in which

$$f(x) = \begin{cases} f_1(x), & x \in I_1 = \{x: 0 < x < 1\} \\ f_2(x), & x \in I_2 = \{x: 1 < x < \infty\} \end{cases}$$

$$g(x) = \begin{cases} g_1(x), & x \in I_1 = \{x: 0 < x < 1\} \\ g_2(x), & x \in I_2 = \{x: 1 < x < \infty\} \end{cases}$$

The problem is as follows: Knowing the functions $k(x)$ [$k(x) \to 0$, $x \to \infty$], f_1, and g_2, is it possible to find the functions ψ, f_2, and g_1? In the following, we consider the special case where $k(x) = 0$, but it is straightforward to generalize the results for $k(x) \neq 0$.

To solve the problem, Sneddon proposed the following trial solution:

$$\psi(x) = S_{\nu/2+\beta,\mu/2-\nu/2-\alpha-\beta}h(x) \tag{67}$$

Putting Eq. (67) into Eqs. (65), we obtain

$$S_{\mu/2-\alpha,2\alpha}S_{\nu/2+\beta,\mu/2-\nu/2-\alpha-\beta}h = f$$
$$S_{\nu/2-\beta,2\beta}S_{\nu/2+\beta,\mu/2-\nu/2-\alpha-\beta}h = g$$

which can be rewritten, using the third and fourth relations of Eq. (49), as

$$I_{\nu/2+\beta,\nu/2-\nu/2+\alpha-\beta}h = f$$
$$K_{\nu/2-\beta,\mu/2-\nu/2-\alpha+\beta}h = g$$

whence

$$h = I^{-1}_{\nu/2+\beta,\mu/2-\nu/2+\alpha-\beta}f$$
$$h = K^{-1}_{\nu/2-\beta,\mu/2-\nu/2-\alpha+\beta}g \tag{68}$$

Writing Eqs. (68) on the intervals I_1 and I_2, we have

$$h_1(x) = \begin{pmatrix} x \\ 0 \end{pmatrix} I^{-1}_{\nu/2+\beta,\mu/2-\nu/2+\alpha-\beta}f_1$$
$$h_2(x) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} I^{-1}_{\nu/2+\beta,\mu/2-\nu/2+\alpha-\beta}f_1 + \begin{pmatrix} x \\ 1 \end{pmatrix} I^{-1}_{\nu/2+\beta,\mu/2-\nu/2+\alpha-\beta}f_2$$
$$h_2(x) = \begin{pmatrix} \infty \\ x \end{pmatrix} K^{-1}_{\nu/2-\beta,\mu/2-\nu/2-\alpha+\beta}g_2 \tag{69}$$
$$h_1(x) = \begin{pmatrix} \infty \\ 1 \end{pmatrix} K^{-1}_{\nu/2-\beta,\mu/2-\nu/2-\alpha+\beta}g_2 + \begin{pmatrix} 1 \\ x \end{pmatrix} K^{-1}_{\nu/2-\beta,\mu/2-\nu/2-\alpha+\beta}g_1$$

Putting the first and third equations of Eqs. (69) into Eq. (67), we obtain the solution for $\psi(x)$. On the other hand, from the second and third equations of Eqs. (69), we deduce that

$$\begin{pmatrix} x \\ 1 \end{pmatrix} I^{-1}_{\nu/2+\beta,\mu/2-\nu/2+\alpha-\beta}f_2 = \begin{pmatrix} \infty \\ x \end{pmatrix} K^{-1}_{\nu/2-\beta,\mu/2-\nu/2-\alpha+\beta}g_2 - \begin{pmatrix} 1 \\ 0 \end{pmatrix} I^{-1}_{\nu/2+\beta,\mu/2-\nu/2+\alpha-\beta}f_1$$

whence it follows by use of the L operator defined by Eq. (53) that

$$f_2 = \begin{pmatrix} x \\ 1 \end{pmatrix} I_{\nu/2+\beta,\mu/2-\nu/2+\alpha-\beta} \begin{pmatrix} \infty \\ x \end{pmatrix} K^{-1}_{\nu/2-\beta,\mu/2-\nu/2-\alpha+\beta} g_2 - \begin{pmatrix} x, & 1 \\ 1, & 0 \end{pmatrix} L_{\nu/2+\alpha,\nu/2-\mu/2+\beta-\alpha} f_1 \tag{70}$$

Thus, f_2 is determined. Similarly, using the first and fourth equations of Eqs. (68), we obtain for g_1 the formula:

$$g_1 = \begin{pmatrix} 1 \\ x \end{pmatrix} K_{\nu/2-\beta,\mu/2-\nu/2-\alpha+\beta} I_{\nu/2+\beta,\mu/2-\nu/2+\alpha-\beta} f_1 - \begin{pmatrix} 1, & \infty \\ x, & 1 \end{pmatrix} M_{\nu/2-\beta,\mu/2-\nu/2-\alpha+\beta} g_2 \tag{71}$$

Thus, the first two equations in Eqs. (69) and Eqs. (70) and (71) give the complete solution to our problem. The same procedure, applied to the case where $k(x) \neq 0$, yields

$$\begin{aligned} h_1 + E(x) &= \begin{pmatrix} x \\ 0 \end{pmatrix} I^{-1} f_1 \qquad (x \in I_1) \\ h_2 + E(x) &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} I^{-1} f_1 + \begin{pmatrix} x \\ 1 \end{pmatrix} I^{-1} f_2 \qquad (x \in I_2) \\ h_2 &= \begin{pmatrix} \infty \\ x \end{pmatrix} K^{-1} g_2 \qquad (x \in I_2) \\ h_1 &= \begin{pmatrix} \infty \\ 1 \end{pmatrix} K^{-1} g_2 + \begin{pmatrix} 1 \\ x \end{pmatrix} K^{-1} g_1 \qquad (x \in I_1) \end{aligned} \tag{72}$$

where

$$E(x) = S_{\mu/2-\alpha,\nu/2+\beta+\alpha-\mu/2} k S_{\nu/2+\beta,\mu/2-\nu/2-\alpha-\beta} h(x) \tag{73}$$

The subscripts with the I and K in Eqs. (72) are the same as those in Eqs. (69). Further details are carried out for the special case where $\nu = \mu$, $\beta = 0$, $g_2 = 0$, which is the most frequently occurring case in applications. In this case, we find from Eqs. (72) that $h_2(x) = 0$ and $h_1(x)$ solves the integral equation

$$h_1(x) + E(x) = \begin{pmatrix} x \\ 0 \end{pmatrix} I^{-1}_{\nu/2,\alpha} f_1 \qquad (x \in I_1) \tag{74}$$

where

$$\begin{aligned} E(x) &\equiv S_{\nu/2-\alpha,\alpha} k S_{\nu/2,-\alpha} h(x) \\ &= 2^{\alpha} x^{-\alpha} \int_0^{\infty} t^{1-\alpha} k(t) 2^{-\alpha} t^{\alpha} J_{\nu-\alpha}(xt)\, dt \int_0^1 u^{1+\alpha} h_1(u) J_{\nu-\alpha}(tu)\, du \end{aligned}$$

and inverting the order of intergration, we have

$$E(x) = x^{-\alpha} \int_0^1 u^{1+\alpha} K(x,u) h_1(u)\, du \tag{75}$$

where

$$K(x,u) = \int_0^{\infty} t k(t) J_{\nu-\alpha}(xt) J_{\nu-\alpha}(ut)\, dt \tag{76}$$

Since the functions h_1 and h_2 have been determined, it is possible to find the functions ψ, f_2, and g_1 following the procedure for the case $k(x) = 0$. These details can be found in the papers by Sneddon (8) and Cooke (19,20).

An Example: Two Coaxial Electrified Circular Disks

The problem of two solid disks, each charged to a uniform potential ϕ_0, was the subject of numerous research starting with Love's paper (24) [for references see Cooke (19)]. If the disks have different potentials the problem may be reduced to two separate problems, in one of which the potentials are equal and in the other they are equal and opposite. Assume that the disks have the same radii, equal to unity, and are situated in the planes $z = 0$ and $z = h$, where r, θ, and z are cylindrical coordinates. Then, the problem reduces to that of solving Laplace's equation in Eq. (1) subject to the following boundary conditions:

$$\begin{aligned} \phi(r,0) &= \phi_0, \qquad 0 < r < 1 \\ \phi(r,0_-) &= \phi(r,0_+), \qquad 0 \le r < \infty \\ \left.\frac{\partial \phi}{\partial z}\right|_{z=0-} &= \left.\frac{\partial \phi}{\partial z}\right|_{z=0+}, \qquad r > 1 \\ \phi(r,h) &= \pm\phi_0, \qquad 0 < r < 1 \\ \phi(r,h_-) &= \phi(r,h_+), \qquad 0 \le r < \infty \\ \left.\frac{\partial \phi}{\partial z}\right|_{z=h-} &= \left.\frac{\partial \phi}{\partial z}\right|_{z=h+}, \qquad r > 1 \end{aligned} \tag{77}$$

The sign in fourth of the preceding conditions is positive or negative according to whether the disks are of like or unlike potentials ϕ_0.

The solution of the problem must satisfy the regularity conditions at infinity. Besides, in order to guarantee uniqueness of solution, it must satisfy the *edge condition* [see Meixner (25) and Mittra and Lee (26)] so that the electric energy stored in any neighborhood of the sharp edge $r = 1$ be finite, which imposes the restriction that the surface charge density not grow more rapidly than $\rho^{-1+\tau}$ with $\tau > 0$ as $\rho \to 0$, where $\rho = 1 - r$.

It can be shown by using zeroth-order Hankel transform to Laplace's equation in Eq. (1) that the electrostatic field can be represented by the potential function

$$\phi(r,z) = \mathsf{H}_0[\phi_0 s^{-1}(e^{-|z|s} + e^{-|z-h|s}) A(s); s \to r] \tag{78}$$

which satisfies the second and fifth continuity conditions in Eqs. (77), the sign in Eq. (78) being positive or negative depending on whether the disks are of like or unlike potentials ϕ_0.

We find that the third and sixth conditions in Eqs. (77) will be satisfied if the function $A(s)$ satisfies the equation

$$\mathscr{H}_0[A(s); s \to r] = 0, \qquad r > 1 \tag{79}$$

Using the second and fourth boundary conditions in Eqs. (77) and Eqs. (79), we obtain the following dual integral equations:

$$\begin{aligned} \mathscr{H}_0[s^{-1}(1 \pm e^{-hs}) A(s); s \to r] &= 1, \qquad 0 \le r < 1 \\ \mathscr{H}_0[A(s); s \to r] &= 0, \qquad r > 1 \end{aligned} \tag{80}$$

Using the modified operator of Hankel transform, we rewrite Eqs. (80) in the form

$$\begin{aligned} S_{-1/2,1}[1 \pm k(r)]A(r) &= 1, \qquad 0 \le r < 1 \\ S_{0,0}A(r) &= 0, \qquad r > 1 \end{aligned} \tag{81}$$

where $k(s) = e^{-hs}$. Thus, for our problem

$$\alpha = \tfrac{1}{2}, \qquad \mu = 0, \qquad \beta = 0, \qquad \nu = 0$$
$$f_1(r) = \frac{r}{2\phi_0}, \qquad g_2(r) = 0$$

Therefore, following the procedure outlined in the previous section, we find that $h_2(r) = 0$ and $h_1(r)$ solves the following integral equation:

$$h_1(r) + r^{-1/2}\int_0^1 u^{3/2}K(r,u)h_1(u)\,du = \begin{pmatrix} r \\ 0 \end{pmatrix} I_{1/2,-1/2}f_1(r) \qquad 0 \le r < 1 \tag{82}$$

where

$$K(r,u) = \pm\int_0^\infty tk(t)J_{-1/2}(rt)J_{-1/2}(ut)\,dt$$

Writing $rh_1(r) = H(r)$, we reduce Eq. (82) to the following Fredholm integral equation of the second kind:

$$H(r) + \int_0^1 H(u)N(r,u)\,du = \begin{pmatrix} r \\ 0 \end{pmatrix} I_{1/2,-1/2}f_1(r) \tag{83}$$

where

$$N(r,u) = \pm\sqrt{ru}\int_0^\infty tk(t)J_{-1/2}(rt)J_{-1/2}(ut)\,dt \tag{84}$$

The kernel $N(r, u)$ in Eq. (84) can be evaluated in closed form, namely,

$$N(r,u) = \pm\left(\frac{1}{(r+u)^2+h^2} + \frac{1}{(r-u)^2+h^2}\right) \tag{85}$$

The integral equation defined by Eqs. (83) and (85) can be solved numerically.

The surface density at any point of a disk in the plane $z = 0$ is equal to

$$\frac{-1}{4\pi}\left(\frac{\partial\phi}{\partial z}\right)_{z=0}$$

When both sides of the disk are taken into account, this gives for the total charge Q

$$\begin{aligned} Q &= \frac{\phi_0}{2\pi}\int_0^1 2\pi r\,dr\int_0^\infty A(s)J_0(sr)\,ds \\ &= \phi_0\int_0^1 rg_1(r)\,dr = \phi_0\int_0^1 r\begin{pmatrix} 1 \\ r \end{pmatrix} K_{0,-1/2}h_1(r)\,dr \\ &= -\frac{\phi_0}{\sqrt{\pi}}\int_0^1 dr\,\frac{d}{dr}\int_x^1 \frac{u^2h_1(u)\,du}{\sqrt{u^2-1}} = \frac{2\phi_0}{\pi}\int_0^1 \frac{uH(u)\,du}{\sqrt{u^2-1}} \end{aligned} \tag{86}$$

Once the integral equation in Eq. (83) is solved for $H(r)$, the total charge can be found by evaluating the integral in Eq. (86) numerically and hence the capacity $C = Q/\phi_0$ can be found.

TRIPLE INTEGRAL EQUATIONS INVOLVING HANKEL TRANSFORMS

As an example of the use of Cooke operators, we consider the solution of certain triple integral equations involving Hankel transforms. The problem consists in finding a function $\Phi(\xi)$ satisfying

$$\begin{aligned} \int_0^\infty \Phi(\xi)J_\nu(\xi x)\,d\xi &= G_1(x), \qquad x \in I_1 \\ \int_0^\infty \xi^{-2\alpha}[1+k(\xi)]\Phi(\xi)J_\nu(\xi x)\,d\xi &= F_2(x), \qquad x \in I_2 \\ \int_0^\infty \Phi(\xi)J_\nu(\xi x)\,d\xi &= G_3(x), \qquad x \in I_3 \end{aligned} \tag{87}$$

where I_j (j = 1, 2, 3) denote, respectively, the intervals (0, a), (a, b), and (b, ∞) with $0 < a < b$. The functions G_1, F_2, and G_3 are assumed to be prescribed. Assuming that

$$\Phi(\xi) = \xi\psi(\xi), \qquad f(x) = \left(\frac{2}{x}\right)^{2\alpha}F(x), \qquad g(x) = G(x)$$

and using the modified operator of the Hankel transform, we rewrite Eqs. (87) in the form

$$\begin{aligned} S_{\nu/2-\alpha,2\alpha}[\{1+k(\xi)\}\psi(\xi);x] &= f(x) \\ S_{\nu/2,0}\psi(\xi) &= g(x) \end{aligned} \tag{88}$$

We first consider the case where $k = 0$, $g_1 = g_3 = 0$, $|\alpha| < 1$. There are two different ways of solving Eqs. (88), one proposed by Sneddon and the other by Borodachev.

The Sneddon Trial Solution

Sneddon proposed the following solution for the equations in Eq. (88):

$$\psi = S_{\nu/2,-\alpha}h \tag{89}$$

Then, putting Eq. (89) into Eq. (88), we find that

$$\begin{aligned} S_{\nu/2,0}\psi &= K_{\nu/2,\alpha}h = g(x) \\ S_{\nu/2-\alpha,2\alpha}\psi &= I_{\nu/2,-\alpha}h = f(x) \end{aligned}$$

and solving for h we obtain

$$\begin{aligned} h &= I^{-1}_{\nu/2,-\alpha}f(x) \\ h &= K^{-1}_{\nu/2,\alpha}g(x) \end{aligned} \tag{90}$$

Now, suppose that $f(x) = f_1(x)$, $x \in I_1$, $f(x) = f_3(x)$, $x \in I_3$, and $g(x) = g_2(x)$, $x \in I_2$. We also write $h(x) = h_j(x)$, $x \in I_j$.

If we evaluate Eqs. (90) on I_3 and use $g_3 = 0$, we deduce that $h_3 = 0$. Similarly, if we evaluate Eq. (90) on I_1, we have

$$f_1(x) = \begin{pmatrix} x \\ 0 \end{pmatrix} I_{\nu/2,\alpha}h_1(x) \tag{91}$$

and if we evaluate Eq. (90) on I_2 we have

$$h_2 = \begin{pmatrix} a \\ 0 \end{pmatrix} I^{-1}_{\nu/2,\alpha} f_1 + \begin{pmatrix} x \\ a \end{pmatrix} I^{-1}_{\nu/2,\alpha} f_2, \qquad x \in I_2 \tag{92}$$

Putting Eq. (92) into Eq. (91) and using the L operator defined by Eq. (53), we obtain

$$h_2 = -\begin{pmatrix} x, & a \\ a, & 0 \end{pmatrix} L_{\nu/2,\alpha} h_1 + \begin{pmatrix} x \\ a \end{pmatrix} I^{-1}_{\nu/2,\alpha} f_2, \qquad x \in I_2 \tag{93}$$

Now, evaluating Eq. (90) on I_2 and I_1, respectively, we obtain the equations

$$g_2 = \begin{pmatrix} b \\ x \end{pmatrix} K_{\nu/2,-\alpha} h_2, \qquad h_1 = \begin{pmatrix} b \\ a \end{pmatrix} K^{-1}_{\nu/2,-\alpha} g_2 \tag{94}$$

Putting first of the relations in Eq. (94) into the second and using the M operator defined by Eq. (54), we obtain

$$h_1 = -\begin{pmatrix} a, & b \\ x, & a \end{pmatrix} M_{\nu/2,-\alpha} h_2 \qquad (x \in I_1) \tag{95}$$

Equations (93) and (95) form a pair of simultaneous equations for the unknown functions h_1 and h_2, but, by eliminating h_1 between them, we can derive a single Fredholm integral equation of the second kind for h_2. Solving it, we can determine h_1 using Eq. (95).

The same procedure applied formally to the case in which $k(\xi) \neq 0$, $g_1 = g_3 = 0$, $|\alpha| < 1$ leads to the set of simultaneous equations

$$\begin{aligned} h_1 + E &= k_1 \qquad (x \in I_1) \\ h_2 + E &= -\begin{pmatrix} x, & a \\ a, & 0 \end{pmatrix} L_{\nu/2,\alpha} k_1 + \begin{pmatrix} x \\ a \end{pmatrix} I^{-1}_{\nu/2,\alpha} f_2 \qquad (x \in I_2) \\ h_1 &= -\begin{pmatrix} a, & b \\ x, & a \end{pmatrix} M_{\nu/2,-\alpha} h_2 \qquad (x \in I_1) \end{aligned} \tag{96}$$

where

$$\begin{aligned} E(x) &\equiv S_{\nu/2-\alpha,\alpha} k S_{\nu/2,-\alpha} h \\ &= 2^\alpha x^{-\alpha} \int_0^\infty t^{1-\alpha} k(t) 2^{-\alpha} t^\alpha J_{\nu-\alpha}(xt)\, dt \\ &\quad \left(\int_0^a u^{1+\alpha} h_1(u) J_{\nu-\alpha}(tu)\, du + \int_a^b u^{1+\alpha} h_2(u) J_{\nu-\alpha}(tu)\, du \right. \\ &\quad \left. + \int_b^\infty u^{1+\alpha} h_3(u) J_{\nu-\alpha}(tu)\, du \right) \end{aligned}$$

and inverting the order of integration in each of the three repeated integrals, we have

$$E(x) = x^{-\alpha} \int_0^\infty u^{1+\alpha} K(x,u) h(u)\, du \tag{97}$$

where

$$K(x,u) = \int_0^\infty t k(t) J_{\nu-\alpha}(xt) J_{\nu-\alpha}(ut)\, dt \tag{98}$$

Thus, we have three equations with three unknowns h_1, h_2, k_1. As before $h_3 = 0$. Solving for them, the unknown functions f_1 and f_3 can be found by the formulas

$$\begin{aligned} f_1 &= \begin{pmatrix} x \\ 0 \end{pmatrix} I_{\nu,\alpha} k_1 \\ f &= I_{\nu/2,\alpha} h + S_{\nu/2-\alpha,2\alpha} k S_{\nu/2,-\alpha} h \end{aligned} \tag{99}$$

Equations (93) and (95) to (98) allow us to obtain the complete solution of the problem. [For further details readers are referred to the papers by Cooke (19,20,27,28)].

The Borodachev Trial Solution

Borodachev (31) developed a different trial solution to solve the triple integral equations (87). He argued as follows: Assume that the solution of the equations has the form

$$\psi(\xi) = S_{\beta,\gamma} h \tag{100}$$

Equations (88) for the case where $k(\xi) = 0$, may be reduced to the following form:

$$I_{\mu_1,\lambda_1} h = f, \qquad K_{\mu_2,\lambda_2} h = g$$

which occur when

$$S_{\nu/2-\alpha} S_{\beta,\gamma} = I_{\mu_1,\lambda_1}, \qquad S_{\nu/2,0} S_{\beta,\gamma} = K_{\mu_2,\lambda_2} \tag{101}$$

Using the third and fourth relations of Eqs. (49), we infer that

$$\begin{aligned} \beta + \gamma &= \frac{\nu}{2} - \alpha, \qquad \mu_1 = \beta, \qquad \lambda_1 = 2\alpha + \gamma \\ \beta &= \frac{\nu}{2}, \qquad \mu_2 = \frac{\nu}{2}, \qquad \lambda_2 = \gamma \end{aligned}$$

which yield

$$\begin{aligned} &\beta = \frac{\nu}{2}, \qquad \gamma = -\alpha, \qquad \mu_1 = \frac{\nu}{2}, \qquad \mu_2 = \frac{\nu}{2}, \\ &\lambda_1 = \alpha, \qquad \lambda_2 = -\alpha \end{aligned}$$

Thus, in this case Eq. (100) takes the form $\psi = S_{\nu/2,-\alpha} h$, that is, we have Sneddon's trial solution.

On the other hand, readers might note that Eqs. (88) can be reduced to the form

$$K_{\mu_3,\lambda_3} H = f, \qquad I_{\mu_4,\lambda_4} H = g \tag{102}$$

Carrying out calculations similar to the ones done, we have

$$\begin{aligned} &\beta = \frac{\nu}{2} + \alpha, \qquad \gamma = -\alpha, \qquad \mu_3 = \frac{\nu}{2} - \alpha, \qquad \mu_4 = \frac{\nu}{2} + \alpha \\ &\lambda_3 = \alpha, \qquad \lambda_4 = -\alpha \end{aligned} \tag{103}$$

Accordingly, in this case,

$$\psi = S_{\nu/2+\alpha,-\alpha} H \tag{104}$$

Equation (104) is called Borodachev's trial solution.

We will now use Borodachev's trial solution to reduce the triple integral equations in Eq. (88) to a Fredholm integral

equation of the second kind. Substituting Borodachev's trial solution in Eq. (104) into Eqs. (88), we obtain [see Eqs. (102) and (103)]

$$K_{\nu/2-\alpha,\alpha}H = f, \qquad I_{\nu/2+\alpha,-\alpha}H = g$$

whence

$$H = K^{-1}_{\nu/2-\alpha,\alpha}f, \qquad H = I^{-1}_{\nu/2+\alpha,-\alpha}g \tag{105}$$

As before, for the sake of simplicity, we consider the case where $g_1 = g_3 = 0$. Then writing Eq. (105) for each interval, we obtain

$$\begin{aligned}
H_1 &= \binom{x}{0} I^{-1}_{\nu/2+\alpha,-\alpha}g_1 = 0 \\
H_2 &= \binom{a}{0} I^{-1}_{\nu/2+\alpha,-\alpha}g_1 + \binom{x}{a} I^{-1}_{\nu/2+\alpha,-\alpha}g_2 = \binom{x}{a} I^{-1}_{\nu/2+\alpha,-\alpha}g_2 \\
H_3 &= \binom{a}{0} I^{-1}_{\nu/2+\alpha,-\alpha}g_1 + \binom{b}{a} I^{-1}_{\nu/2+\alpha,-\alpha}g_2 + \binom{x}{b} I^{-1}_{\nu/2+\alpha,-\alpha}g_3 \\
&= \binom{b}{a} I^{-1}_{\nu/2+\alpha,-\alpha}g_2 \\
H_3 &= \binom{\infty}{x} K^{-1}_{\nu/2-\alpha,\alpha}f_3 \\
H_2 &= \binom{\infty}{b} K^{-1}_{\nu/2-\alpha,\alpha}f_3 + \binom{b}{x} K^{-1}_{\nu/2-\alpha,\alpha}f_2
\end{aligned} \tag{106}$$

From the second and fourth formulas in Eqs. (106), we deduce that

$$g_2 = \binom{x}{a} I_{\nu/2+\alpha,-\alpha}H_2, \qquad f_3 = \binom{\infty}{x} K_{\nu/2-\alpha,\alpha}H_3 \tag{107}$$

Substituting Eq. (107) into the third and fifth equations in Eqs. (106) and making use of the operators L and M, we obtain the following system of equations:

$$\begin{aligned}
H_2 &= \binom{b}{x} K^{-1}_{\nu/2-\alpha,\alpha}f_2 - \begin{pmatrix} b, & \infty \\ x, & b \end{pmatrix} M_{\nu/2-\alpha,\alpha}H_3 \qquad (a < x < b) \\
H_3 &= -\begin{pmatrix} x, & b \\ b, & a \end{pmatrix} L_{\nu/2+\alpha,-\alpha}H_2 \qquad (b < x < \infty)
\end{aligned} \tag{108}$$

Using the definitions of the L and M operators, we see that the formulas in Eq. (108) constitute a pair of coupled integral equations, upon solving for which we can find the functions and H_2 and H_3, while $H_1 = 0$.

Putting the second formula of Eq. (108) into the first equation, we obtain a single integral equation of the second kind involving only H_2:

$$H_2(x) = \varphi(x) - \left(\frac{2}{\pi}\right)^2 \int_a^b K(x,y)H_2(y)\,dy \tag{109}$$

where

$$\begin{aligned}
\varphi(x) &= \binom{b}{x} K_{\nu/2,-\alpha}f_2 \\
K(x,y) &= \sin^2(\alpha\pi)\frac{x^\nu y^{1+2\alpha+\nu}}{(b^2-x^2)^\alpha(b^2-y^2)^\alpha}\int_0^\infty \frac{t^{1-2\nu-2\alpha}(t^2-b^2)^{2\alpha}}{(t^2-x^2)(t^2-y^2)}\,dt \\
&\qquad (-\tfrac{1}{2} < \alpha < 1)
\end{aligned} \tag{110}$$

An Example: An Electrified Annular Disk

To illustrate the application of Cooke's and Borodachev's solutions to the set of triple integral equations in Eq. (98), we consider the electrostatic field induced by an annular disk with internal and external radii a and b, respectively, the disk being charged to a potential equal to ϕ_0. The disk is assumed to lie in the plane $z = 0$.

The solution of the problem must satisfy Laplace's equation in Eq. (1) and the following boundary conditions:

$$\begin{aligned}
&\phi(r,0) = \phi_0, \qquad a < r < b \\
&\left.\frac{\partial\phi}{\partial z}\right|_{z=0} = 0, \quad 0 \le r < \alpha, \qquad b < r < \infty
\end{aligned} \tag{111}$$

Furthermore, the solution must satisfy the regularity condition and the edge conditions at the edges $r = a$ and $r = b$. As before, applying zeroth-order Hankel transform to the Eq. (1), it can be shown that the electrostatic potential is given by the equation

$$\phi(r,z) = \phi_0 \mathscr{H}_0[s^{-1}A(s)e^{-sz}; s \to r] \tag{112}$$

where $A(s)$ is an unknown function of s to be determined. Equation (112) automatically satisfies the radiation conditions.

Making use of the boundary conditions in Eq. (111), we obtain the following triple integral equations:

$$S_{-1/2,1}A(r) = f(r), \qquad S_{0,0}A(r) = g(r)$$

where $f_2(r) = 2r/\phi_0$, $g_1(r) = 0$, $g_3(r) = 0$. Following Sneddon's trial solution in Eq. (89), we obtain the following Fredholm integral equation of the second kind:

$$\frac{x^2-\epsilon^2}{x^2}F(x) = 1 - \left(\frac{2}{\pi}\right)^2 \int_\epsilon^1 K(x,y)F(y)\,dy \tag{113}$$

where

$$\begin{aligned}
x &= \frac{r}{a}, \qquad \epsilon = \frac{a}{b}, \qquad F(x) = h_2^*(xb) \\
K(x,y) &= \frac{1}{2(x^2-y^2)}\left(\frac{x^2-\epsilon^2}{x}\log\frac{x+\epsilon}{x-\epsilon} - \frac{y^2-\epsilon^2}{y}\log\frac{y+\epsilon}{y-\epsilon}\right)
\end{aligned}$$

On the other hand, making use of Borodachev's trial solution in Eq. (104), we obtain the following Fredholm integral equation:

$$\frac{1-x^2}{x^2}G(x) = 1 - \left(\frac{2}{\pi}\right)^2 \int_\epsilon^1 M(x,y)G(y)\,dy \tag{114}$$

where

$$G(x) = h_2^*(bx), \qquad h_2^*(r) = \frac{\sqrt{\pi} r^2}{2\phi_0 \sqrt{b^2 - r^2}} h_2(r)$$
$$M(x,y) = \frac{1}{2(x^2 - y^2)} \left(\frac{1-y^2}{y} \log \frac{1+y}{1-y} - \frac{1-x^2}{x} \log \frac{1+x}{1-x} \right) \tag{115}$$

The surface charge density at any point of the disk is

$$q = \frac{-1}{4\pi} \left(\frac{\partial \phi}{\partial z} \right)_{z=0} = \frac{\phi_0}{4\pi} g_2(r)$$
$$= \begin{cases} \dfrac{1}{2\pi^2 r} \dfrac{d}{dr} \displaystyle\int_a^r \sqrt{\frac{b^2 - u^2}{r^2 - u^2}} h_2^*(u)\, du, & a < r < b \\ \dfrac{b\phi_0}{2\pi^2 r} \dfrac{d}{dr} \displaystyle\int_\epsilon^{r/b} \sqrt{\frac{1-y^2}{r^2/b^2 - y^2}}\, G(y)\, dy, & a < r < b \end{cases} \tag{116}$$

Thus, the charge density at any point of the disk can be calculated once the integral equation in Eq. (114) is solved.

Considering both sides of the disk, the total charge is

$$Q = 4\pi \int_a^b r q(r,0)\, dr = \frac{2\phi_0 b}{\pi \gamma}, \qquad \gamma^{-1} = \int_\epsilon^1 G(y)\, dy$$

whence

$$\phi_0 = \frac{\pi Q \gamma}{2b}$$

so that formula in Eq. (116) takes the form

$$q(r,0) = \frac{\gamma Q}{2\pi r} \frac{d}{dr} \int_\epsilon^{r/b} \sqrt{\frac{1-y^2}{r^2/b^2 - y^2}}\, G(y)\, dy, \qquad a < r < b \tag{117}$$

Of great interest is to find the asymptotic representation of the charge density $q(r, 0)$ as $r \to a + 0$ in the sense of Erdelyi, that is, the first term in the asymptotic expansion of $q(r, 0)$ as $r \to a + 0$. By letting $r \to a + 0$ in Eq. (117), we obtain

$$q(r,0) \approx \frac{Q \omega_a(\epsilon)}{2\sqrt{2}\pi b^2} \left(\frac{r}{b} - \epsilon \right)^{-1/2}, \qquad r \to a + 0 \tag{118}$$

where

$$\omega_a(\epsilon) = \frac{\gamma}{\epsilon} \sqrt{\frac{1-\epsilon^2}{\epsilon}}\, G(\epsilon) \tag{119}$$

Performing similar analyses on the Sneddon's trial solution, it can be shown that the surface charge density exhibits the following behavior as the outer contour of the disk is approached:

$$q(r,0) = \frac{Q \omega_b(\epsilon)}{\sqrt{2}\pi b^2} \left(1 - \frac{r}{b} \right)^{-1/2}, \qquad r \to b - 0 \tag{120}$$

where

$$\omega_b(\epsilon) = \gamma \sqrt{1 - \epsilon^2} F(1) \tag{121}$$

Equations (118) to (121) show that the surface charge density exhibits a square-root singularity as the inner and outer edges of the disk are approached. Thus, edge conditions (Meixner's conditions) are satisfied.

Integral equations in Eqs. (113) and (114) admit closed-form solutions only in the special case where $\epsilon = 0$, that is, for the case of a circular disk:

$$F(x) = 1, \qquad G(x) = \frac{x}{\pi \sqrt{1 - x^2}} \log \frac{1+x}{1-x} \tag{122}$$

In the context of mathematically similar elastic contact problems, Borodachev (29) showed that the values of $G(x)$ do not differ practically in the range $0 \le \epsilon \le 0.5$. Therefore for this range, approximate values of the surface charge density can be calculated by using formulas in Eq. (122) while the integral equation in Eq. (113) can be solved to find the surface charge density for the range $0.5 < \epsilon < 1.0$.

Many other applications of the triple integral equations considered here to problems of electrostatics are given in Sneddon's book (8). It should be noted that using the same approach, it is possible to solve a wide variety of problems concerning diffraction of a plane electromagnetic wave by an annular disk and by a system of coaxial annular disks. Many examples of electromagnetic scattering by objects of different shapes are analyzed in the books by Bowman, Senior, and Uslenghi (30) and by Uslenghi (31).

QUADRUPLE INTEGRAL EQUATIONS INVOLVING HANKEL TRANSFORMS

We now use Cooke operators to reduce certain quadruple integral equations involving Hankel transforms to a Fredholm integral equation of the second kind or a system of those. The problem is to find a function $\psi(x)$ satisfying the equations

$$\begin{aligned} &S_{\nu/2-\alpha,2\alpha} \psi(x) = f_1(x), && x \in I_1 = \{x: 0 < x < \alpha\} \\ &S_{\nu/2-\beta,2\beta} \psi(x) = 0, && x \in I_2 = \{x: a < x < b\} \\ &S_{\nu/2-\alpha,2\alpha} \psi(x) = f_3(x), && x \in I_3 = \{x: b < x < c\} \\ &S_{\nu/2-\beta,2\beta} \psi(x) = 0, && x \in I_4 = \{x: c < x < \infty\} \end{aligned} \tag{123}$$

Taking a trial solution in the form

$$\psi(x) = S_{\nu/2+\beta,-\alpha-\beta} h(x)$$

and then using the third and fourth relations from Eqs. (49), we obtain

$$\begin{aligned} f(x) &\equiv S_{\nu/2-\alpha,2\alpha} \psi(x) = I_{\nu/2+\beta,\alpha-\beta} h(x) \\ g(x) &\equiv S_{\nu/2-\beta,2\beta} \psi(x) = K_{\nu/2-\beta,\beta-\alpha} h(x) \end{aligned} \tag{124}$$

whence

$$\begin{aligned} h(x) &= I^{-1}_{\nu/2+\beta,\alpha-\beta} f(x) \\ h(x) &= K_{\nu/2-\beta,\alpha-\beta} g(x) \end{aligned} \tag{125}$$

Writing out Eqs. (125) on $I_j (j = 1, \ldots, 4)$, we have

$$h_1(x) = \begin{pmatrix} x \\ 0 \end{pmatrix} I^{-1}_{\nu/2+\beta,\alpha-\beta} f_1 \qquad (x \in I_1)$$

$$h_2(x) = \begin{pmatrix} a \\ 0 \end{pmatrix} I_{\nu/2+\beta,\alpha-\beta} f_1 + \begin{pmatrix} x \\ a \end{pmatrix} I^{-1}_{\nu/2+\beta,\alpha-\beta} f_2 \qquad (x \in I_2)$$

$$h_3(x) = \begin{pmatrix} a \\ 0 \end{pmatrix} I^{-1}_{\nu/2+\beta,\alpha-\beta} f_1 + \begin{pmatrix} b \\ a \end{pmatrix} I^{-1}_{\nu/2+\beta,\alpha-\beta} f_2 + \begin{pmatrix} x \\ b \end{pmatrix} I^{-1}_{\nu/2+\beta,\alpha-\beta} f_3 \qquad (x \in I_3)$$

$$h_4(x) = \begin{pmatrix} a \\ 0 \end{pmatrix} I^{-1}_{\nu/2+\beta,\alpha-\beta} f_1 + \begin{pmatrix} b \\ a \end{pmatrix} I^{-1}_{\nu/2+\beta,\alpha-\beta} f_2 + \begin{pmatrix} c \\ b \end{pmatrix} I^{-1}_{\nu/2+\beta,\alpha-\beta} f_3 + \begin{pmatrix} \infty \\ c \end{pmatrix} I^{-1}_{\nu/2+\beta,\alpha-\beta} f_4 \qquad (x \in I_4) \tag{126}$$

$$h_4(x) = \begin{pmatrix} \infty \\ x \end{pmatrix} K^{-1}_{\nu/2-\beta,\alpha-\beta} g_4 = 0 \qquad (x \in I_4)$$

$$h_3(x) = \begin{pmatrix} c \\ x \end{pmatrix} K^{-1}_{\nu/2-\beta,\alpha-\beta} g_3 \qquad (x \in I_3)$$

$$h_2(x) = \begin{pmatrix} c \\ b \end{pmatrix} K^{-1}_{\nu/2-\beta,\alpha-\beta} g_3 \qquad (x \in I_2)$$

$$h_1(x) = \begin{pmatrix} c \\ b \end{pmatrix} K^{-1}_{\nu/2-\beta,\alpha-\beta} g_3 + \begin{pmatrix} a \\ x \end{pmatrix} K^{-1}_{\nu/2-\beta,\alpha-\beta} g_1 \qquad (x \in I_1)$$

From sixth equation of Eqs. (126), we have

$$g_3 = \begin{pmatrix} c \\ b \end{pmatrix} K_{\nu/2-\beta,\beta-\alpha} h_3$$

which upon substitution into the seventh equation of Eq. (126) yields

$$h_2(x) = -\begin{pmatrix} b, & c \\ x, & b \end{pmatrix} M_{\nu/2-\beta,\beta-\alpha} h_3(x) \tag{127}$$

Writing Eq. (124) on I_3, we obtain

$$f_3(x) = \begin{pmatrix} a \\ 0 \end{pmatrix} I_{\nu/2+\beta,\alpha-\beta} h_1 + \begin{pmatrix} b \\ a \end{pmatrix} I_{\nu/2+\beta,\beta-\alpha} h_2 + \begin{pmatrix} x \\ b \end{pmatrix} I_{\nu/2+\beta,\beta-\alpha} h_3 \tag{128}$$

Applying the operator

$$\begin{pmatrix} x \\ b \end{pmatrix} I^{-1}_{\nu/2+\beta,\beta-\alpha}$$

to both sides of Eq. (128), we obtain

$$h_3(x) = \Lambda(x) + \begin{pmatrix} x, & b \\ b, & a \end{pmatrix} L_{\nu/2+\beta,\alpha-\beta} h_2(x) \tag{129}$$

where $\Lambda(x)$ is the known function given by

$$\Lambda(x) = \begin{pmatrix} x \\ b \end{pmatrix} I^{-1}_{\nu/2+\beta,\alpha-\beta} f_3(x) - \begin{pmatrix} x, & a \\ b, & a \end{pmatrix} L_{\nu/2+\beta,\alpha-\beta} h_2(x) \tag{130}$$

Equations (127) and (129) constitute a pair of coupled integral equations for the determination of the unknown functions h_2 and h_3, but eliminating h_2, we obtain a single Fredholm equation of the second kind, namely,

$$h_3(x) + \mu \int_b^c K(x, x_0) h_3(x_0)\, dx_0 = \Lambda(x) \qquad (b < x < c) \tag{131}$$

where $\mu = (4/\pi^2) \sin^2[\pi(\alpha - \beta)]$ and the kernel is given by the equation

$$K(x, x_0) = x^{-\nu-2\beta}(x^2 - b^2)^{\beta-\alpha}(x_0^2 - b^2)^{\beta-\alpha} x_0^{2\alpha-\nu+1} \times \int_a^b \frac{(b^2 - y^2)^{2\alpha-2\beta} y^{2\nu-2\alpha+2\beta+1}}{(x_0^2 - y^2)(x^2 - y^2)}\, dy \qquad (b < x, x_0 < c) \tag{132}$$

Further details can be found in the article by Sneddon (21) and the references therein.

Quadruple integral equations of the type in Eq. (123) arise in many boundary-value problems of mathematical physics. For instance, the electrostatic problem of three coplanar circular disks charged to a uniform potential can be reduced to this kind of quadruple integral equations.

Finally, it should be noted that a new set of particular solutions can be derived for the quadruple integral equations of the type in Eq. (123) analogous to Borodachev's trial solution for the triple integral equations in Eq. (88), by assuming

$$\psi(x) = S_{\nu/2+\alpha,-\alpha-\beta} H(x) \tag{133}$$

Upon substituting Eq. (133) into Eqs. (123), we obtain

$$f(x) = K_{\nu/2-\alpha,\alpha-\beta} H(x)$$
$$g(x) = I_{\nu/2+\alpha,-\alpha+\beta} H(x)$$

MISCELLANEOUS

1. There is a generalization of the Hankel integral theorem in Eq. (12), known as Weber's integral [see Titchmarsh (32)]

$$f(r) = \int_0^\infty \frac{\varphi_s(r) s\, ds}{J_\nu^2(sa) + Y_\nu^2(sa)} \int_a^\infty r_0 f(r_0) \varphi_s(r_0)\, dr_0 \qquad a < r < \infty \tag{134}$$

involving the linear combination

$$\varphi_s(r) = J_\nu(sa) Y_\nu(sr) - Y_\nu(sa) J_\nu(sr) \tag{135}$$

of Bessel functions of the first and second kinds ($\nu > -\frac{1}{2}$). A sufficient condition for the validity of the Eq. (133) is that $f(r)$ be piecewise continuous and of bounded variation in every finite subinterval $[\alpha, \beta]$, where $a < \alpha < \beta < \infty$, and the integral

$$\int_a^\infty \sqrt{r} |f(r)|\, dr < \infty$$

It should be noted that Weber's integral reduces to Hankel's integral in the limit as $a \to 0$. Derivation of equations in Eqs. (134) and (135) is given in the famous book

by Titchmarsh (32). Properties of Weber's transformation are also similar to those derived for Hankel transforms. Weber's transform is suited for solving equations of the form in Eq. (1) for domains with an excluded circular region. Below we illustrate the use of Weber's integral by one example.

Example. A cylindrical hole of radius a is drilled in an infinite body, and the walls of the hole are maintained at a temperature T_0 starting from the time $t = 0$. It is required to determine the temperature distribution in the body assuming that the initial temperature is zero.

The two-dimensional temperature distribution in the body is governed by the heat-conduction equation

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right) = \frac{\partial T}{\partial t}, \qquad a < r < \infty \tag{136}$$

satisfying the initial condition $T|_{t=0} = 0$ and the boundary and radiation conditions

$$T|_{r=a} = T_0, \qquad T|_{r\to\infty} \to 0$$

Multiplying both sides of Eq. (136) by $r\varphi_s(r)$ and integrating the resulting expression from a to ∞, we obtain

$$\frac{d\tilde{T}}{dt} - s^2\tilde{T} = \frac{2T_0}{\pi} \tag{137}$$

where

$$\tilde{T}(s,t) = \int_a^\infty r\tilde{T}(r,t)\varphi_s(r)\,dr$$

may be called the *Weber transform* of zeroth order of the function $T(r, t)$. In deriving Eq. (137), use has been made of the relations

$$\varphi_s(a) = 0, \qquad \varphi_s'(a) = \frac{2}{\pi a}$$

The solution of Eq. (137) satisfying the boundary condition $\tilde{T}|_{t=0}$ is

$$\tilde{T}(s,t) = \frac{2T_0}{\pi s^2}(1 - e^{-s^2t})$$

Now, using Weber's inversion, we finally obtain the following formula for the temperature evolution in the body:

$$T(r,t) = \frac{2T_0}{\pi}\int_0^\infty \frac{(1-e^{-s^2t})\varphi_s(r)\,ds}{s[J_0^2(sa) + Y_0^2(sa)]} \tag{138}$$

Many other practical applications of Weber's integral are given in the book by Lebedev, Skalskaya, and Ufliand (9).

2. In applications of Hankel transforms to many physical problems, integrals of the following form are encountered:

$$\int_0^\infty s\tilde{f}_n(s)\tilde{F}_m(s)J_{m+n}(sr)\,ds \tag{139}$$

The need to evaluate such integrals arises in connection with the desire of transforming the solution for the physical quantities given in the space of Hankel transform domain into the physical space.

Using Parseval's relation in Eq. (39), we reduce the integral in Eq. (139) to the form

$$\int_0^\infty s\tilde{f}_n(s)\tilde{F}_m(s)J_{m+n}(sr)\,ds = \int_0^\infty r_0 f(r_0)\Phi(r_0)\,dr_0 \tag{140}$$

where

$$\begin{aligned}\Phi(r_0) &= \mathsf{H}_n[\tilde{F}_m(s)J_{m+n}(sr); s\to r_0] \\ &= \int_0^\infty s\tilde{F}_m(s)J_{m+n}(sr)J_n(sr_0)\,dr_0\end{aligned} \tag{141}$$

For the product of Bessel functions in Eq. (141), we use Neumann's formula generalized by Rahman (14) and then interchanging the order of integration, we get

$$\begin{aligned}\Phi(r_0) = \frac{1}{\pi}\int_0^\pi \Bigg[&\cos(n\phi)T_m\left(\frac{r - r_0\cos\phi}{R}\right) \\ &+ \frac{r_0\sin n\phi\sin\phi}{R}U_{m-1}\left(\frac{r-r_0\cos\phi}{R}\right)\Bigg]F(R)\,d\phi\end{aligned} \tag{142}$$

In specific physical problems, however, the cases where $m = 0$ and $m = 1$, $n = 0$ are the most frequently encountered ones. In these cases, formula in Eq. (142) simplifies significantly. For instance, for $m = 0$, we have

$$\Phi(r_0) = \frac{1}{\pi}\int_0^\pi \cos(n\phi)F(R)\,d\phi$$

while for $m = 1$, $n = 0$, we have

$$\Phi(r_0) = \frac{1}{\pi}\int_0^\pi \frac{r - r_0\cos\phi}{R}F(R)\,d\phi$$

3. An efficient method of solving the integral equation (74) is based on representing the unknown function $h_1(x)$ in the form [Rahman (33)]

$$h_1(x) = x^{\nu-2\alpha}\sum_{n=0}^\infty a_n P_n^{\nu-\alpha,0}(1-2x^2) \tag{143}$$

where $P_n^{\nu-\alpha,0}(1 - 2x^2)$ is the Jacobi polynomial and a_n are the unknown expansion coefficients to be determined.

Putting the expansion in Eq. (143) into Eq. (73) and considering the orthogonality relationship for the Jacobi polynomials

$$\begin{aligned}\int_0^1 \frac{P_n^{\alpha,\beta}(1-2x^2)P_m^{\alpha,\beta}(1-2x^2)\,dx}{2^{-2-\alpha-\beta}x^{-1-2\alpha}(1-x^2)^{-\beta}} \\ = \frac{2^{\alpha+\beta+1}\Gamma(\alpha+n+1)\Gamma(\beta+n+1)}{n!\Gamma(n+\alpha+\beta+1)(\alpha+\beta+2n+1)}\,\delta_{mn}\end{aligned}$$

(δ_{mn} – Kronecker's delta)

we obtain the following infinite system of linear algebraic equations:

$$\frac{a_m}{2(1+\nu-\alpha+2m)}+\sum_{n=0}^{\infty}a_nK_{mn}=r_m \qquad (m=0,1,2,\ldots,\infty) \quad (144)$$

where

$$K_{mn}=\int_0^{\infty}t^{-1}k(t)J_{1+\nu-\alpha+2m}(t)J_{1+\nu-\alpha+2n}(t)\,dt$$
$$r_m=\int_0^1 x^{1+\alpha}r(x)P_m^{\nu-\alpha,0}(1-2x^2)\,dx \quad (145)$$

A key result that was used to obtain Eqs. (144) and (145) is the following integral [Rahman (34)]:

$$S_{a,-a}[(1-x^2)^{-b}P_n^{a,-b}(1-2x^2)\}=\frac{\Gamma(1-b+n)}{2^{a+b}n!x^{1-a-b}}J_{1+a-b+2n}(x)$$

The infinite system in Eq. (144) can be solved by truncation for the unknown expansion coefficients a_n.

In boundary-value problems, often it is often the case that the quantity $g_1(x)$ is of prime importance. For instance, in the charged disk problems, the function $g_1(x)$ is directly proportional to the surface charge density $q(r, 0)$ $(0 \le r < a)$, which, in turn, is essential for finding the capacitance. Rahman (33) showed that with the representation in Eq. (143), the function g_1 is given by

$$g_1(x)=x^{\nu}\sum_{n=0}^{\infty}a_n\frac{n!}{\Gamma(1-\alpha+n)}(1-x^2)^{-b}P_n^{\nu,-\alpha}(1-2x^2)$$

This method of solution is certainly preferrable to that based on using the numerical quadrature, because it bypasses the arduous job of evaluating Abel integrals numerically.

Furthermore, it was shown [Rahman (34)] that the following relation holds:

$$\binom{1}{x}K_{\nu/2,-\alpha}\left(\frac{x^{\nu-2\alpha}}{(1-x^2)^b}P_n^{\nu-\alpha,-b}(1-2x^2)\right)=\frac{\Gamma(1-b+n)}{\Gamma(1-b-\alpha+n)}x^{\nu}(1-x^2)^{-b-\alpha}\times P_n^{\nu,-b-\alpha}(1-2x^2) \quad (146)$$

Formula in Eq. (146) gives a class of *spectral relationship* for the operator $K_{\nu/2,-\alpha}$. It can be seen by writing out Eq. (146) in full that it gives a closed-form expression for a class of Abel integrals involving Jacobi polynomials. It can be used to a polynomial solution to Abel integral equations, which a number of boundary value problems of electrostatics can be reduced to.

4. The methods described in this article for solving dual integral equations are also applicable to a system of those of the form

$$S_{\mu_i/2-\alpha,2\alpha}\sum_{j=1}^{n}c_{ij}\psi_j(x)=f_i(x), \qquad x\in I_1$$
$$S_{\nu_i/2-\beta,2\beta}\psi_i(x)=g_i(x), \qquad x\in I_2$$

By a systematic use of the properties of Erdelyi–Kober operators Lowndes was able to show that the problem of solving a system of simultaneous equations of this type can be reduced to that of solving a system of simultaneous integral equations. Details of these results can be found in Sneddon's book (8). To the best of the writer's knowledge, generalization of these results has not yet been attempted for the case of simultaneous triple and quadruple integral equations.

5. The theory of Hankel transforms can also be extended to generalised functions or distributions via embedding theory or adjoint method. Interested readers are referred to consult the books by Zayed (5), Zemanian (34,35) and Brychkov and Prudnikov (36).

COMPENDIUM OF BASIC FORMULAS

For the sake of convenience of the readers, below we give a compendium of the basic formulas that are of frequent use in applications.

Definition of Hankel Transforms.

$$\tilde{f}_{\nu}(s)=\mathscr{H}_{\nu}[f(r),r\to s]=\int_0^{\infty}rf(r)J_{\nu}(sr)\,dr$$
$$f(r)=\mathscr{H}_{\nu}[\tilde{f}_{\nu}(s),s\to r]=\int_0^{\infty}s\tilde{f}_{\nu}(s)J_{\nu}(sr)\,dr$$

Some Properties of Hankel Transforms.

$$\mathscr{H}_{-m}[f(r),r\to s]=(-1)^m\mathscr{H}_m[f(r),r\to s] \qquad (m=\pm1,\pm2,\ldots,\pm n,\ldots)$$
$$\mathscr{H}_{\nu}[f(ar),r\to s]=a^{-2}\mathscr{H}_{\nu}\left[f(r),r\to\frac{s}{a}\right]$$
$$\mathscr{H}_{\nu}[r^{-1}f(r),r\to s]=\frac{s}{2\nu}[\tilde{f}_{\nu-1}(s)+\tilde{f}_{\nu+1}(s)] \qquad (\nu\neq0)$$
$$\mathscr{H}_n[f(r-a)H(r-a),r\to s]=\sum_{m=-\infty}^{\infty}\alpha_m\tilde{f}_m(s)$$
$$\alpha_m=J_{n-m}(sa)+\frac{1}{2}as[(m+1)^{-1}J_{n-m-1}(sa)+(m-1)^{-1}J_{n-m+1}(sa)]$$
$$\mathscr{H}_{\nu}[\mathscr{B}_{\nu}f(r),r\to s]=-s^2\mathscr{H}_{\nu}[f(r),r\to s]$$
$$\mathscr{B}_{\nu}=\frac{d^2}{dr^2}+\frac{1}{r}\frac{d}{dr}-\frac{\nu^2}{r^2} \qquad (\nu=0,1,\ldots)$$
$$\mathscr{H}_{\nu}\left[r^{\nu-1}\frac{d}{dr}[r^{1-\nu}f(r)],r\to s\right]=-s\mathscr{H}_{\nu-1}[f(r),r\to s]$$

Parseval's Relation.

$$\int_0^{\infty}s\tilde{f}_{\nu}(s)\tilde{g}_{\nu}(s)\,ds=\int_0^{\infty}r_0f(r_0)g(r_0)\,dr_0$$

Modified Operator of Hankel Transform and Erdelyi-Kober Operators.

$$I_{\eta,0} = K_{\eta,0} = I$$
$$I_{\eta,\alpha} x^{2\beta} f(x) = x^{2\beta} I_{\eta+\beta,\alpha} f(x)$$
$$I_{\eta,\alpha} I_{\eta+\alpha,\beta} = I_{\eta,\alpha+\beta}$$
$$K_{\eta,\alpha} x^{2\beta} f(x) = x^{2\beta} K_{\eta-\beta,\alpha} f(x)$$
$$K_{\eta,\alpha} K_{\eta+\alpha,\beta} = K_{\eta,\alpha+\beta}$$
$$I_{\eta,-n} f(x) = x^{2n-2\eta-1} D_x^n x^{2\eta+1} f(x)$$
$$I_{\eta,\alpha} f(x) = x^{-2\eta-2n-1} D_x^n x^{2n+2\eta+2\alpha+1} I_{\eta,\alpha+n} f(x)$$
$$K_{\eta,-n} f(x) = (-1)^n x^{2\eta-1} D_x^n x^{2n-2\eta+1} f(x)$$
$$K_{\eta,\alpha} f(x) = (-1)^n x^{2\eta-1} D_x^n x^{2n-2\eta+1} f(x) K_{\eta-n,\alpha+n} f(x)$$
$$I_{\eta,\alpha}^{-1} = I_{\eta+\alpha,-\alpha}$$
$$K_{\eta,\alpha}^{-1} = K_{\eta+\alpha,-\alpha}$$
$$S_{\eta,\alpha} f(x) = 2^{\alpha} x^{-\alpha} H_{2\eta+\alpha}[t^{-\alpha} f(t), t \to x]$$
$$S_{\eta,\alpha}^{-1} = S_{\eta+\alpha,-\alpha}$$
$$S_{\eta,\alpha} f(x) = 2^{-\lambda} x^{\lambda} S_{\eta\lambda/2,\alpha+\lambda}[x^{\lambda} f(x)]$$
$$I_{\eta+\alpha,\beta} S_{\eta,\alpha} = S_{\eta,\alpha+\beta}$$
$$K_{\eta,\alpha} S_{\eta+\alpha,\beta} = S_{\eta,\alpha+\beta}$$
$$S_{\eta+\alpha,\beta} S_{\eta,\alpha} = I_{\eta,\alpha+\beta}$$
$$S_{\eta,\alpha} S_{\eta+\alpha,\beta} = K_{\eta,\alpha+\beta}$$
$$S_{\eta+\alpha,\beta} I_{\eta,\alpha} = S_{\eta,\alpha+\beta}$$
$$S_{\eta,\alpha} K_{\eta+\alpha,\beta} = S_{\eta,\alpha+\beta}$$

Some Beltrami-Type Relations of Common Occurence.

$$\mathscr{H}_0[s^{-1}\tilde{f}_0(s), s \to r] = \frac{2}{\pi}\int_r^{\infty} \frac{dt}{\sqrt{t^2-r^2}} \int_0^t \frac{x f(x)dx}{\sqrt{t^2-x^2}}$$
$$\mathscr{H}_0[s\tilde{f}_0(s), s \to r] = \frac{-2}{\pi r}\frac{d}{dr}\int_r^{\infty} \frac{t\,dt}{\sqrt{t^2-r^2}} \frac{d}{dt}\int_0^t \frac{x f(x)dx}{\sqrt{t^2-x^2}}$$

Some Useful Relations.

$$\int_0^{\infty} s\tilde{f}_n(s)\tilde{F}_m(s) J_{m+n}(sr)\,ds = \int_0^{\infty} r_0 f(r_0)\Phi(r_0)\,dr_0$$

$$\Phi(r_0) = \frac{1}{\pi}\int_0^{\pi} \left[\cos(n\phi) T_m\left(\frac{r-r_0\cos\phi}{R}\right) + \frac{r_0 \sin n\phi \sin\phi}{R} U_{m-1}\left(\frac{r-r_0\cos\phi}{R}\right)\right] F(R)\,d\phi$$

$$J_{m+n}(sr) J_n(sr_0) = \frac{1}{\pi}\int_0^{\pi} \left[\cos(n\phi) T_m\left(\frac{r-r_0\cos\phi}{R}\right) + \frac{r_0 \sin n\phi \sin\phi}{R} U_{m-1}\left(\frac{r-r_0\cos\phi}{R}\right)\right] \times J_m(sR)\,d\phi$$

$$S_{\eta,-\eta}[(1-x^2)^{-\alpha} P_n^{\eta,-\alpha}(1-2x^2)] = \frac{\Gamma(1-\alpha+n)}{2^{\eta+\alpha} n! x^{1-\eta-\alpha}} J_{1+\eta-\alpha+2n}(x)$$

$$\binom{1}{x} K_{\nu/2,-\alpha}\left(\frac{x^{\nu-2\alpha}}{(1-x^2)^{\beta}} P_n^{\nu-\alpha,-\beta}(1-2x^2)\right) = \frac{\Gamma(1-\beta+n)}{\Gamma(1-\beta-\alpha+n)} x^{\nu}(1-x^2)^{-\alpha-\beta} \times P_n^{\nu,-\alpha-\beta}(1-2x^2)$$

SUGGESTED FURTHER READING

Readers interested in rigorous proofs of various aspects of the theory of Hankel transforms are referred to the books by Sneddon (1,2), Davies (3), Andrews and Shivamoggi (4), and Zayed (5) and the papers by Erdelyi (15) and Erdelyi and Kober (16). Many applications of the theory of Hankel transforms to physical problems are given in the books by Sneddon (8) and Lebedev, Skalskaya, and Ufliand (9). Fractional integrals and derivatives and their applications to dual, triple, and quadruple integral equations involving Hankel transforms are discussed at greater length in Sneddon (8,21), Cooke (19,20,27,28), Borodachev (29) and Samko, Kilbas, and Marichev (37). Extension of the theory of Hankel transforms to generalized functions or distributions is presented in the books by Zayed (5), Zemanian (34,35), and Brychkov and Prudnikov (36).

ACKNOWLEDGMENTS

The writer wishes to express his deepest gratitude to Professor Raj Mittra (Department of Electrical Engineering, Pennsylvania State University) for his kind response to queries with respect to some of the materials presented herein. He also gratefully acknowledges encouragement from his friends S. Rajeswaran and P. K. Jindal during the period of writing of the article.

BIBLIOGRAPHY

1. I. N. Sneddon, *Fourier Transforms,* New York: McGraw-Hill, 1951.
2. I. N. Sneddon, *The Use of Integral Transforms,* New York: McGraw-Hill, 1972.
3. B. Davies, *Integral Transforms and Their Applications,* Berlin: Springer-Verlag, 1978.
4. L. Andrews and B. Shivamoggi, *Integral Transforms for Engineers and Mathematicians,* New York: Macmillan, 1988.
5. A. I. Zayed, *Handbook of Function and Generalized Function Transformations,* Boca Raton, FL: CRC Press, 1996.
6. G. N. Watson, *A Treatise on the Theory of Bessel Functions,* London: Cambridge Univ. Press, 1944.
7. A. Erdelyi et al., *Tables of Integral Transforms,* New York: McGraw-Hill, 1954, Vol. II.
8. I. N. Sneddon, *Mixed Boundary Value Problems in Potential Theory,* Amsterdam: North Holland, 1965.
9. N. N. Lebedev, I. P. Skalskaya, and Ya. S. Ufliand, *Problems of Mathematical Physics,* translated from Russian, Englewood Cliffs, NJ: Prentice-Hall, 1965.
10. A. Erdelyi et al., *Higher Transcendental Functions,* New York: McGraw-Hill, 1953, Vol. II.
11. J. Jeans, *The Mathematical Theory of Electricity and Magnetism,* Cambridge, UK: Cambridge Univ. Press, 1908.
12. W. R. Smythe, *Static and Dynamic Electricity,* New York: McGraw-Hill, 1968.
13. I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series and Products,* New York: Academic Press, 1980.
14. M. Rahman, On a generalization of Neumann's formula for the product of two first kind Bessel functions of integral orders, *J. Appl. Math. Mech ZAMM,* **77** (2): 156–157, 1997.

15. A. Erdelyi, On fractional integration and its application to the theory of Hankel transforms, *Quart. J. Math. Oxford,* **11**: 293–303, 1940.
16. A. Erdelyi and H. Kober, Some remarks on Hankel transforms, *Quart. J. Math. Oxford,* **11**: 212–221, 1940.
17. H. Kober, On fractional integrals and derivatives, *Quart. J. Math. Oxford,* **11**: 193–211, 1940.
18. A. Erdelyi and I. N. Sneddon, Fractional integration and dual integral equations, *Can. J. Math.,* **14**: 685–693, 1962.
19. J. C. Cooke, The solution of triple integral equations in operational form, *Quart. J. Mech. Appl. Math.* **18**, part 3: 57–72, 1965.
20. J. C. Cooke, The solution of triple and quadruple integral equations and Fourier-Bessel series, *Quart. J. Mech. Appl. Math.* **25**: 247–263, 1972.
21. I. N. Sneddon, The use in mathematical physics of Erdelyi-Kober operators and some of their generalizations, in B. Ross (ed.), *Fractional Calculus and Its Applications,* Lecture Notes in Mathematics No. 457, Berlin: Springer-Verlag, 1975.
22. E. Beltrami, Sulla theoria delle funzione potenziali simmetriche, *Mem. Accad. Sci. Bologna,* **22** (IV): 462, 1881.
23. I. N. Sneddon, A relation involving Hankel transforms with applications to boundary value problems in potential theory, *J. Appl. Math. Mech.,* **14** (1): 33–40, 1965.
24. E. R. Love, The electrostatic field of two equal circular coaxial conducting disks, *Quart. J. Math. Mech.,* **11** (2): 428–451, 1949.
25. J. Meixner, The behavior of electromagnetic fields at edges, Inst. Math. Sci. Res. Rept. EM-72, New York University, 1954.
26. R. Mittra and S. W. Lee, *Analytical Techniques in the Theory of Guided Waves,* New York: Macmillan, 1971.
27. J. C. Cooke, Triple integral equations, *Quart. J. Mech. Appl. Math.,* **16**: 193–203, 1963.
28. J. C. Cooke, Some further triple integral equations, *Proc. Edinburgh Math. Soc.,* 303–316, 1963.
29. N. M. Borodachev, On a particular class of solutions of triple integral equations, *J. Appl. Math. Mech. PMM,* **40** (4): 605–611, 1976.
30. J. J. Bowman, T. B. A. Senior, and P. L. E. Uslenghi, *Electromagnetic and Acoustic Scattering by Simple Shapes,* New York: Wiley, 1970.
31. P. L. E. Uslenghi, *Electromagnetic Scattering,* New York: Academic Press, 1978.
32. E. C. Titchmarsh, *Eigenfunction Expansions Associated with Second-Order Differential Equations,* London: Oxford Univ. Press, 1946, vol. I.
33. M. Rahman, A note on the polynomial solution of a class of dual integral equations arising in mixed boundary value problems of elasticity, *J. Appl. Math. Phys. ZAMP,* **46** (1): 107–121, 1995.
34. A. H. Zemanian, *Distribution Theory and Transform Analysis,* New York: McGraw-Hill, 1965.
35. A. H. Zemanian, *Generalized Integral Transforms,* New York: Dover, 1987.
36. Yu. A. Brychkov and A. P. Prudnikov, *Integral Transforms of Generalized Functions,* translated from Russian, New York: Gordon and Breach, 1989.
37. S. G. Samko, A. A. Kilbas, and O. I. Marichev, *Fractional Integrals and Derivatives: Theory and Applications,* translated from Russian, Philadelphia: Gordon and Breach, 1993.

M. RAHMAN

HARMONIC ANALYSIS. See FOURIER TRANSFORM.

HARMONIC DISTORTION MEASUREMENT. See ELECTRIC DISTORTION MEASUREMENT.

HARMONIC FACTOR. See POWER SYSTEM HARMONICS.

Hartley Transforms

Standard Article

Dulal C. Kar[1] and V. V. Bapeswara Rao[2]

[1]Virginia Polytechnic Institute and State University, Blacksburg, VA

[2]North Dakota State University, Fargo, ND

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2424

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (162K)

Abstract

The sections in this article are

Definitions

Hartley Transforms of Energy Signals

Relationship Between the Hartley and the Fourier Transforms

Properties of the Hartley Transform

Hartley Transform of Power Signals

Discrete Hartley Transform

Multidimensional Hartley Transform

| | |

Copyright © 1999-2008 All Rights Reserved.

HARTLEY TRANSFORMS

Transform methods are used to determine the characteristics and to analyze the properties of a function describing a signal or a system that conveys information about or energy of a physical process. It is to be noted that transformation involves some sort of mathematical operation on the signal from one domain (time, space, or frequency) to another. Of all

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

known transform methods, the most popular and widely used is the Fourier transform used in scientific and engineering applications. However, the Fourier transform is generally a complex function. Along with the Fourier transform, many researchers have proposed many Fourier-like transform methods such as the cosine transform, the sine transform, the Hilbert transform, and the Hartley transform, all of which provide some alternative methods of analyzing signals and can lead to an efficient implementation in some specific applications.

The key advantage of the Hartley transform is that it is real for any real signal. It works very much like the Fourier transform. In fact, there exists a very simple relationship between the Fourier transform and the Hartley transform. As a result, wherever the Fourier transform is being used, the Hartley transform can be used as well. Because the Hartley transform is a real function, in some cases, it offers considerable advantages over the Fourier transform. For this reason, the Hartley transform has attracted the attention of many researchers, who have found many applications for it in science and engineering.

For digital signal processing (DSP), the discrete version of the Hartley transform exists. It has spurred research on fast algorithms on the discrete Hartley transform, which are also called the fast Hartley transforms (FHT). To some extent, the discrete Hartley transform requires less time and memory or hardware compared to the discrete Fourier transform.

DEFINITIONS

In 1942, Hartley introduced a Fourier-like transform (later known as the Hartley transform), which can be described for a function $h(t)$ as

$$H(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h(t)\mathrm{cas}(\omega t)\, dt \tag{1}$$

and the corresponding inverse transform can be described as

$$h(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} H(\omega)\mathrm{cas}(\omega t)\, d\omega \tag{2}$$

where the integral kernel function cas(ωt) is defined as

$$\mathrm{cas}(\omega t) = \cos(\omega t) + \sin(\omega t) \tag{3}$$

Some useful properties of the integral function, cas, are shown in Table 1. For the sake of convenience, a slightly different transform pair from the original Hartley transform pair is used in most literature and can be described for a function $h(t)$ as

$$H(f) = \int_{-\infty}^{\infty} h(t)\mathrm{cas}(2\pi ft)\, dt \tag{4}$$

and

$$h(t) = \int_{-\infty}^{\infty} H(f)\mathrm{cas}(2\pi ft)\, df \tag{5}$$

where the angular frequency ω and the radian frequency f are related by $\omega = 2\pi f$. Notice that the inverse integral function is the same as the direct integral function. It is evident from Eqs. (4) and (5) that the definition of the Hartley transform and the definition of the inverse Hartley transform are essentially the same. For our forthcoming discussions, we will refer to the second definition of the Hartley transform.

As is the case with the Fourier transform, the Hartley transform does not exist for all functions. In fact, the existence of the Hartley transform of a function is governed by the Dirichlet's conditions, which can be described for a function $h(t)$ as follows:

1. The function $h(t)$ must be absolutely integrable, that is

$$\int_{-\infty}^{\infty} |h(t)|\, dt < \infty \tag{6}$$

must hold.

2. The function $h(t)$ must have a finite number of maxima and minima and also must have a finite number of discontinuities in any interval.

Some useful signals, classified as power signals, do not satisfy the Dirichlet's conditions. However, their Hartley transforms can be expressed in terms of a special function called the Dirac delta function or the impulse function, which is used extensively for signal representation and analysis.

Table 1. Properties of the cas Function

1. $\mathrm{cas}(\alpha) = \cos(\alpha) + \sin(\alpha)$
2. $\mathrm{cas}(0) = 1$
3. $\int \mathrm{cas}(\alpha) = -\mathrm{cas}(-\alpha)$
4. $\dfrac{d}{d\alpha}\mathrm{cas}(\alpha) = \mathrm{cas}(-\alpha)$
5. $\mathrm{cas}^2(\alpha) + \mathrm{cas}^2(-\alpha) = 2$
6. $\cos(\alpha) = \frac{1}{2}[\mathrm{cas}(\alpha) + \mathrm{cas}(-\alpha)]$
7. $\sin(\alpha) = \frac{1}{2}[\mathrm{cas}(\alpha) - \mathrm{cas}(-\alpha)]$
8. $\mathrm{cas}(\alpha_1 + \alpha_2) = \cos(\alpha_1)\,\mathrm{cas}(\alpha_2) + \sin(\alpha_1)\,\mathrm{cas}(-\alpha_2)$
9. $\mathrm{cas}(2\alpha) - \mathrm{cas}(-2\alpha) = \mathrm{cas}^2(\alpha) - \mathrm{cas}^2(-\alpha)$
10. $\mathrm{cas}(2\alpha) = \mathrm{cas}^2(\alpha) + \mathrm{cas}(\alpha) - \mathrm{cas}(-\alpha) - 1$
11. $\mathrm{cas}(\theta) = \dfrac{1-j}{2}e^{j\theta} + \dfrac{1+j}{2}e^{-j\theta}$
12. $e^{j\theta} = \dfrac{1+j}{2}\mathrm{cas}(\theta) + \dfrac{1-j}{2}\mathrm{cas}(-\theta)$

HARTLEY TRANSFORMS OF ENERGY SIGNALS

Signals $h(t)$ for which $\int_{\infty}^{\infty} h^2(t)\, dt < \infty$ are classified as energy signals. Evidently, energy signals satisfy the Dirichlet's conditions. Here, we discuss the Hartley transform for some simple energy signals.

Rectangular Pulse

A rectangular pulse shown in Fig. 1, also called a gate function, is given by

$$\Pi(t/T) = \begin{cases} 1, & |t| < T/2 \\ 0, & \text{otherwise} \end{cases}$$

where T is the width of the pulse.

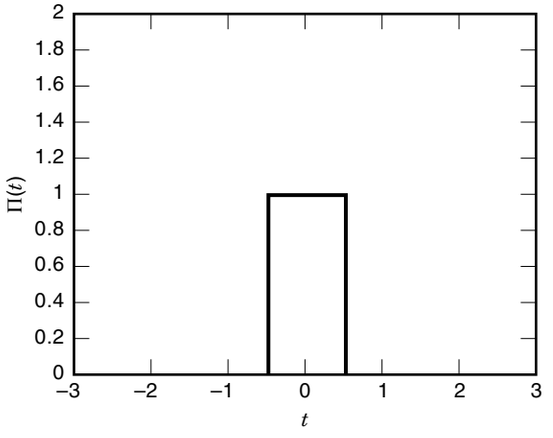


Figure 1. Rectangular pulse ($T = 1$).

From the definition of the Hartley transform as stated in Eq. (4), the Hartley transform of $\Pi(t/T)$ is given by

$$\begin{aligned} G(f) &= \int_{-\infty}^{\infty} \Pi(t/T)\text{cas}(2\pi ft)\,dt \\ &= \int_{-T/2}^{T/2} \text{cas}(2\pi ft)\,dt \\ &= \frac{1}{2\pi f}\left[\sin(2\pi ft) - \cos(2\pi ft)\right]_{T/2}^{T/2} \\ &= \frac{\sin(\pi fT)}{\pi f} = T\text{sinc}(fT) \end{aligned}$$

Note that the sinc function is defined as $\text{sinc}(x) = \sin(\pi x)/(\pi x)$. The plot of $G(f)$ is shown in Fig. 2. Notice that the Hartley transform of the gate function is the same as its Fourier transform. This is indeed true for all even functions. Notice from Fig. 2 that the first zero crossing of $G(f)$ occurs at frequency $f = 1/T$ and that as the pulse width T increases or decreases, the first zero crossing moves toward or away from the origin. In general, the shorter the duration of a signal, the wider its spectrum, and vice versa.

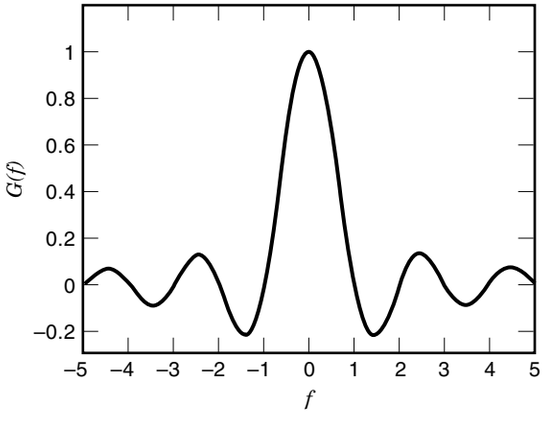


Figure 2. Hartley transform spectrum of a rectangular pulse ($T = 1$).

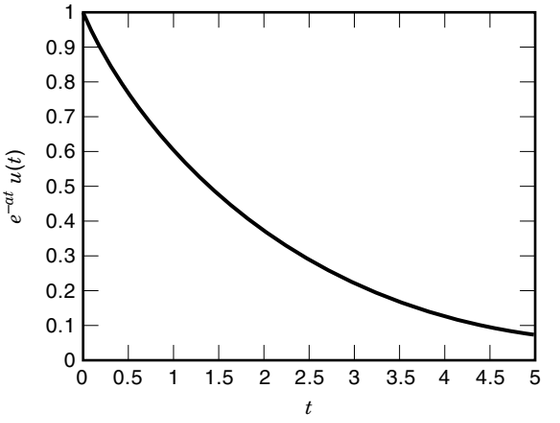


Figure 3. Exponential pulse ($a = 5$).

One-Sided Exponential Pulse

The analytic expression of a one-sided exponential pulse, shown in Fig. 3, is given by

$$h(t) = e^{-at}u(t)$$

where $u(t)$ is the unit step function and a is a constant that represents the rate of decay of the exponential pulse. The Hartley transform of $h(t)$ is $H(f)$ given by

$$\begin{aligned} H(f) &= \int_{-\infty}^{\infty} h(t)\text{cas}(2\pi ft)\,dt \\ &= \int_{-\infty}^{\infty} e^{-at}u(t)\text{cas}(2\pi ft)\,dt \\ &= \int_{0}^{\infty} e^{-at}\cos(2\pi ft\,dt + \int_{0}^{\infty} e^{-at}\sin(2\pi ft)\,dt \\ &= \frac{a + 2\pi f}{a^2 + 4\pi^2 f^2} \end{aligned}$$

The plot of $H(f)$ is shown in Fig. 4.

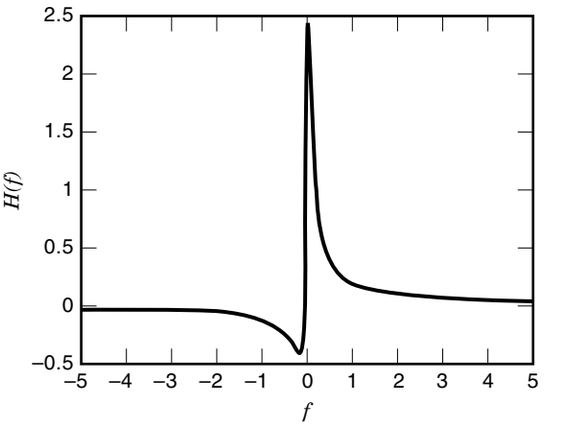


Figure 4. Hartley transform spectrum of an exponential pulse ($a = 5$).

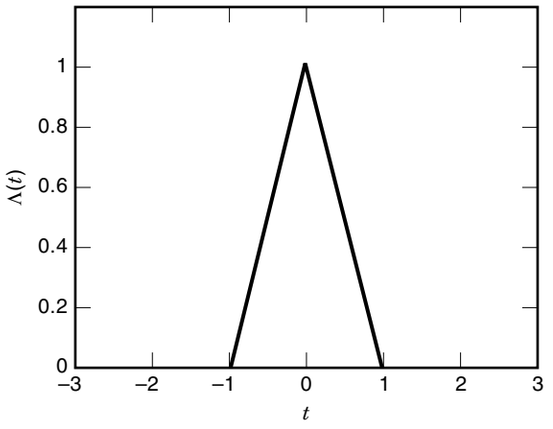


Figure 5. Triangular pulse ($T = 1$).

Triangular Pulse

The triangular pulse shown in Fig. 5 can be described as

$$\Lambda(t/T) = \begin{cases} 1 - |t|/T, & |t| < T \\ 0, & \text{otherwise} \end{cases}$$

where T represents the half of the width of the triangular pulse. The Hartley transform of the triangular pulse is

$$\begin{aligned} T(f) &= \int_{-\infty}^{\infty} \Lambda(t/T)\text{cas}(2\pi ft)\,dt \\ &= \int_{-T}^{0} (t/T + 1)\text{cas}(2\pi ft)\,dt + \int_{0}^{T} (-t/T + 1)\text{cas}(2\pi ft)\,dt \end{aligned}$$

After performing the integrations and simplifying, this expression yields the transform as

$$T(f) = T\text{sinc}^2(fT)$$

Figure 6 shows the plot of $T(f)$.

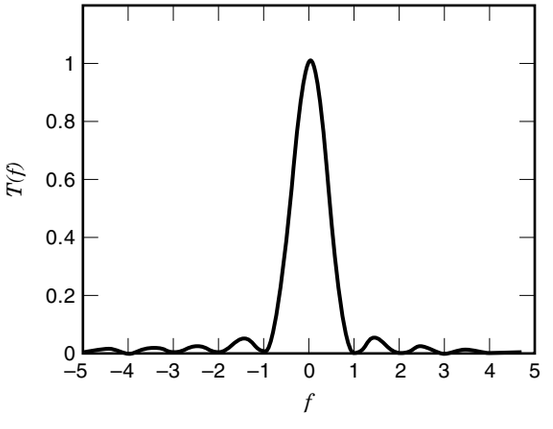


Figure 6. Hartley transform spectrum of a triangular pulse ($T = 1$).

Table 2. Hartley Transforms of Energy Signals

Function, $h(t)$	Hartley transform, $H(f)$
1. $e^{-at}u(t)$	$\dfrac{a + 2\pi f}{a^2 + 4\pi^2 f^2}$
2. $te^{-at}u(t)$	$\dfrac{a^2 + 4\pi af - 4\pi^2 f^2}{(a^2 + 4\pi^2 f^2)^2}$
3. $\prod(t/T) = \left(t + \dfrac{T}{2}\right) - u\left(t - \dfrac{T}{2}\right)$	$T\,\text{sinc}(fT)$
4. $\Lambda(t/T) = 1 - \dfrac{\lvert t\rvert}{T}, \lvert t\rvert < T$	$T\,\text{sinc}^2(fT)$
5. $e^{-a\lvert t\rvert}$	$\dfrac{2a}{a^2 + 4\pi^2 f^2}$
6. $\cos(2\pi f_0 t)\Pi(t/T)$	$\dfrac{T}{2}[\text{sinc}(T(f - f_0)) + \text{sinc}(T(f + f_0))]$
7. $e^{-a\lvert t\rvert}\cos(2\pi f_0 t)$	$\dfrac{a}{a^2 + 4\pi^2(f - f_0)^2} + \dfrac{a}{a^2 + 4\pi^2(f + f_0)^2}$
8. e^{-at^2}	$\sqrt{\dfrac{\pi}{a}}e^{\frac{-\pi^2 f^2}{a}}$
9. $\dfrac{1}{a^2 + t^2}$	$\dfrac{\pi}{a}e^{-2\pi a\lvert f\rvert}$

Table 2 lists the Hartley transforms of some frequently used energy signals including the rectangular pulse, the one-sided exponential pulse, and the triangular pulse. Hartley transforms of other energy signals can be obtained in the same way as discussed for the rectangular pulse, the one-sided exponential pulse, and the triangular pulse.

RELATIONSHIP BETWEEN THE HARTLEY AND THE FOURIER TRANSFORMS

Perhaps the most important property of the Hartley transform is its simple relationship with the Fourier transform. Note that the Fourier transform of a function $h(t)$ is defined by

$$F(f) = \int_{-\infty}^{\infty} h(t)e^{-j2\pi ft}\,dt \tag{7}$$

and the inverse Fourier transform is defined by

$$h(t) = \int_{-\infty}^{\infty} F(f)e^{j2\pi ft}\,df \tag{8}$$

From the Euler's relation that $e^{j\theta} = \cos(\theta) + j\sin(\theta)$ and from the relations of the sine and cosine functions with the cas function as listed in Table 1, the Fourier transform $F(f)$ can be expressed as

$$\begin{aligned} F(f) &= \int_{-\infty}^{\infty} h(t)\cos(2\pi ft)\,dt - j\int_{-\infty}^{\infty} h(t)\sin(2\pi ft)\,dt \\ &= \int_{-\infty}^{\infty} h(t)\frac{\text{cas}(2\pi ft) + \text{cas}(-2\pi ft)}{2}\,dt \\ &\quad - j\int_{\infty}^{\infty} h(t)\frac{\text{cas}(2\pi ft) - \text{cas}(-2\pi ft)}{2}\,dt \\ &= \frac{H(f) + H(-f)}{2} - j\frac{H(f) - H(-f)}{2} \end{aligned} \tag{9}$$

where $H(f)$ is the Hartley transform of the function $h(t)$. It is obvious from Eq. (9) that if the Hartley transform of a func-

tion is known, the Fourier transform of the function can readily be obtained. In fact, there are some situations in which knowing the amplitude and the phase characteristics of a function separately is important. In those situations, the Fourier transform of the function can be obtained quickly from the Hartley transform in order to determine the amplitude and the phase spectrums of the function.

In the same way, using the Euler's relation, the Hartley transform $H(f)$ of the function $h(t)$ can be expressed as

$$\begin{aligned} H(f) &= \int_{-\infty}^{\infty} h(t)[\cos(2\pi ft) + \sin(2\pi ft)]\,dt \\ &= \int_{-\infty}^{\infty} h(t)\frac{e^{j2\pi ft} + e^{-j2\pi ft}}{2}\,dt + \int_{-\infty}^{\infty} h(t)\frac{e^{j2\pi ft} - e^{-j2\pi ft}}{2j}\,dt \\ &= \frac{1+j}{2}\int_{-\infty}^{\infty} h(t)e^{-j2\pi ft}\,dt + \frac{1-j}{2}\int_{-\infty}^{\infty} h(t)e^{j2\pi ft}\,dt \\ &= \frac{1+j}{2}F(f) + \frac{1-j}{2}F(-f) \end{aligned} \tag{10}$$

Thus, the Hartley transform of a function can readily be obtained by using the relation in Eq. (10) if the Fourier transform of the function is known. Fortunately, for many functions, the Fourier transforms are known.

PROPERTIES OF THE HARTLEY TRANSFORM

The Hartley transform provides an alternative representation of a function $h(t)$ from one domain to another. Obtaining the Hartley transform of the inverse Hartley transform from the definition is a straightforward task. Note that the information content in the Hartley transform of $h(t)$ and the information content in the function $h(t)$ itself are the same. But, one form or the other provides a better insight into the physical aspects of the signal or the system associated with it. Certain useful manipulations or operations such as scaling, shifting, and integration on the function cause distinctive changes in its corresponding Hartley transform, and vice versa. Some useful properties of the Hartley transform related to such operations are summarized in Table 3. These properties can be obtained from the definition of the Hartley transform and/or from the relationship between the Hartley transform and the Fourier transform. Proofs for some important properties follow.

Table 3. Properties of the Hartley Transform

1.	Transformation	$h(t) \leftrightarrow H(f)$
2.	Linearity	$a_1h_1(t) + a_2h_2(t) \leftrightarrow a_1H_1(f) + a_2H_2(f)$
3.	Symmetry	$H(t) \leftrightarrow h(f)$
4.	Scaling	$h(t/a) \leftrightarrow \lvert a\rvert H(af)$
5.	Delay	$h(t - t_0) \leftrightarrow \cos(2\pi ft_0)H(f) + \sin(2\pi ft_0)H(-f)$
6.	Modulation	$\cos(2\pi f_0t)h(t) \leftrightarrow \frac{1}{2}[H(f - f_0) + H(f + f_0)]$
7.	Convolution	$h_1(t) \otimes h_2(t) \leftrightarrow \frac{1}{2}[H_1(f)H_2(f) + H_1(-f)H_2(f) + H_1(f)H_2(-f) - H_1(-f)H_2(-f)]$
8.	Time differentiation	$\frac{d}{dt}h(t) \leftrightarrow -2\pi fH(-f)$
9.	Time integration	$\int_{-\infty}^{t} h(\tau)\,d\tau \leftrightarrow \frac{H(-f)}{2\pi f} + \frac{H(0)\delta(f)}{2}$
10.	Reversal	$h(-t) \leftrightarrow H(-f)$
11.	Autocorrelation	$h(t) * h(t) \leftrightarrow \frac{1}{2}[H^2(f) + H^2(-f)]$
12.	Multiplication	$h_1(t)h_2(t) \leftrightarrow \frac{1}{2}[H_1(f) \otimes H_2(f) + H_1(-f) \otimes H_2(f) + H_1(f) \otimes H_2(-f) - H_1(-f) \otimes H_2(-f)]$

Delay. If $h(t) \leftrightarrow H(f)$ represents the Hartley transform pair then

$$h(t - t_0) \leftrightarrow \cos(2\pi ft_0)H(f) + \sin(2\pi ft_0)H(-f)$$

Proof. From the definition, the Hartley transform of $h(t - t_0)$ is

$$h(t - t_0) \leftrightarrow \int_{-\infty}^{\infty} h(t - t_0)[\cos(2\pi ft) + \sin(2\pi ft)]\,dt$$

After substituting $t' = t - t_0$, $dt' = dt$, and $t = t_0 + t'$ and simplifying, we can obtain

$$\begin{aligned} h(t - t_0) \leftrightarrow \int_{-\infty}^{\infty} h(t')[\cos(2\pi f(t' + t_0)) + \sin(2\pi f(t' + t_0))]\,dt' \\ = \cos(2\pi ft_0)H(f) + \sin(2\pi ft_0)H(-f) \end{aligned}$$

Convolution. If $h_1(t) \leftrightarrow H_1(f)$ and $h_2(t) \leftrightarrow H_2(f)$ are two Hartley transform pairs, then the Hartley transform of the convolution of $h_1(t)$ and $h_2(t)$ is

$$\begin{aligned} h_1(t) \otimes h_2(t) &= \int_{-\infty}^{\infty} h_1(\tau)h_2(t - \tau)\,d\tau \\ &\leftrightarrow \frac{1}{2}[H_1(f)H_2(f) + H_1(-f)H_2(f) \\ &\quad + H_1(f)H_2(-f) - H_1(-f)H_2(-f)] \end{aligned}$$

Proof. From the definition

$$\begin{aligned} h_1(t) \otimes h_2(t) &\leftrightarrow \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} h_1(\tau)h_2(t - \tau)\,d\tau\right]\cos(2\pi ft)\,dt \\ &= \int_{-\infty}^{\infty} h_1(\tau)\left[\int_{-\infty}^{\infty} h_2(t - \tau)\cos(2\pi ft)\,dt\right]d\tau \end{aligned}$$

Using the delay property and the relations between the cosine and the sine functions with the cas function as listed in Table 1,

$$\begin{aligned} &h_1(t) \otimes h_2(t) \\ &\quad \leftrightarrow \int_{-\infty}^{\infty} h_1(\tau)[\cos(2\pi f\tau)H_2(f) + \sin(2\pi f\tau)H_2(-f)]\,d\tau \\ &\quad = H_2(f)\int_{-\infty}^{\infty} h_1(\tau)\frac{\cos(2\pi f\tau) + \cos(-2\pi f\tau)}{2}\,d\tau \\ &\qquad + H_2(-f)\int_{-\infty}^{\infty} h_1(\tau)\frac{\cos(2\pi f\tau) - \cos(-2\pi f\tau)}{2}\,d\tau \\ &\quad = \frac{1}{2}[H_1(f)H_2(f) + H_1(-f)H_2(f) + H_1(f)H_2(-f) \\ &\qquad - H_1(-f)H_2(-f)] \end{aligned}$$

For some special cases, a simplified transform expression for the convolution of $h_1(t)$ and $h_2(t)$ can be obtained.

Case 1: If $h_1(t)$ or $h_2(t)$ is even or both are even, then

$$h_1(t) \otimes h_2(t) \leftrightarrow H_1(f)H_2(f)$$

Case 2. If $h_1(t)$ is odd, then

$$h_1(t) \otimes h_2(t) \leftrightarrow H_1(f)H_2(-f)$$

Case 3. If $h_2(t)$ is odd, then

$$h_1(t) \otimes h_2(t) \leftrightarrow H_1(-f)H_2(f)$$

Case 4. If both $h_1(t)$ and $h_2(t)$ are odd, then

$$h_1(t) \otimes h_2(t) \leftrightarrow -H_1(f)H_2(f)$$

Note that a function $h(t)$ is even if $h(t) = h(-t)$ or odd if $h(t) = -h(-t)$. Thus for the above-mentioned cases, the Hartley transform of the convolution of two functions can be obtained by a single multiplication from their transforms.

Power Spectrum. The power spectrum $P(f)$ of a function $h(t)$ is

$$P(f) = \frac{1}{2}[H^2(f) + H^2(-f)]$$

where $H(f)$ is the Hartley transform of $h(t)$.

Proof. If $F(f)$ is the fourier transform of $h(t)$, then

$$P(f) = F(f)F^*(f)$$

where $F^*(f)$ is the complex conjugate of $F(f)$. Hence, using relation in Eq. (9) between the Hartley and the Fourier transforms and simplifying, we can easily obtain

$$P(f) = \frac{1}{2}[H^2(f) + H^2(-f)]$$

Thus, finding the power spectrum of a signal from the Hartley transform is considerably easier than finding it from its Fourier transform becuse the process involves only real arithmetic and only two multiplications.

HARTLEY TRANSFORM OF POWER SIGNALS

So far we have considered only energy signals for the Hartley transform. These energy signals possess finite energy over the interval $(-\infty, \infty)$. Therefore, they are abolutely integrable and so satisfy the Dirichlet's conditions for the existence of $H(f)$. However, there is a class of signals, called power signals that are very useful but are not absolutely integrable. More rigorously, a power signal $f(t)$ has infinite energy but finite power such as the sine wave or the unit step function. This means that $f(t)$ does not satisfy the condition, $\int_{-\infty}^{\infty} f^2(t)\,dt < \infty$, but

$$P = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} f^2(t)\,dt < \infty$$

holds. It is possible to obtain the Hartley transform of these power signals if we allow impulse functions as part of the Hartley transform.

Impulse Function

The Hartley transform of the impulse function $\delta(t)$ is

$$H(f) = \int_{-\infty}^{\infty} \delta(t)\text{cas}(2\pi ft)\,dt = 1$$

Thus, we have the transform pair $\delta(t) \leftrightarrow 1$. Because of the symmetry in the Hartley and inverse Hartley transforms, we also have

$$1 \leftrightarrow \delta(f)$$

Thus the Hartley transform of unity is an impulse at the origin. Figure 7 shows the constant function and the corresponding Hartley transform.

The Signum Function

The signum function is defined as

$$\text{sgn}(t) = \begin{cases} -1, & t < 0 \\ 0, & t = 0 \\ 1, & t > 0 \end{cases}$$

Notice that if $h(t) \leftrightarrow H(f)$, then from the time differentiation property (see Table 3),

$$h^{(1)}(t) \leftrightarrow -2\pi fH(-f)$$

By differentiating the signum function, we obtain

$$\frac{d}{dt}\text{sgn}(t) = 2\delta(t)$$

If $H(f)$ denotes the Hartley transform of $\text{sgn}(t)$, then from the differentiation property listed in Table 3, we obtain

$$(-2\pi f)H(-f) = 2$$

which yields

$$H(-f) = \frac{1}{-\pi f}$$

Hence by replacing $-f$ by f, we obtain

$$H(f) = \frac{1}{\pi f}$$

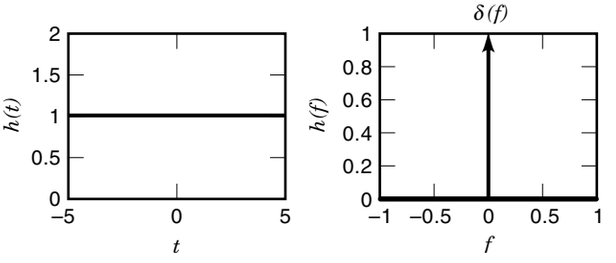


Figure 7. Constant function and its Hartley transform.

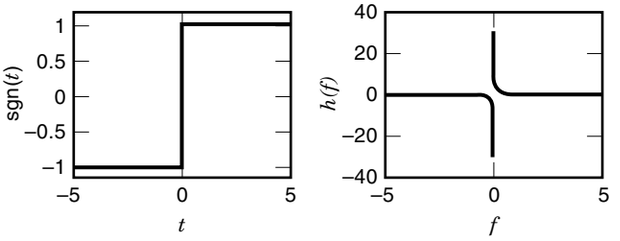


Figure 8. Signum function and its Hartley transform.

Figure 8 shows the plots of the signum function and its Hartley transform.

The Unit Step Function

The unit step function can be expressed in terms of the signum function as

$$u(t) = \frac{1}{2} + \frac{1}{2}\,\mathrm{sgn}(t)$$

Thus, the transform pair for $u(t)$ is

$$u(t) \leftrightarrow \frac{1}{2}\,\delta(f) + \frac{1}{2\pi f}$$

The Hartley transform of the useful power signals are summarized in Table 4 and can be derived independently as demonstrated previously or from their known Fourier transforms.

The foregoing discussions on the Hartley transform were based on the integral definition or the continuous-time definition of the Hartley transform. The integral definition allows us to study many analytical properties as well as to develop theory and explore properties for the discrete version of the Hartley transform. The discrete Hartley transform has found popularity in many real-time DSP applications.

DISCRETE HARTLEY TRANSFORM

The discrete version of the Hartley transform (DHT) of a data sequence $x(n)$, $n = 0, 1, 2, \ldots, N-1$, is described as

$$H(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} h(n)\mathrm{cas}(2\pi nk/N) \tag{11}$$

Table 4. Hartley Transforms of Power Signals

$h(t)$	$H(f)$
1. $\delta(t)$	1
2. 1	$\delta(f)$
3. $u(t)$	$\frac{1}{2}\delta(f) + \frac{1}{2\pi f}$
4. $\mathrm{sgn}(t)$	$\frac{1}{\pi f}$
5. $\cos w_0 t$	$\frac{1}{2}[\delta(f - f_0) + \delta(f + f_0)]$
6. $\sin w_0 t$	$\frac{1}{2}[\delta(f - f_0) - \delta(f + f_0)]$
7. $\sum_{k=-\infty}^{\infty} \delta(t - kT)$	$\frac{1}{2}\sum_{k=-\infty}^{\infty} \delta(f - k/T)$

for $k = 0, 1, 2, \ldots, N-1$. The corresponding inverse discrete Hartley transform is described as

$$h(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} H(k)\mathrm{cas}(2\pi nk/N) \tag{12}$$

The direct implementation of the discrete Hartley transform is computationally intensive when N is very large. Many fast Hartley transform algorithms can be found in Ref. 3 and are mostly based on the assumption that the number of data samples N is a power of 2. Most of the algorithms require either that data samples in the input sequence be sorted in bit-reversed order before processing or that the items in the transform sequence need to be sorted in bit-reversed order after processing.

The term bit-reversed ordering refers to finding the new index of an item by reversing the bits of the binary representation of the index of an element in given sequence and then placing the item in the new index position of the resulting sequence. For example, notice that three binary digits are required to index the data samples of a sequence $h(n) = \{h(0), h(1), h(2), h(3), h(4), h(5), h(6), h(7)\}$ where $N = 8$. If (n_2, n_1, n_0) represents the index of an item in $h(n)$, then the item should be copied in index position (n_0, n_1, n_2) in the bit-reversed sequence. Thus, after bit-reversed ordering, the resulting sequence will contain items in the order $h(0)$, $h(4)$, $h(2)$, $h(6)$, $h(1)$, $h(5)$, $h(3)$, $h(7)$.

Based on the Hou's fast Hartley transform algorithm described in Ref. 8, and algorithm for the fast computation of the Hartley transform is described here for the case when the transform size N is a power of two:

1. Perform permutation of data samples in sequence $x(n)$, $n = 0, 1, \ldots, N-1$ so that samples are in bit-reversed order.
2. Perform the following for $i = 2, 4, 8, 16, \ldots, N$ and for $j = 0, 2i, 3i, \ldots, N-i$:
   a. Copy $x(j)$, $x(j+1)$, $\ldots$, $x(j+i-1)$ to $g(0)$, $g(1)$, $\ldots$, $g(i-1)$, respectively.
   b. Perform the following for $k = 0, 1, \ldots, i/2 - 1$:

$$y(k) = g(k) + g(k + i/2)\cos(2\pi k/i) + g(i-k)\sin(2\pi k/i)$$

$$y(k + i/2) = g(k) - g(k + i/2)\cos(2\pi k/i) - g(i-k)\sin(2\pi k/i)$$

   c. Copy $y(0)$, $y(1)$, $\ldots$, $y(i-1)$ to $x(j)$, $x(j+1, \ldots, x(j+i-1)$, respectively.
3. Divide each item of $x(n)$ by the square root of N to get the Hartley transform sequence. The resulting sequence $x(n)$ where $n = 0, 1, \ldots, N-1$ holds the transform.

A complete C++ source code corresponding to this algorithm is given in Fig. 9. The source code is general enough to handle any case for $N = 2^m$, where m is a positive integer.

Compared to the discrete Fourier transform, the discrete Hartley transform involves only real arithmetic and provides a real transform sequence. As a result, it requires less arithmetic and memory or storage space for computational pur-

```
// This C++ implementation of the fast Hartley
// transform is based on the algorithm
// proposed by H.S. Hou, IEEE Transactions on Computers,
// Vol. C-36, No. 2, pp. 147-156, February 1987.

// The input data file ''INPUT.DAT'' must contain the sample
// size, N, as the first data in the file. Then, the data
// sequence, with items separated by whitespaces, must
// follow in the file.
// The sample size, N, must be an integer power of 2.
// The transform sequence along with the sample size is
// saved in the output file ''OUTPUT.DAT''.

#include <fstream.h>
#include <math.h>
const double pi = 3.1415927;

void main()
{
  int N;
  double sqn;
  double *x, *y, *g;
  int n;

  ifstream infile;
  ofstream outfile;

  void FHT(double [], double [], double [], const int);

  // Read Data from file INPUT.DAT
  infile.open(''INPUT.DAT'');  // open file input file
  infile >> N;                 // read size, N
  x = new double[N];           // setup input data array
                               // with N items

  for (n = 0; n < N; ++ n)     // read data sequence
    infile >> x[n];
  infile.close();              // close input file

  // Setup auxiliary arrays
  y = new double[N];
  g = new double[N];

  // Perform fast Hartley transform
  FHT(y, g, x, N);

  // Release auxliary arrays
  delete [] y;
  delete [] g;

  // Scan transform sequence
  sgn = sqrt(N);
  for (n = 0; n < N; ++ n)
  x[n] = x[n] / sqn;

  // Save transform sequence
  outfile.open(''OUTPUT.DAT'');
  outfile << N << endl;                    // save size
  for (n = 0; n < N; ++ n)
     outfile << ''   '' << x[n] << endl;   // save transform
                                           // sequence
  outfile.close();                         // close output
                                           // file

  // Release the data array
  delete [] x;
}

void FHT (double y[], double g[], double x[], const int N)
{
 int i, j, m;

 void bit_reverse(double [], double [], const int);
 void RHT(double [], double [], const int);

 bit_reverse(y, x, N);

 for (i = 2; i <= N; i = i * 2)
 {
  for (j = 0; j < N; j = j + i)
  {
   for (m = j; m < j + i; ++m)  // copy i data items
    g[m-j] = x[m];              // of x in g

   RHT(y, g, i);                // perform computation

   for (m = j; m < j + i; ++m)  // copy i calculated
                                // items
    x[m] = y[m-j];              // of y in x
  }
 }
}

// Recursive computation step
void RHT(double y[], double g[], const in M)
{
 int k;
 int L;
 double cfk, sfk;

 L = M >> 1;              // Divide M by 2 (L = M/2)

 y[0] = g[0] + g[L];
 y[L] = g[0] - g[L];

 for (k = 1; k < L; ++ k)
 {

  cfk = cos (2*pi*k/M);
  sfk = sin (2*pi*k/M);

  y[k] = g[k] + g[k + L] * cfk + g[M-k] * sfk;
  y[k + L] = g[k] - g[k + L] * cfk - g[M-k] * sfk;
 }
}

void bit_reverse(double y[], double x[], const int N)
{
  int i, incr, j;
  void arrange(double [], double [], int, int);

  // Bit reverse
  for (i = 1; i < N/2; i = i * 2)
  {
    incr = N/i;
    for (j = 0; j < N; j = j + incr)
      arrange(y, x, i, j + incr);
  }
}

void arrange(double y[], double x[], int first, int last)
{
 int mid, i, j;
 mid = (first + last)/2;
 for (i = first, j = first; i < mid; ++ i, j = j + 2)
   y[i] = x[j];

 for (i = mid, j = first + 1; i < last; ++ i, j = j + 2)
   y[i] = x[j];

 for (i = first; i < last; ++ i)
   x[i] = y[i];
}
```

Figure 9. A C++ program for fast Hartley transform.

poses. Also, for speed-critical, real-time applications, the hardware implementation of the discrete Hartley transform requires less hardware and is more efficient. These inherent advantages and the availability of the fast algorithms are the reasons why the Hartley transform is finding applications in many areas of science and engineering such as power engineering, data compression, speech coding, speech processing, image coding, image processing, optics, digital filtering, and biomedical engineering.

MULTIDIMENSIONAL HARTLEY TRANSFORM

The one-dimensional definition of the Hartley transform can easily be extended for multidimensional cases. Particularly, the two-dimensional Hartley transform for a function $h(x, y)$ can be described as

$$H(\alpha, \beta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y)\text{cas}[2\pi(\alpha x + \beta y)]\, dx\, dy \tag{13}$$

and the corresponding two-dimensional inverse Hartley transform can be described as

$$h(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(\alpha, \beta)\text{cas}[2\pi(\alpha x + \beta y)]\, d\alpha\, d\beta \tag{14}$$

Properties of the two-dimensional Hartley transform can be obtained in the same way as for the one-dimensional case. Two-dimensional Hartley transform techniques are used in image processing as well as in analog and digital optical image processing. The two-dimensional discrete Hartley transforms and its inverse are given by

$$H(u, v) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h(m, n) \cos 2\pi \left(\frac{mu}{M} + \frac{nv}{N}\right) \tag{15}$$

and

$$x(m, n) = \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} H(u, v) \cos 2\pi \left(\frac{mu}{M} + \frac{nv}{N}\right) \tag{16}$$

The fast Hartley transform algorithm described in the earlier section for the one-dimensional case also can be used for this two-dimensional case if we manipulate Eq. (15). The kernel cas function in Eq. (15) is not separable. However, from Table 1, we can establish that

$$\begin{aligned}\text{cas}(\alpha + \beta) &= \cos(\alpha)\text{cas}(\beta) + \sin(\alpha)\text{cas}(-\beta) \\ &= \frac{1}{2}[\text{cas}(\alpha)\text{cas}(\beta) + \text{cas}(-\alpha)\text{cas}(\beta) \\ &\quad + \text{cas}(\alpha)\text{cas}(-\beta) - \text{cas}(-\alpha)\text{cas}(-\beta)]\end{aligned}$$

Accordingly, the kernel function $\text{cas}(2\pi(mu/M + nv/N))$ in Eq. (15) can be expanded, and hence $H(u, v)$ can be expressed as

$$H(u, v) = \frac{1}{2}[F(u, v) + F(-u, v) + F(u, -v) - F(-u, -v)]$$

where

$$F(u, v) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) \cos 2\pi \left(\frac{mu}{M}\right) \cos 2\pi \left(\frac{nv}{N}\right) \tag{17}$$

Obviously, $F(u, v)$ is separable. Hence the one-dimensional fast transform algorithm can be used to compute $F(u, v)$, $F(-u, v)$, $F(u, v)$, and $F(-u, -v)$, and afterwards $H(u, v)$ can be obtained through simple addition and subtraction.

As stated earlier, the Hartley transform has many advantages over the Fourier transform, mainly because the Hartley transform is real for a real function or a real data sequence. It is computationally more efficient with respect to time and storage space. Additionally, for hardware implementation, the Hartley transform requires less hardware or VLSI area on the chip than the Fourier transform. An application that uses the Fourier transform can use the Hartley transform instead with some possible advantages.

Although the transform was introduced in 1942 by Hartley, it is R. N. Bracewell's 1983 work (2) and his other subsequent works that have brought attention to and popularized the Hartley transform. It has been found that the Hartley transform is very suitable for optical implementation because the transform representing the optical intensity is real for a real image (2). The Hartley transform has found many applications in science and engineering. The trend shows that the interest in the Hartley transform will continue in the future. The interest is evident from increasing number of publications on its theoretical development as well as on its applications every year.

BIBLIOGRAPHY

1. R. V. L. Hartley, A more symmetrical Fourier analysis applied to transmission problems, *Proc. IRE,* **30**: 144–150, 1942.
2. R. N. Bracewell, *The Hartley Transform,* New York: Oxford Univ. Press, 1986.
3. K. J. Olejniczak and G. T. Heydt (eds.), Special section on the Hartley transform, *Proc. IEEE,* **82**: 372–447, 1994.
4. R. N. Bracewell et al., Optical synthesis of Hartley transform, *Appl. Opt.,* **24**: 1401–1402, 1985.
5. A. D. Poularikas and S. Seely, *Signals and Systems,* 2nd ed., Malabar, FL: Krieger, 1994.
6. C. H. Paik and M. D. Fox, Fast Harltey transforms for image processing, *IEEE Trans. Med. Imaging,* **7**: 149–153, 1988.
7. Z. Wang, Fast algorithms for the discrete W transform and for the discrete Fourier transform, *IEEE Trans. Acoust. Speech Signal Process.,* **ASSP-32**: 803–816, 1984.
8. H. S. Hou, The fast Hartley transform algorithm, *IEEE Trans. Comput.,* **C-36**: 147–156, 1987.
9. S. Boussakta and A. G. J. Holt, Calculation of the discrete Hartley transform via the Fermat number transform using a VLSI chip, *Proc. Inst. Elec. Eng., part G: Electron. Circuits Syst.,* **135**: 101–103, 1988.
10. D. C. Kar and V. V. B. Rao, A CORDIC-based unified systolic architecture for sliding window applications of discrete transforms, *IEEE Trans. Acoust. Speech Signal Process.,* **ASSP-44**: 441–444, 1996.

DULAL C. KAR
Virginia Polytechnic Institute and State University

V. V. BAPESWARA RAO
North Dakota State University

HAZARDS, ELECTROLYTIC CELLS. See ELECTROLYTIC CELL SAFETY.

HDTV TRANSMITTERS. See TRANSMITTERS FOR DIGITAL TELEVISION.

HEALTH CARE BY TELEVISION. See TELEMEDICINE.

HEALTH CARE ENGINEERING. See CLINICAL ENGINEERING.

HEALTHCARE INFORMATION SYSTEMS. See MEDICAL INFORMATION SYSTEMS.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Hilbert Spaces
Standard Article
Frank Massey[1] and Daoqi Yang[2]
[1]University of Michigan-Dearborn, Dearborn, MI
[2]Wayne State University, Detroit, MI
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2425
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (223K)

Abstract

The sections in this article are

The Geometry of Hilbert Space

Linear Operators

The Equation Au = f

Spectral Theory and Evolution Equations

Applications to Quantum Mechanics

Other Methods

Acknowledgment

| | |

Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

HILBERT SPACES

Hilbert spaces are an essential tool for formulating and proving theories in many fields of science and engineering. For example, in quantum mechanics corresponding to each observable (a quantity that can be measured, such as position, momentum, or energy) is a self-adjoint operator in an appropriate Hilbert space. In wave scattering theory the convergence of various computational (finite difference or finite element) methods can be validated in suitable Hilbert spaces, which can guide one to choose the most suitable (efficient, accurate, robust) computational methods for a particular simulation.

A Hilbert space is a vector space equipped with an inner product, also called a scalar product, together with the requirement that a sequence $\{\boldsymbol{u}_n\}$ of vectors has a limit within the space whenever it has the property that the distance between two members of the sequence $\boldsymbol{u}_n$ and $\boldsymbol{u}_m$ can be made arbitrarily small when the indices n and m are large. One familiar example is the Euclidean plane, where the vectors can be represented by ordered pairs (x, y) of real numbers. In this case the inner product between (x_1, y_1) and (x_2, y_2) is just $x_1x_2 + y_1y_2$. The plane is an example of a finite dimensional Hilbert space. In applications of Hilbert space theory to differential equations, one uses infinite dimensional Hilbert spaces in which the "vectors" are functions defined on some fixed set. The inner product in a Hilbert space enables one to define the distance and angle between vectors, which, in turn, leads to the expansion of vectors in terms of special sets of vectors called orthonormal bases. When applied to Hilbert spaces of functions, this includes expansions of functions in terms of Fourier series and other well-known orthogonal sets of functions, such as wavelets.

In a Hilbert space setting an operator is a mapping that takes an input from a Hilbert space and maps it to an output in the same or another Hilbert space. Operators are generalizations of functions that map complex numbers to complex numbers. In finite dimensions each matrix defines an operator by multiplication. In Hilbert spaces of functions one is interested in differential and integral operators, such as the Laplace operator and Fourier transform. Many of the boundary value problems in classical field theory can be formulated as operator equations in a Hilbert space. For example, Poisson's equation [see Eq. (11)] can be considered as an operator equation in a Hilbert space of functions. Similarly, partial differential equations such as the wave equation [Eq. (18)] can be viewed as an ordinary differential equation in which the value of the unknown at each time is an element of a Hilbert space of functions.

An important class of operators are linear operators, which preserve vector addition and multiplication by complex numbers. They can be measured in magnitude via their norms, which are analogous to the absolute value of a complex number. Furthermore, certain problems involving linear operators can be solved via their spectra, which are a generalization of the set of eigenvalues of a matrix. In particular, this enables one to form functions of an operator. When applied to quantum mechanics, this allows one to find the operator corresponding to a function of an observable and to prove that an observable can assume values only in the spectrum of its corresponding operator.

The Geometry of Hilbert Space

The following is an overview of some important concepts and results in Hilbert space theory; for more details the reader can consult standard references in this area such as Blank, Exner, and Havlicek (1), Dunford and Schwartz (2), Kato (3), Naylor and Sell (4), Riesz and Sz.-Nagy (5), Schechter (6), von Neumann (7), and Yosida (8).

A *Hilbert space* H is a vector space equipped with an inner product such that H is complete in the sense defined in this section. An *inner product* is a function that assigns a complex number $(\boldsymbol{u}, \boldsymbol{v})$ to each pair of elements $\boldsymbol{u}, \boldsymbol{v}$ in H so that the following algebraic laws are satisfied:

$$(\boldsymbol{u}, \boldsymbol{u}) \geq 0 \text{ for all } \boldsymbol{u}, \text{ and } \quad (\boldsymbol{u}, \boldsymbol{u}) = 0 \text{ if and only if } \boldsymbol{u} = 0$$
$$(\boldsymbol{u} + \boldsymbol{v}, \boldsymbol{w}) = (\boldsymbol{u}, \boldsymbol{w}) + (\boldsymbol{v}, \boldsymbol{w}) \text{ for all } \boldsymbol{u}, \boldsymbol{v}, \text{ and } \boldsymbol{w}$$
$$(\alpha \boldsymbol{u}, \boldsymbol{v}) = \alpha(\boldsymbol{u}, \boldsymbol{v}) \text{ for all } \boldsymbol{u} \text{ and } \boldsymbol{v} \text{ in } H \text{ and all complex numbers } \alpha$$
$$(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{v}, \boldsymbol{u})^* \text{ for all } \boldsymbol{u} \text{ and } \boldsymbol{v} \text{ where } {}^* \text{ denotes complex conjugate}$$

H is called a *real* Hilbert space if "complex numbers" in the definition is replaced by "real numbers." A well-known example of a Hilbert space is the set of all n-tuples $\boldsymbol{x} = (x_1, \ldots, x_n)$ of complex numbers x_i with the inner product $(\boldsymbol{x}, \boldsymbol{y}) = x_1 y^*_1 + \cdots + x_n y^*_n$. This space is often denoted by C^n, and by R^n if the x_i are restricted to be real numbers. An infinite dimensional analog of this is l^2, which denotes the Hilbert space of all infinite sequences $\boldsymbol{x} = \{x_n\}$ of complex numbers such that $\Sigma_n |x_n|^2 < \infty$. The inner product is given by $(\boldsymbol{x}, \boldsymbol{y}) = \Sigma_n x_n y^*_n$. The L^2 spaces are continuous analogs of l^2 and are useful in problems where the independent variable is continuous instead of discrete, such as in differential equations and quantum mechanics. If T is a positive number, then the Hilbert space $L^2(0, T)$ consists of all square integrable functions $f(t)$ defined on the interval $0 < t < T$; that is, the $f(t)$ that satisfy $\int^T_0 |f(t)|^2 \, dt < \infty$. The inner product, defined by $(f, g) = \int^T_0 f(t) g(t)* \, dt$, is often associated with energy in many physical situations. For example, let $v(t)$ be the voltage dropped between the two ports of a circuit element and let $i(t)$ be the current through the element. Then $\int^T_0 v(t) i(t) \, dt = (v, i)$ is the energy lost in the element during the time interval $0 < t < T$. The space $L^2(a, b)$ is defined in a similar manner for any interval $a < t < b$ in the real line. More generally, if E is a set in R^n, then $L^2(E)$ is the space of all functions $\boldsymbol{u}(\boldsymbol{x})$ defined for $\boldsymbol{x} = (x_1, \ldots x_n)$ in E such that $\int_E |\boldsymbol{u}(\boldsymbol{x})|^2 \, dx < \infty$, where $d\boldsymbol{x} = dx_1 \ldots, dx_n$, and the integral is a multiple integral over E. Throughout the following E will denote an open set in R^n and B is the boundary of E. (E being *open* means that B is not included in E.) We shall just write L^2 for $L^2(E)$ when the set E is clear from the context.

A real Hilbert space H can naturally be extended to a complex Hilbert space consisting of all objects of the form $\boldsymbol{u} + i\boldsymbol{v}$, where $\boldsymbol{u}$ and $\boldsymbol{v}$ are in H. On the other hand, a complex Hilbert space can also be regarded as a real Hilbert space by introducing the real-valued inner product defined by $\text{Re}(\boldsymbol{u}, \boldsymbol{v})$, where Re denotes the operation of taking the real part of a complex number. In the following H can be either a real or complex Hilbert space except where noted otherwise. In the case of a real Hilbert space one may omit the complex operations such as $*$ and Re in the formulas.

An inner product provides a vector space with all the geometric structure of two- and three-dimensional Euclidean space. The *length* or *norm* $\|\boldsymbol{u}\|$ of a vector $\boldsymbol{u}$ in H is defined by $\|\boldsymbol{u}\| = \sqrt{(u,u)}$, while $\|\boldsymbol{u} - \boldsymbol{v}\|$ is the *distance* between two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. (Note, that the length $|\boldsymbol{x}| = (|x_1|^2 + \cdots + |x_n|^2)^{1/2}$ of a vector in C^n will be denoted by $|\boldsymbol{x}|$ instead of $\|\boldsymbol{x}\|$.) This length satisfies a number of the familiar properties of the length of two- and three-dimensional vectors, including the triangle inequality and parallelogram law. The *triangle inequality*, $\|\boldsymbol{u} + \boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$, says that the length of any side of a triangle is less than the sum of the lengths of the

other two sides. The *parallelogram law*,

$$\|\boldsymbol{u}+\boldsymbol{v}\|^2+\|\boldsymbol{u}-\boldsymbol{v}\|^2=2\|\boldsymbol{u}\|^2+2\|\boldsymbol{v}\|^2 \qquad (1)$$

expresses the fact that the sum of the squares of the lengths of the diagonals of a parallelogram is equal to the sum of the squares of the lengths of the sides. The parallelogram law is an easy consequence of the definition of length in terms of the inner product and the properties of the inner product. The triangle inequality follows from the easy-to-verify identity $\|\boldsymbol{u}+\boldsymbol{v}\|^2=\|\boldsymbol{u}\|^2+\|\boldsymbol{v}\|^2+2\,\mathrm{Re}(\boldsymbol{u},\boldsymbol{v})$ and the Schwarz inequality

$$|(\boldsymbol{u},\boldsymbol{v})|\le\|\boldsymbol{u}\|\,\|\boldsymbol{v}\| \qquad (2)$$

which says that the absolute value of the inner product of two elements is no bigger than the product of the lengths of the elements. See Ref. 8 (p. 40) for a proof of the Schwarz inequality.

An important technique in Hilbert space theory is the construction of solutions of various equations as the limit of a sequence or series. In a general Hilbert space there are two ways in which a vector $\boldsymbol{u}$ can be the limit of a sequence of vectors $\{\boldsymbol{u}_n\}$. One says $\{\boldsymbol{u}_n\}$ *converges* to $\boldsymbol{u}$ (or *converges strongly* to $\boldsymbol{u}$) if the distance from $\boldsymbol{u}_n$ to $\boldsymbol{u}$ approaches 0 as $n\to\infty$ (i.e., $\|\boldsymbol{u}_n-\boldsymbol{u}\|\to 0$ as $n\to\infty$; one often writes $\boldsymbol{u}_n\to\boldsymbol{u}$ when this holds). One says $\{\boldsymbol{u}_n\}$ *converges weakly* to $\boldsymbol{u}$ if for each $\boldsymbol{v}$ in H the sequence of complex numbers $(\boldsymbol{u}_n,\boldsymbol{v})$ converges to $(\boldsymbol{u},\boldsymbol{v})$. One writes $u_n\rightharpoonup u$ in this case. For a sequence of vectors in C^m both methods of convergence are the same; that is, $\{\boldsymbol{x}_n\}=\{(x_{n1},\ldots,x_{nm})\}$ converges both strongly and weakly to $\boldsymbol{x}=(x_1,\ldots,x_m)$ if for each j the sequence of complex numbers $\{x_{nj}\}$ converges to x_j. However, in infinite dimensions these two methods of convergence differ. It follows from the Schwarz inequality that if $\{\boldsymbol{u}_n\}$ converges strongly to $\boldsymbol{u}$, then it converges weakly to $\boldsymbol{u}$. However, a sequence can converge weakly but not strongly. Consider, for example, $H=L^2(0,T)$. Here a sequence of functions $\{f_n(t)\}$ converges strongly to a function $f(t)$ if $\int_0^T|f_n(t)-f(t)|^2\,dt\to 0$ as $n\to\infty$. This is also called convergence in the L^2 sense. On the other hand, $\{f_n(t)\}$ converges weakly to $f(t)$ if for each $g(t)$ in $L^2(0,T)$ one has $\int_0^T f_n(t)g(t)\,dt\to\int_0^T f(t)g(t)\,dt$ as $n\to\infty$. For example, let $f_n(t)=1$ for $n<t<n+1$ and $f_n(t)=0$ for all other t. Then, using Lebesque's convergence theorem (Ref. 9, p. 88), one can show that $\{f_n(t)\}$ converges weakly to 0 in $L^2(0,\infty)$ but not strongly to 0. A series $\Sigma_{n=1}^{\infty}\boldsymbol{u}_n$ converges to $\boldsymbol{u}$ if the sequence of partial sums $\{\Sigma_{n=1}^{N}\boldsymbol{u}_n\}$ converges to $\boldsymbol{u}$ as $N\to\infty$.

A sequence $\{\boldsymbol{u}_n\}$ is said to be a *Cauchy sequence* if the distance between different elements of the sequence approaches 0 as the indices get large (i.e., $\|\boldsymbol{u}_n-\boldsymbol{u}_m\|\to 0$ as $n,m\to\infty$). A sequence that converges is a Cauchy sequence since $\|\boldsymbol{u}_n-\boldsymbol{u}_m\|\le\|\boldsymbol{u}_n-\boldsymbol{u}\|+\|\boldsymbol{u}_m-\boldsymbol{u}\|$ by the triangle inequality. However, the opposite is not necessarily true (i.e., there are vector spaces equipped with inner products in which a vector space with some Cauchy sequences do not converge). The case in which a vector space with an inner product has the property that every Cauchy sequence converges is important for establishing the existence of solutions to various types of equations. A vector space H is said to be *complete* if every Cauchy sequence converges. It turns out that every vector space equipped with an inner product can be extended to a Hilbert space, which is called the *completion* of the original vector space; see Ref. 5, p. 331. In what follows many of the definitions and concepts are valid in any vector space with an inner product, while the theorems that show the existence of a solution to certain types of equations are usually valid only for a Hilbert space. The proof of the completeness of the L^2 spaces involves the theory of Lebesgue integration; see Ref. 9, p. 117.

In the L^2 spaces two functions $f(t)$ and $g(t)$ are considered to be the same if they are equal *for almost all* t (i.e., if they are equal except on a set of measure zero). A set S of real numbers has *measure zero* if for each positive number ϵ there is a sequence of intervals such that the total length of the intervals is less than ϵ and each point of S is contained in one of the intervals; see Ref. 9, p. 54. For example, the set S consisting of all the integers has measure zero.

A subset S of H is *closed* if, whenever a sequence $\{\boldsymbol{u}_n\}$ of elements of S converges to an element $\boldsymbol{u}$ of H, $\boldsymbol{u}$ is in S. In other words, S is closed if it contains the limit of any convergent sequence in S. For example, let S be the set of functions $\boldsymbol{u}(t)$ in $L^2(-\infty, \infty)$ such that $\boldsymbol{u}(-t) = \boldsymbol{u}(t)$ for almost all t (i.e. the set of even functions). Then S is closed. On the other hand, let S_1 be the set of functions that are equal almost everywhere to a function that is continuous. S_1 is not closed since there are sequences of continuous functions that converge in L^2 to functions that are not continuous.

The closure of a subset S of H is the set of elements $\boldsymbol{u}$ in H such that there exists a sequence $\{\boldsymbol{u}_n\}$ in S which converges to $\boldsymbol{u}$. A subset S of H is called *dense* if its closure is all of H. For example, let ${C^\infty}_0(E)$ be the set of all functions $u(x)$ that have continuous partial derivatives of all orders and for which there is a closed, bounded subset K of E such that $u(x) = 0$ outside K. Then ${C^\infty}_0(E)$ is a dense subset of $L^2(E)$; see Ref. 1, p. 1641. This fact can be used to show that most differential operators are defined on dense subsets of $L^2(E)$.

If a linear subspace M of a Hilbert space is closed, then it is a Hilbert space in its own right. If M is not closed, then it is not complete with the inner product of H, so it is an example of a vector space with an inner product that is not a Hilbert space. For example, the set S_1 mentioned previously is an example of a vector space with an inner product that is not complete.

The Schwarz inequality allows one to define the *angle* θ between two vectors via the formula $\mathrm{Re}(\boldsymbol{u}, v) = \|\boldsymbol{u}\| \|\boldsymbol{v}\| \cos\theta$, which is familiar in two and three dimensions. Two elements u and v are said to be *orthogonal* if $(\boldsymbol{u}, \boldsymbol{v}) = 0$. In the case that H is a real Hilbert space, this says that the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$ is $\pi/2$. We say that $\boldsymbol{u}$ is orthogonal to a subset M of H if $\boldsymbol{u}$ is orthogonal to every $\boldsymbol{v}$ in M. The *orthogonal complement* of a set M is the set of all elements in H that are orthogonal to M.

Many interesting least squares problems can be viewed as finding the shortest distance from a vector to a subspace. More precisely, suppose M is a linear subspace of H and $\boldsymbol{u}$ is a vector in H and one wants to find $\boldsymbol{v}$ in M such that the distance from $\boldsymbol{v}$ to $\boldsymbol{u}$ is less than the distance from $\boldsymbol{w}$ to $\boldsymbol{u}$ for all other $\boldsymbol{w}$ in M. It is not hard to see (see Ref. 8, p. 82) that this occurs precisely if $\boldsymbol{v} - \boldsymbol{u}$ is orthogonal to M. This $\boldsymbol{v}$ is called the *orthogonal projection* of u onto M, and the function P that maps u to v is called the *projection operator* of H onto M. The following *projection theorem* establishes that there is precisely one such $\boldsymbol{v}$ if M is closed:

Theorem 1. (projection theorem) Let M be a closed linear subspace of a Hilbert space H. For any $\boldsymbol{u}$ in H there is a unique $\boldsymbol{v}$ in H such that $\|\boldsymbol{u} - \boldsymbol{v}\| \leq \|\boldsymbol{u} - \boldsymbol{w}\|$ for all $\boldsymbol{w}$ in M. This $\boldsymbol{v}$ is characterized by $(\boldsymbol{v} - \boldsymbol{u}, \boldsymbol{w}) = 0$ for all $\boldsymbol{w}$ in M. Thus $\boldsymbol{u}$ can be uniquely written as $\boldsymbol{u} = \boldsymbol{v} + \boldsymbol{z}$, where $\boldsymbol{v}$ is in M and $\boldsymbol{z}$ is in the orthogonal complement of M.

The basic idea of the proof is quite simple. Let d be the greatest lower bound of the distances of elements of M to $\boldsymbol{u}$. Take a sequence $\{\boldsymbol{v}_n\}$ of vectors in M such that the distance from $\boldsymbol{v}_n$ to $\boldsymbol{u}$ approaches d. Then one can use the parallelogram law to show that this is a Cauchy sequence and hence converges to an element $\boldsymbol{v}$ of M. It is then quite easy to show that the distance from $\boldsymbol{v}$ to $\boldsymbol{u}$ is d, and hence v is the desired element; see Ref. 8, p. 82 for details.

It is well known that in finite dimensions an orthogonal coordinate system can be defined by a finite orthonormal set of vectors. This is still useful in an arbitrary Hilbert space, but one has to allow infinite orthonormal sets. For simplicity we restrict our attention to countably infinite sets. A finite or infinite sequence $\{\boldsymbol{u}_n\}$ of elements is called an *orthonormal* set if each $\boldsymbol{u}_n$ has length one and $\boldsymbol{u}_n$ is orthogonal to $\boldsymbol{u}_m$ for different n and m. The set is called *complete* if there is no other vector in H orthogonal to all the $\boldsymbol{u}_n$. A complete orthonormal set is also called an *orthonormal basis*. A classic example of a complete orthonormal set is the sequence of trigonometric functions

$$\frac{1}{\sqrt{2\pi}}, \frac{\cos t}{\sqrt{\pi}}, \frac{\sin t}{\sqrt{\pi}}, \ldots, \frac{\cos(nt)}{\sqrt{\pi}}, \frac{\sin(nt)}{\sqrt{\pi}}, \ldots \qquad (3)$$

considered as elements of $L^2(-\pi, \pi)$. Showing that these are orthonormal is simply a matter of integration, but showing that they are complete requires more advanced tools; see Courant and Hilbert (10), p. 65. There are many more examples of complete orthonormal sets, including the well-known orthogonal polynomials such as the Legendre polynomials in $L^2(-1, 1)$ and the spherical harmonics in $L^2(S)$, where S is the unit sphere in three dimensions; see Ref. 10, chap. II. Orthonormal bases are important because one can expand an arbitrary vector as a superposition of the basis elements, as in the next theorem. This, in turn, leads to formulas for the solution of differential equations for which the basis elements are particularly suited.

Theorem 2. Let $\{\boldsymbol{u}_n\}$ be an orthonormal set and $\{a_n\}$ be a sequence of complex numbers. Then $\Sigma^{\infty}{}_{n=1}\, a_n \boldsymbol{u}_n$ converges if and only if $\Sigma^{\infty}{}_{n=1}\, |a_n|^2 < \infty$. In this case one has $\|\Sigma^{\infty}{}_{n=1}\, a_n \boldsymbol{u}_n\|^2 = \Sigma^{\infty}{}_{n=1}\, |a_n|^2$, a generalization of the Pythagorean theorem. If u is in H, then $\Sigma^{\infty}{}_{n=1}\, (\boldsymbol{u}, \boldsymbol{u}_n)\boldsymbol{u}_n$ is the orthogonal projection of $\boldsymbol{u}$ on the subspace M, which is the closure of the set of all superpositions of the $\boldsymbol{u}_n$, and *Bessel's inequality* $\Sigma^{\infty}{}_{n=1}\, |(\boldsymbol{u}, \boldsymbol{u}_n)|^2 \leq \|\boldsymbol{u}\|^2$ holds. If the set is complete, then Bessel's inequality becomes an equality, called *Parseval's formula,* and $\boldsymbol{u} = \Sigma^{\infty}{}_{n=1}\, (\boldsymbol{u}, \boldsymbol{u}_n)\boldsymbol{u}_n$.

The scalars $(\boldsymbol{u}, \boldsymbol{u}_n)$ are called the *(generalized) Fourier coefficients* of $\boldsymbol{u}$ with respect to $\boldsymbol{u}_n$. When this theorem is applied to the trigonometric sequence of Eq. (3), one obtains the classical Fourier series expansion

$$f(t) = a_0 + \sum_{n=1}^{\infty} [a_n \cos(nt) + b_n \sin(nt)]$$

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t)\, dt, \quad a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(nt)\, dt$$

$$b_n = \frac{1}{\pi} f(t) \sin(nt)\, dt$$

Other examples include the expansions of functions in terms of other well-known orthogonal sets of functions, such as the Legendre polynomials and spherical harmonics; see Ref. 10, chap. II.

Linear Operators

A *linear operator* A is a function that maps vectors $\boldsymbol{u}$ in one vector space, called the *domain* of A and denoted by D_A, to vectors $A\boldsymbol{u}$ in another vector space, which satisfies $A(\alpha\boldsymbol{u} + \beta\boldsymbol{v}) = \alpha A\boldsymbol{u} + \beta A\boldsymbol{v}$ for all $\boldsymbol{u}$ and $\boldsymbol{v}$ in D_A and complex numbers α and β. Here we shall be interested in the case where D_A is a subset of a Hilbert space H and A maps D_A into H, or possibly a different Hilbert space. In the case where H is R^n or C^n, a linear operator can be defined by an $n \times n$ matrix $\{a_{ik}\}$ by letting $A\boldsymbol{x} = \boldsymbol{y}$, where $y_i = \Sigma^{n}{}_{k=1}\, a_{ik} x_k$. The identity operator I defined by $I\boldsymbol{u} = \boldsymbol{u}$ is a linear operator in any Hilbert space. Multiplication by a fixed function $a(\boldsymbol{x})$ is a linear operator in $L^2(E)$. More precisely, let $H = L^2(E)$ with $a(\boldsymbol{x})$ a fixed scalar valued function defined for $\boldsymbol{x}$ in E. D_A is the set of all functions $\boldsymbol{u}(\boldsymbol{x})$ in H such that the function $a(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})$ is also in H, and one sets

$$(A\boldsymbol{u})(\boldsymbol{x}) = a(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x}) \tag{4}$$

for $\boldsymbol{u}(\boldsymbol{x})$ in D_A. If $a(\boldsymbol{x})$ is bounded, then $D_A = H$.

Perhaps the most important class of operators are *differential operators,* such as the ordinary differential operator

$$L\boldsymbol{u} = p(t) \frac{d^2\boldsymbol{u}}{dt^2} + q(t) \frac{d\boldsymbol{u}}{dt} + r(t)\boldsymbol{u} \tag{5}$$

where $p(t)$, $q(t)$, and $r(t)$ are given functions defined for t in an interval $a < t < b$. When working with differential operators, another group of Hilbert spaces called the *Sobolev spaces* are very useful; see Ref. 8, p. 55. The Sobolev space $H^1(a, b)$ consists of all functions $f(t)$ in $L^2(a, b)$ whose derivatives $f'(t) = df/dt$ are also in L^2. The inner product and norm are given by $(f, g)_1 = \int^b_a [f(t)g(t)* + f'(t)g'(t)*]\, dt$ and $\|f\|^2_1 = \int^b_a (|f(t)|^2 + |f'(t)|^2)\, dt$. $H^2(a, b)$ is defined in an analogous manner by requiring that the second derivative of $f(t)$ lie in L^2 as well. If $p(t)$, $q(t)$, and $r(t)$ are all bounded and $\boldsymbol{u}(t)$ is in $H^2(a, b)$, then $L\boldsymbol{u}$ in $L^2(a, b)$. If one is solving an initial or boundary value problem associated with L, then one can incorporate the initial or boundary conditions into the domain of L. For example, if the problem requires the solution to be zero at $t = a$ and $t = b$, then one can restrict the domain of L to functions $\boldsymbol{u}$ in $H^2(a, b)$ such that $\boldsymbol{u}(a) = \boldsymbol{u}(b) = 0$. See Ref. 3, p. 146 for a more detailed discussion of these operators.

An important partial differential operator is the Laplacian $\Delta\boldsymbol{u} = \partial^2\boldsymbol{u}/\partial x^2_1 + \cdots + \partial^2\boldsymbol{u}/\partial x^2_n$. As with ordinary differential operators, the Sobolev spaces are an invaluable tool when working with partial differential operators. If E is an open set in R^n, then $H^1(E)$ is the set of all functions $\boldsymbol{u}(\boldsymbol{x})$ in $L^2(E)$ with the property that all of its first partial derivatives also lie in $L^2(E)$. The inner product $(,)_1$ in this space is given by $(\boldsymbol{u}, \boldsymbol{v})_1 = (\boldsymbol{u}, \boldsymbol{v}) + \Sigma^n_{i=1} (\partial\boldsymbol{u}/\partial x_i, \partial\boldsymbol{v}/\partial x_i)$, where $(,)$ is the inner product in $L^2(E)$. $H^2(E)$ is defined in a similar manner. In this context one can use the following generalization of the classical notion of partial derivatives: One says that $\partial\boldsymbol{u}/\partial x_i = \boldsymbol{v}$ in E if there is a sequence of functions $\{\boldsymbol{u}_m(\boldsymbol{x})\}$ in $L^2(E)$ such that each $\boldsymbol{u}_m$ has continuous first partial derivatives that lie in $L^2(E)$ and such that $\boldsymbol{u}_m \to \boldsymbol{u}$ and $\partial\boldsymbol{u}_m/\partial x_i \to \boldsymbol{v}$ in $L^2(E)$ as $m \to \infty$; see Ref. 8, pp. 46–59 for more discussion of generalized derivatives or weak derivatives. Just as with ordinary differential operators, one often includes boundary conditions in the domain of the operator. Suppose one has a boundary value problem involving the Laplace operator on a subset E of R^n with zero Dirichlet boundary conditions. Then the corresponding operator, Δ_d, would have a domain, $D\Delta_d$, consisting of all functions in $H^2(E)$ that are 0 on B and

$$\Delta_d\boldsymbol{u} = \frac{\partial^2\boldsymbol{u}}{\partial x^2_1} + \cdots + \frac{\partial^2\boldsymbol{u}}{\partial x^2_n} \quad \text{for} \boldsymbol{u} \text{ in} D_{\Delta_d} \tag{6}$$

See Ref. 3, pp. 297–305 for more details and examples.

The solution of many differential equations can be expressed in terms of *integral operators* of the form

$$K\boldsymbol{u}(\boldsymbol{x}) = \int_E k(\boldsymbol{x}, \boldsymbol{y})\boldsymbol{u}(y)\, d\boldsymbol{y} \tag{7}$$

where the *kernel* $k(\boldsymbol{x}, \boldsymbol{y})$ is a given function of $\boldsymbol{x}$ and $\boldsymbol{y}$ in an open set E of R^n. (For a boundary value problem, the kernel is often called the *Green's function.*) For example, the solution of $d^2\boldsymbol{u}/dt^2 = f(t)$ for $0 < t < 1$ with boundary conditions $\boldsymbol{u}(0) = \boldsymbol{u}(1) = 0$ is $u(t) = \int^1_0 k(t, s)f(s)\, ds$ with $k(t, s) = (t - 1)s$ for $s < t$ and $k(t, s) = (s - 1)t$ for $t < s$; see Ref. 10, pp. 351, 371.

One possible domain for the integral operator of Eq. (7) is the set of all functions $\boldsymbol{u}$ in $H = L^2(E)$ such that the integral of Eq. (7) exists for almost all $\boldsymbol{x}$ and the resulting function $K\boldsymbol{u}$ is also in H. However, it often turns out that K has a natural extension to a larger domain; this is discussed further later in this article.

Convolution operators are integral operators of the form $G\boldsymbol{u} = \boldsymbol{g}*\boldsymbol{u}$, where

$$(\boldsymbol{g}^*\boldsymbol{u})(\boldsymbol{x}) = \int_{R^n} \boldsymbol{g}(\boldsymbol{x} - \boldsymbol{y})\boldsymbol{u}(\boldsymbol{y})\, d\boldsymbol{y} \tag{8}$$

with $\boldsymbol{g}(\boldsymbol{x})$ a function of $\boldsymbol{x}$ in R^n. These operators arise in the solution of constant coefficient differential equations, in filtering problems, and in other translation invariant problems. For example, a solution of Poisson's equation,

$\Delta \boldsymbol{u} = \boldsymbol{f}(\boldsymbol{x})$, in three dimensions is $\boldsymbol{u} = \boldsymbol{g} * \boldsymbol{f}$, where $\boldsymbol{g}(\boldsymbol{x}) = 1/(4\pi|\boldsymbol{x}|)$; see Ref. 10, p. 368. The *Hilbert transform* $H\boldsymbol{u}(\boldsymbol{x}) = \int^{\infty}_{-\infty} \boldsymbol{u}(\boldsymbol{y})/(\boldsymbol{x} - \boldsymbol{y})\, d\boldsymbol{y}$ is a convolution operator with kernel $1/x$.

An important integral operator is the *Fourier transform* $\mathscr{F}$, which maps the function $f(t)$ to the function $F(\omega) = 1/\sqrt{2\pi} \int^{\infty}_{-\infty} e^{-j\omega t} f(t)\, dt$, where $j = \sqrt{-1}$. Thus $F = \mathscr{F} f$. This integral exists for all ω if $\int^{\infty}_{-\infty} |f(t)|\, dt < \infty$. Even though not every function $f(t)$ in $L^2(-\infty, \infty)$ meets this requirement, it turns out that the operator $\mathscr{F}$ has a natural extension to all $f(t)$ in $L^2(-\infty, \infty)$ so that $F = \mathscr{F} f$ also lies in $L^2(-\infty, \infty)$. In fact, $\mathscr{F}$, as well as differentiation and convolution, can be extended to the class of tempered distributions, which includes not only functions in $L^2(-\infty, \infty)$ but also distributions like the Dirac delta function; see Ref. 8, p. 146. A function $f(t)$ can be recovered from its Fourier transform $F(\omega)$ via the *inverse Fourier transform* $\mathscr{F}^{-1}$, which is given by $f(t) = 1/\sqrt{2\pi} \int^{\infty}_{-\infty} e^{j\omega t} F(\omega)\, d\omega$; see Ref. 8, p. 147. This can be written as $\mathscr{F}^{-1}\mathscr{F} f = f$. Thus $\mathscr{F}^{-1}\mathscr{F} = I$.

This last formula is stated in terms of the product (or composition) of operators. If A and B are linear operators, then the product AB denotes the linear operator defined by $(AB)\boldsymbol{u} = A(B\boldsymbol{u})$; the domain of AB is the set of all $\boldsymbol{u}$ in the domain of B such that $B\boldsymbol{u}$ is in the domain of A. If A and B are operators defined by $n \times n$ matrices, then the matrix of AB is the usual matrix product of the matrices of A and B. The Fourier transform extends to functions $\boldsymbol{u}(\boldsymbol{x})$ of n variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ by the formula

$$\mathscr{F}\boldsymbol{u}(\boldsymbol{\xi}) = (2\pi)^{-n/2} \int_{R^n} e^{-j(\boldsymbol{\xi}, \boldsymbol{x})} \boldsymbol{u}(\boldsymbol{x})\, d\boldsymbol{x} \tag{9}$$

where $(\xi, \boldsymbol{x}) = \xi_1 x_1 + \cdots + \xi_n x_n$. The Fourier transform converts differentiation and convolution to multiplication; that is,

$$\mathscr{F}(\partial \boldsymbol{u}/\partial \boldsymbol{x}_k) = j\xi_k \mathscr{F}(f), \qquad \mathscr{F}(\boldsymbol{u} * \boldsymbol{v}) = (2\pi)^{n/2} \mathscr{F}(\boldsymbol{u}) \mathscr{F}(\boldsymbol{v}) \tag{10}$$

See Ref. 8, p. 160. Thus taking Fourier transforms converts any constant coefficient differential operator on R^n into a multiplication operator. For example, applying Eq. (7) twice gives $\mathscr{F}(\Delta \boldsymbol{u}) = -|\xi|^2\, \mathscr{F}(\boldsymbol{u})$.

A linear operator A is *bounded* if there exists a constant M such that $\|A\boldsymbol{u}\| \le M\|\boldsymbol{u}\|$ for all $\boldsymbol{u}$ in D_A, and the smallest constant M is defined to be the *norm* of operator A, denoted by $\|A\|$. A being bounded is equivalent to A being *continuous* (i.e., if $\{\boldsymbol{u}_n\}$ is a sequence in D_A that converges to $\boldsymbol{u}$, which is also in D_A, then $A\boldsymbol{u}_n \to A\boldsymbol{u}$). For example, a multiplication operator (5) is bounded if and only if $a(\boldsymbol{x})$ is a bounded function; that is, there is a constant C such that $|a(\boldsymbol{x})| \le C$ for almost all $\boldsymbol{x}$. In this case $\|A\| \le C$. Showing that a particular integral operator of the form of Eq. (7) is bounded can require some sophisticated work with inequalities. For example, suppose $\int_E |k(\boldsymbol{x}, \boldsymbol{y})|\, d\boldsymbol{y} \le M$ for all x, and $\int_E |k(\boldsymbol{x}, \boldsymbol{y})|\, d\boldsymbol{x} \le M$ for all $\boldsymbol{y}$. Then it can be shown that K is defined on all of $L^2(E)$ and is bounded from $L^2(E)$ into $L^2(E)$ with $\|K\| \le M$; see Ref. 3, p. 144. In particular, a convolution operator is bounded if $\int R^n |g(\boldsymbol{x})|\, d\boldsymbol{x} < \infty$. The Hilbert transform does not meet this condition, but it can also be shown to be bounded in $L^2(-\infty, \infty)$; see Ref. 1, pp. 1041–1073.

A linear operator T is an *isometry* if the domain of T is all of H and it preserves inner products; that is, $(T\boldsymbol{u}, T\boldsymbol{v}) = (\boldsymbol{u}, \boldsymbol{v})$ for all $\boldsymbol{u}$ and $\boldsymbol{v}$ in H. This implies T also preserves lengths and distances $\|T\boldsymbol{u}\| = \|\boldsymbol{u}\|$ for all $\boldsymbol{u}$. In particular, an isometry is bounded and $\|T\| = 1$. If, in addition, the equation $T\boldsymbol{u} = f$ has a solution $\boldsymbol{u}$ for each f in H, then T is said to be *unitary.* The Fourier transform and its inverse are examples of unitary operators; see Ref. 8, p. 154.

Most differential operators are not bounded when regarded as an operator from L^2 to itself. For example, consider $L\boldsymbol{u} = d^2\boldsymbol{u}/dt^2$ as an operator from $L^2(0, 2\pi)$ to itself. Then $L(\sin(nt)) = -n^2 \sin(nt)$ for any n. So $\|L(\sin(nt))\|/\|\sin(nt)\| = n^2$, which can be arbitrarily large. On the other hand, many of these same operators are bounded when regarded as operators from a Sobolev space to L^2. For example, the operator L given by Eq. (5) is a bounded operator from H^2 to L^2 if $p(t)$, $q(t)$, and $r(t)$ are bounded.

A linear operator A is *closed* if whenever $\{\boldsymbol{u}_n\}$ is a sequence in D_A, and both $\boldsymbol{u}_n \to \boldsymbol{u}$ and $A\boldsymbol{u}_n \to \boldsymbol{f}$, then $\boldsymbol{u}$ is in D_A and $A\boldsymbol{u} = \boldsymbol{f}$; that is, the set of ordered pairs $(\boldsymbol{u}, A\boldsymbol{u})$, where $\boldsymbol{u}$ varies over D_A, is a closed set in the product space $H \times H$ of all ordered pairs $(\boldsymbol{u}, \boldsymbol{v})$, where $\boldsymbol{u}$ and $\boldsymbol{v}$ are in H. A bounded operator is closed if and only if D_A is a closed subset of H. If an operator is not closed, it may be possible to extend it to a closed operator A^c, by defining $A^c\boldsymbol{u} = \boldsymbol{f}$ if $\{\boldsymbol{u}_n\}$ is a sequence in D_A, and both $\boldsymbol{u}_n \to \boldsymbol{u}$ and $A\boldsymbol{u}_n \to \boldsymbol{f}$. This gives a unique $\boldsymbol{f}$ for the value of $A^c\boldsymbol{u}$ if A satisfies the condition that if whenever $\{\boldsymbol{u}_n\}$ is a sequence in D_A, and $\boldsymbol{u}_n \to 0$ and $A\boldsymbol{u}_n \to \boldsymbol{f}$, then $\boldsymbol{f} = 0$; such an operator is called *closable.* For example, any bounded operator is closable and the closure is also bounded with the same norm as is defined on the closure of D_A. In particular, it is possible to extend the Fourier transform and its inverse in this way to all of $L^2(R^n)$. For simplicity, we will denote the extended operators by $\mathscr{F}$ and $\mathscr{F}^{-1}$.

The Equation $A\boldsymbol{u} = \boldsymbol{f}$

Many problems in the classical theory of fields can be thought of as operator equations of the form $A\boldsymbol{u} = \boldsymbol{f}$ in $H = L^2(E)$, where E, a subset of R^n, is the domain of the independent variables. In the operator equation $A\boldsymbol{u} = \boldsymbol{f}$, the problem is to find $\boldsymbol{u}$ given A and $\boldsymbol{f}$. Consider, for example, the boundary value problem consisting of Poisson's equation in E with zero Dirichlet boundary conditions on B, the boundary of E. Given $\boldsymbol{f}(\boldsymbol{x})$ defined in E, one wants to find $\boldsymbol{u}(\boldsymbol{x})$ such that

$$\Delta\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{x}) \quad \text{in } E, \text{ and } \boldsymbol{u}(\boldsymbol{x}) = 0 \text{ on } B \tag{11}$$

This can be written as $\Delta_d\boldsymbol{u} = \boldsymbol{f}$, where Δ_d is the Laplace operator defined by Eq. (6).

The problem $A\boldsymbol{u} = \boldsymbol{f}$ is *well posed* if there is one and only one solution $\boldsymbol{u}$ for any $\boldsymbol{f}$ and $\boldsymbol{u}$ depends continuously on $\boldsymbol{f}$. Given an operator A, its *range,* denoted by R_A, is the set of elements $\boldsymbol{f}$ for which the equation $A\boldsymbol{u} = \boldsymbol{f}$ has a solution $\boldsymbol{u}$ in D_A. Thus the equation $A\boldsymbol{u} = \boldsymbol{f}$ has a solution for each $\boldsymbol{f}$ in H if and only if the range of A is H. The operator A is said to be *invertible* if there is only one solution to the equation $A\boldsymbol{u} = \boldsymbol{f}$ for any $\boldsymbol{f}$ in the range of A. If A is linear, this is equivalent to $A\boldsymbol{u} = 0$ only if $\boldsymbol{u} = 0$. If A is invertible, then one can define the inverse operator of A, denoted by A^{-1}, by setting $A^{-1}\boldsymbol{f} = \boldsymbol{u}$ if $A\boldsymbol{u} = \boldsymbol{f}$. Often the term A^{-1} is used as a synonym for A being invertible. The domain of A^{-1} is the range of A, and one has $A^{-1}A = I$ and $AA^{-1} = I$, where the identity operator is restricted to D_A in the first case and ${D_A}^{-1}$ in the second. If A is the operator corresponding to an $n \times n$ matrix, then A is invertible if and only if the determinant of the matrix is not zero and the matrix of A^{-1} is the usual matrix inverse of A. A multiplication operator of the form of Eq. (4) is invertible if $a(\boldsymbol{x}) \neq 0$ for almost all $\boldsymbol{x}$, in which case A^{-1} is the multiplication operator by $1/a(\boldsymbol{x})$. In this case A^{-1} will be bounded with domain $L^2(E)$ if $1/a(\boldsymbol{x})$ is bounded uniformly almost everywhere. Since the Fourier transform converts differentiation into multiplication [see Eq. (10)], it follows that the Laplace operator on all of R^n is invertible and its inverse is the convolution operator with kernel equal to $\mathscr{F}^{-1}(-(2\pi)^{-n/2}|\xi|^{-2})$. To summarize, the problem $A\boldsymbol{u} = \boldsymbol{f}$ is well posed if and only if A is invertible and A^{-1} is a bounded operator with domain equal to H.

Showing that an operator A is invertible and that A^{-1} is bounded is equivalent to showing that $\|A\boldsymbol{u}\| \geq c\|\boldsymbol{u}\|$ for some positive constant c. To obtain this inequality, it suffices to show an inequality such as $|(A\boldsymbol{u}, \boldsymbol{u})| \geq c\|\boldsymbol{u}\|^2$ for some positive constant c. Then one can use the Schwarz inequality of Eq. (2) to conclude that $\|A\boldsymbol{u}\| \|\boldsymbol{u}\| \geq c\|\boldsymbol{u}\|^2$, which gives $\|A\boldsymbol{u}\| \geq c\|\boldsymbol{u}\|$. Consider, for example, the Laplace operator Δ_d with Dirichlet boundary values defined by Eq. (6). For simplicity, suppose that E is a bounded domain with smooth boundary B. Then

by Green's integral formula (see Ref. 11, p. 441) one has

$$(\Delta \boldsymbol{u}, \boldsymbol{v}) = -\sum_{k=1}^{n} \left(\frac{\partial \boldsymbol{u}}{\partial x_k}, \frac{\partial \boldsymbol{v}}{\partial x_k} \right) \tag{12}$$

if $\boldsymbol{v} = 0$ on B. Here (,) is the inner product in $L^2(E)$. Applying this formula, one obtains $(\Delta_d \boldsymbol{u}, \boldsymbol{u}) = -\Sigma^n{}_{k=1} \|\partial \boldsymbol{u}/\partial x_k\|^2$ for $\boldsymbol{u}$ in $D\Delta_d$, where $\| \ \|$ is the norm in $L^2(E)$. An inequality of Poincaré (see Ref. 12, p. 169) says that for a bounded domain E there is a constant C such that $\|\boldsymbol{u}\|^2 \leq C \ \Sigma^n{}_{k=1} \|\partial \boldsymbol{u}/\partial x_k\|^2$ for all $\boldsymbol{u}$ in $H^1(E)$ that are 0 on B. Thus

$$-(\Delta_d \boldsymbol{u}, \boldsymbol{u}) \geq c\|\boldsymbol{u}\|_1^2 \geq c\|\boldsymbol{u}\|^2 \tag{13}$$

where $\| \ \|_1$ is the norm in $H^1(E)$ and c is a positive constant. Thus Δ_d has a bounded inverse. Showing that the range of Δ_d is all of $L^2(E)$ requires some more work; see the Lax–Milgram theorem later in this article.

A *functional* on a vector space V is a mapping from V to the complex numbers; a *linear functional* is one that is linear. The *conjugate* (or *dual*) space of H, denoted by $H*$, consists of all bounded linear functionals F on H such that $D_F = H$. A simple example is $F(f) = 1/2\pi \int^{2\pi}{}_0 f(t)\, dt$, which is the functional that assigns to each function f in $H = L^2(0, 2\pi)$ its average value. This is a special case of the following general way to obtain linear functionals. Let v be a fixed element of H and let $F_v(\boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{v})$. Then this linear functional that is bounded by the Schwarz inequality. In fact, $\|F_v\| = \|\boldsymbol{v}\|$. The *Riesz representation theorem* says that any bounded linear functional on H can be obtained from an element $\boldsymbol{v}$ of H in this way.

Theorem 3. (Riesz) Let F be a bounded linear functional defined everywhere on a Hilbert space H. Then there exists a unique element $\boldsymbol{v}$ in H such that $\|\boldsymbol{v}\| = \|F\|$ and $F(\boldsymbol{u}) = (\boldsymbol{u}, \boldsymbol{v})$ for all $\boldsymbol{u}$ in H.

The proof is quite easy. Let M be the set of all $\boldsymbol{u}$ such that $F(\boldsymbol{u}) = 0$. M is a closed linear subspace of H. By the projection theorem there is a $\boldsymbol{v}$ in H that is orthogonal to M and has length 1. By multiplying F by a constant, we may reduce to the case where $F(\boldsymbol{v}) = 1$. Now let $\boldsymbol{u}$ be any element of H. Note that $\boldsymbol{u} - F(\boldsymbol{u})\boldsymbol{v}$ is in M. Thus $(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u} - F(\boldsymbol{u})\boldsymbol{v}, \boldsymbol{v}) + (F(\boldsymbol{u})\boldsymbol{v}, \boldsymbol{v}) = F(\boldsymbol{u})$.

The Riesz representation theorem can be used to prove the *Lax-Milgram theorem,* which, in turn, is useful for proving the existence of solutions to many elliptic boundary value problems. This theorem is often stated in terms of bilinear forms that are closely related to linear operators. A *sesquilinear form* B associates a scalar $B(\boldsymbol{u}, \boldsymbol{v})$ to each pair of elements $\boldsymbol{u}$ and $\boldsymbol{v}$ in a vector space D_B, called the domain of B, such that $B(\boldsymbol{u}, \boldsymbol{v})$ is linear in $\boldsymbol{u}$ for each fixed $\boldsymbol{v}$ and conjugate linear in $\boldsymbol{v}$ for each fixed $\boldsymbol{u}$; that is, $B(\boldsymbol{u}, \alpha\boldsymbol{v}_1 + \beta\boldsymbol{v}_2) = \alpha* B(\boldsymbol{u}, \boldsymbol{v}_1) + \beta* B(\boldsymbol{u}, \boldsymbol{v}_2)$. An example is the *Dirichlet form*

$$D(\boldsymbol{u}, v) = \sum_{k=1}^{n} \int_E \frac{\partial \boldsymbol{u}}{\partial x_k} \frac{\partial \boldsymbol{v}^*}{\partial x_k}\, dx = \sum_{k=1}^{n} \left(\frac{\partial \boldsymbol{u}}{\partial x_k}, \frac{\partial \boldsymbol{v}^*}{\partial x_k} \right) \tag{14}$$

where (,) is the inner product in $L^2(E)$. The domain is the set of functions in $H^1(E)$ that are 0 on B, the boundary of E. A bilinear form $B(\boldsymbol{u}, \boldsymbol{v})$ is said to be *bounded* if there is a constant C such that $|B(\boldsymbol{u}, \boldsymbol{v})| \leq C\|\boldsymbol{u}\| \ \|\boldsymbol{v}\|$ for all $\boldsymbol{u}$ and $\boldsymbol{v}$ in D_B. For example, the Dirichlet form is bounded with respect to the norm in $H^1(E)$ but not $L^2(E)$.

Given a sesquilinear form B with dense domain, there is a linear operator A associated with B. Its domain is the set of all $\boldsymbol{u}$ such that there is an $\boldsymbol{f}$ such that $B(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v})$ for all $\boldsymbol{v}$ in D_B. This $\boldsymbol{f}$ is uniquely determined since D_B is assumed to be dense. Consider the Dirichlet form defined by Eq. (14). Since $(-\Delta_d \boldsymbol{u}, \boldsymbol{v}) = D(\boldsymbol{u}, \boldsymbol{v})$ for $\boldsymbol{u}$ and $\boldsymbol{v}$ in $D\Delta_d$, it follows that the operator associated with the Dirichlet form is either equal to or an extension of $-\Delta_d$. In fact, the two operators are equal; this is an important theorem of Friedrichs; see Ref. 1, p. 1789.

The importance of sesquilinear forms is that they are a useful tool for proving the existence of solutions to the equation $A\boldsymbol{u} = \boldsymbol{f}$ when A is the associated operator. This is the content of the following theorem.

Theorem 4. (Lax–Milgram) Let H_1 be a Hilbert space with norm $\| \ \|_1$ and $B(\boldsymbol{u}, \boldsymbol{v})$ a bounded bilinear form on H_1. Assume that B is coercive on H_1—that is, there exists a constant $c > 0$ such that $|B(\boldsymbol{u}, \boldsymbol{u})| \geq c\|\boldsymbol{u}\|^2{}_1$ for all $\boldsymbol{u}$ in H_1. Then for any bounded linear functional F on H_1 there exists a unique element $\boldsymbol{w}$ in H_1 such that $B(\boldsymbol{u}, \boldsymbol{w}) = F(\boldsymbol{u})$ for all $\boldsymbol{u}$ in H_1. Suppose H is another Hilbert space with norm $\| \ \|$ such that H_1 is a dense subset of H and there exists a constant C such that $\|\boldsymbol{u}\| \leq \mathrm{C}\|\boldsymbol{u}\|_1$ for all $\boldsymbol{u}$ in H_1. Let A be the operator in H associated with B. Then A is invertible, the range of A is all of H, and A^{-1} is a bounded linear operator on H.

For a proof, see Ref. 8, p. 92. According to Eq. (13), the Dirichlet form defined by Eq. (14) satisfies the hypotheses of the Lax–Milgram theorem with H_1 being the subspace of $H^1(E)$ consisting of functions in $H^1(E)$ that are 0 on B. If one takes $H = L^2(E)$, then we have noted that the operator in H associated with the Dirichlet form is Δ_d. Thus the Lax–Milgram theorem establishes the existence of a solution to Poisson's equation [Eq. (11)].

A linear operator defined by a matrix mapping C^n to itself has the property that its range is all of C^n if and only if it is invertible. This is not true for a general operator in infinite dimensions, but there are some important special cases in which this is true. One of these involves compact operators. A linear operator on a Hilbert space is said to be *compact* if it maps any bounded sequence into a sequence with a convergent subsequence. In particular, a compact operator is bounded. It can also be shown that the identity mapping is a compact operator from $H^1(E)$ to $L^2(E)$ if E is a bounded set; see Ref. 1, p. 1691. This can be used to show that $\Delta^{-1}{}_D$ is a compact operator in $L^2(E)$ if E is bounded. The following theorem, called the Fredholm alternative theorem, extends the aforementioned result of matrices in finite dimensions; for a proof see Ref. 8, sec. X.5.

Theorem 5. (Fredholm) Let $A = C - \lambda I$, where C is compact and $\lambda \neq 0$ is a complex number. Then A is invertible if and only if the range of A is all of H.

Let H be a Hilbert space and A an operator on it with D_A is dense in H. The function $B(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}, A\boldsymbol{v})$ is a sesquilinear form. The associated operator $A*$ is called the adjoint of A; that is, $(\boldsymbol{u}, A\boldsymbol{v}) = (A*\boldsymbol{u}, \boldsymbol{v})$ for $\boldsymbol{u}$ in D_{A*} and $\boldsymbol{v}$ in D_A. For example, if A is an $n \times n$ matrix $\{a_{ik}\}$, then the matrix $\{b_{ik}\}$ of $A*$ is just the conjugate of the transpose of A (i.e., $b_{ik} = a^*{}_{ki}$). If A is the multiplication operator of Eq. (4), then $A*$ is the multiplication operator by $a(x)*$. If K is an integral operator of the form of Eq. (7), then $K*$ is equal to or an extension of the integral operator with kernel $p(x, y)$ given by $p(x, y) = k(y, x)*$. For a unitary operator T one has $T* = T^{-1}$.

An operator A is said to be *Hermitian* if $(A\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}, A\boldsymbol{v})$ for all $\boldsymbol{u}$ and $\boldsymbol{v}$ in D_A. If A is Hermitian, then $A*$ is equal to or an extension of A. For example, it follows from Eq. (12) that Δ_d is Hermitian. If A is invertible, then A is Hermitian if and only if A^{-1} is Hermitian. If $A* = A$, then A is said to be *self-adjoint.* For example, if A is an $n \times n$ matrix $\{a_{ik}\}$, then A is self-adjoint if $a_{ik} = a^*{}_{ki}$ for all i and k. The multiplication operator of Eq. (4) is self-adjoint if $a(x)$ is real for all x. If A is a Hermitian operator with domain equal to H, then A is self-adjoint. If A is invertible, then A is self-adjoint if and only if A^{-1} is self-adjoint. If A is Hermitian and invertible and A^{-1} has domain equal to H, then A is self-adjoint. For example, Δ_d is self-adjoint. An operator A is *normal* if $AA* = A*A$. Any self-adjoint or unitary operator is normal. If A is normal, then αA is normal for any complex number α. Any multiplication operator given by Eq. (4) is normal. Normal operators are important because of their spectral properties, which are considered next.

Spectral Theory and Evolution Equations

A number of initial boundary value problems of classical field theory can be cast in the form of an ordinary differential equation of one of the following two forms:

$$\frac{d\boldsymbol{u}}{dt} = A\boldsymbol{u}(t), \quad \text{for } t > 0 \text{ together with } \boldsymbol{u}(0) = \boldsymbol{\phi} \tag{15}$$

$$\frac{d^2\boldsymbol{u}}{dt^2} = A\boldsymbol{u}(t), \quad \text{for } t > 0 \text{ together with } \boldsymbol{u}(0) = \boldsymbol{\phi} \text{ and } \frac{d\boldsymbol{u}}{dt}(0) = \boldsymbol{\psi} \tag{16}$$

In these equations A is a linear operator in a Hilbert space H and the unknown $\boldsymbol{u}(t)$ is a function of $t \geq 0$ whose value at each t is an element of H. The solution $\boldsymbol{u}(t)$ should satisfy the differential equation for $t > 0$ and the initial conditions for $t = 0$. For example, the initial boundary value for the heat (or diffusion) equation

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} &= \Delta \boldsymbol{u}, && \text{for } t \geq 0, \boldsymbol{x} \text{ in } E \\ \boldsymbol{u}(t, \boldsymbol{x}) &= 0, && \text{for } t \geq 0, \boldsymbol{x} \text{ on } B \\ \boldsymbol{u}(0, \boldsymbol{x}) &= \boldsymbol{\phi}(\boldsymbol{x}), && \text{for } x \text{ in } E \end{aligned} \tag{17}$$

can be viewed as an equation of the form of Eq. (15) with $A = \Delta_d$ defined by Eq. (6). Similarly, the corresponding problem for the wave equation

$$\begin{aligned} \frac{\partial^2 \boldsymbol{u}}{\partial t^2} &= \Delta \boldsymbol{u}, && \text{for } t \geq 0, \boldsymbol{x} \text{ in } E \\ \boldsymbol{u}(t, \boldsymbol{x}) &= 0, && \text{for } t \geq 0, \boldsymbol{x} \text{ on } B \\ \boldsymbol{u}(0, \boldsymbol{x}) &= \boldsymbol{\phi}(\boldsymbol{x}), \quad \frac{\partial \boldsymbol{u}}{\partial t}(0, \boldsymbol{x}) = \boldsymbol{\psi}(\boldsymbol{x}) && \text{for } x \text{ in } E \end{aligned} \tag{18}$$

has the form of Eq. (16) with the same A. Equations (15) and (16) are examples of simple evolution equations; for more general evolution equations, see Ref. 8, chap. XIV. In many cases the solution of these equations can be obtained in terms of the eigenvalues λ_n and eigenvectors $\boldsymbol{u}_n$ of the operator A (i.e., $A\boldsymbol{u}_n = \lambda_n \boldsymbol{u}_n$). Eigenvalues are part of the *spectrum* $\sigma(A)$ of A, which consists of all complex numbers that are not in the resolvent set of A. The *resolvent set,* $\rho(A)$, of A consists of all complex numbers λ such that $A - \lambda I$ is invertible and $(A - \lambda I)^{-1}$ is bounded and with domain H; the operator valued function $R(\lambda; A) = (A - \lambda I)^{-1}$ defined for λ in the resolvent set is called the *resolvent* of A. A number λ can be in the spectrum for one of three reasons:

(1) $A - \lambda I$ does not have an inverse (i.e., there exists $\boldsymbol{u} \neq 0$ satisfying $A\boldsymbol{u} = \lambda \boldsymbol{u}$). We say λ is an eigenvalue of A (or λ belongs to the *point spectrum* of A) and that $\boldsymbol{u}$ is an eigenvector corresponding to λ.

(2) $A - \lambda I$ is invertible, but the domain of $(A - \lambda I)^{-1}$ is not dense in H. In this case one says that λ is in the *residual spectrum* of A.

(3) $A - \lambda I$ is invertible, and $(A - \lambda I)^{-1}$ has a dense domain, but it is unbounded. In this case one says that λ is in the *continuous spectrum* of A.

Note that cases 2 and 3 are impossible if H is infinite dimensional. In the case where A is an operator in a Hilbert space of functions like $L^2(E)$, it is common to use the term *eigenfunction* for an eigenvector. For an operator A given by a matrix $\{a_{ik}\}$, a number λ is an eigenvalue if $\det\{a_{ik} - \lambda\delta_{ik}\} = 0$; otherwise λ is in the resolvent. Here det is the determinant and δ_{ik} is 1 if $i = k$ and 0 otherwise.

For the multiplication operator of Eq. (4) a number λ is in the resolvent if $a(\boldsymbol{x}) \neq \lambda$ for almost all $\boldsymbol{x}$ and the function $1/(a(\boldsymbol{x}) - \lambda)$ is bounded and $R(\lambda, A)$ is the multiplication operator by $1/(a(\boldsymbol{x}) - \lambda)$. If $a(\boldsymbol{x}) \neq \lambda$ for almost all x, but the function $1/(a(\boldsymbol{x}) - \lambda)$ is not bounded, then λ is in the continuous spectrum. A number λ is an eigenvalue for this operator if the set S of $\boldsymbol{x}$ where $a(\boldsymbol{x}) = \lambda$ has positive measure. In this case any function that is 0 outside S is a corresponding eigenfunction.

Suppose two operators A and B are related by $A = T^{-1}BT$, where T is an invertible operator, and both T and T^{-1} are bounded and defined on all of H. Then A and B have the same spectrum and points in the spectrum have the same type. For example, Consider the Laplacian Δ as an operator in $L^2(R^n)$. It follows from Eq. (10) that $\Delta = \mathscr{F}^{-1}M\mathscr{F}$, where M is the multiplication operator by $-|\xi|^2$; that is, $M\boldsymbol{v}(\xi) = -|\xi|^2\boldsymbol{v}(\xi)$. Hence the spectrum of Δ is the negative real axis together with the number 0 and the spectrum consists entirely of continuous spectra.

It can be shown that for an operator the spectrum is a closed set; see Ref. 3, p. 174. Furthermore, the spectrum of a self-adjoint operator lies on the real axis. The eigenvectors of a normal operator corresponding to distinct eigenvalues are orthogonal; see Ref. 3, pp. 271, 274. In the case where the eigenvectors of A form a complete orthonormal set $\{\boldsymbol{u}_n\}$ and there exists a constant C such that the real part of each eigenvalue does not exceed C, then the solution to Eq. (15) is

$$\boldsymbol{u}(t) = \sum_{n=1}^{\infty} (\boldsymbol{\phi}, \boldsymbol{u}_n) e^{\lambda_n t} \boldsymbol{u}_n \tag{19}$$

If, in addition, the eigenvalues are all real, then the solution to Eq. (16) is

$$\boldsymbol{u}(t) = \sum_{n=1}^{\infty} [(\boldsymbol{\phi}, \boldsymbol{u}_n) \cos(\sqrt{-\lambda_n}t) + (\boldsymbol{\psi}, \boldsymbol{u}_n) \sin(\sqrt{-\lambda_n}t)/\sqrt{-\lambda_n}]\boldsymbol{u}_n \tag{20}$$

See Ref. 12, pp. 149–150. Thus it is important to know when a normal operator has the property that its eigenvectors form a complete orthonormal set. One case where this occurs is when the operator is compact.

Theorem 6. (Riesz–Schauder and Hilbert–Schmidt) If C is a compact operator, then the spectrum of C consists of a finite or infinite sequence $\{\lambda_n\}$ of eigenvalues and possibly the number 0 (which can also be an eigenvalue, but need not be). Any nonzero eigenvalue λ has *finite multiplicity* (i.e., the collection of eigenvectors corresponding to λ is a subspace of finite dimension). If, in addition, C is normal, then there is a complete orthonormal set consisting of eigenvectors of C. If A is an operator such that $(A - \lambda I)^{-1}$ is compact for some λ, then the spectrum of A consists of a finite or infinite sequence of eigenvalues. In the infinite case the eigenvalues have no finite accumulation point. Every eigenvalue is of finite multiplicity. If, in addition, A is normal, then there is a complete orthonormal set consisting of eigenvectors of A.

See Ref. 3, pp. 185–188, 280 for a proof of this theorem. According to this theorem, it follows that if E is a bounded set, then the eigenvalues of Δ_d form a sequence having no finite limit point, every eigenvalue is of finite multiplicity, and any complex number that is not an eigenvalue is in the resolvent. Furthermore, there is a complete orthonormal set consisting of eigenvectors of Δ_d. In the case of one dimension, where E is the interval $0 < x < \pi$ and $\Delta_d\boldsymbol{u} = d^2\boldsymbol{u}/dx^2$ and functions $\boldsymbol{u}$ in the domain Δ_d are 0 at $x = 0$ and $x = \pi$, the eigenvalues are the positive integers and the eigenfunctions are the functions $\sin(nx)$.

Not every self-adjoint operator has a complete orthonormal set of eigenvectors. The spectral theorem (Theorem 7) is the generalization of the Hilbert–Schmidt theorem to arbitrary self-adjoint operators. It requires a generalization of the notion of a complete orthonormal set called a spectral family. A *projection* P is a self-adjoint operator satisfying $P^2 = P$ and $\|P\| = 1$ if $P \neq 0$. It can be shown that every projection is given by the orthogonal projection of H on some closed linear subspace, as in Theorem 1. A family of projections $\{E(r)\}$, $-\infty < r < \infty$ is called a *spectral family* if (1) $E(r)E(s) = E(s)E(r) = E(r)$ for $r < s$, (2) $E(r) \to E(s)u$ as $r \to s-$, (3) $E(r)$ has a limit as $r \to s$, (4) $E(r)\boldsymbol{u} \to 0$ as $r \to -\infty$, and (5) $E(r)\boldsymbol{u} \to \boldsymbol{u}$ as $r \to \infty$. Given a complete orthonormal set that is the eigenvectors of an operator A, the associated spectral family is defined by $E(r)\boldsymbol{u} = \Sigma\lambda_n \leq r\ (\boldsymbol{u}, \boldsymbol{u}_n)\boldsymbol{u}_n$. Using a spectral family a series of the form $\Sigma^{\infty}{}_{n=1}\, f(\lambda_n)(\boldsymbol{u}, \boldsymbol{u}_n)\boldsymbol{u}_n$ like the ones in Eqs. (19) and (20) can be generalized to an integral. If $f(r)$ is a complex-valued function defined for real r, then the operator $\int^{\infty}{}_{-\infty} f(r)\ dE(r)$ is defined via the formula $(\int^{\infty}{}_{-\infty} f(r)\ dE(r)\boldsymbol{u}, \boldsymbol{v}) = \int^{\infty}{}_{-\infty} f(r)\ d(E(r)\boldsymbol{u}, \boldsymbol{v})$; see Ref. 3, p. 357.

Theorem 7. (Spectral Theorem) To every self-adjoint operator A in a Hilbert space H there corresponds a unique spectral family, called the spectral family of A, such that any bounded linear operator that commutes with A commutes with each $E(r)$ and $A = \int^{\infty}{}_{-\infty} r\ dE(r)$.

See Ref. 3, p. 360 for a proof. For any complex-valued function $f(r)$ defined for real r, let $f(A) = \int^{\infty}{}_{-\infty} f(r)\ dE(r)$. Using the spectral theorem, the formulas of Eq. (19) for the solution to Eq. (15) can be generalized to $\boldsymbol{u}(t) = e^{tA}\phi$ provided that A is a self-adjoint operator whose spectrum lies entirely to the right of some number C. The formula of Eq. (20) for the solution of Eq. (25) can be generalized to $\boldsymbol{u}(t) = \cos(\sqrt{-A}\ t)\phi + (-A)^{-1/2} \sin(\sqrt{-A}\ t)\psi$ for the same type of operator A.

Applications to Quantum Mechanics

To formulate the quantum mechanical description of a physical system, it is first necessary to have a description of the system in terms of classical Hamiltonian mechanics, so we begin with a brief review of this; for more details see Ref. 13. A physical system is described by position coordinates $\boldsymbol{q} = (q_1, \ldots, q_n)$ and their time derivatives $\dot{\boldsymbol{q}} = (\dot{q}_1, \ldots, \dot{q}_n) = d\boldsymbol{q}/dt$, and the time evolution of the system is described by a system of differential equations. For example, the Cartesian coordinates $\boldsymbol{q} = (q_1, q_2, q_3)$ of a single particle of mass m in three dimensions acted on by a force F satisfy *Newton's equations:* $m\, d\dot{\boldsymbol{q}}/d\mathrm{t} = F$. For conservative systems the equations of motion can be put in the form of *Lagrange's equations:*

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0, \qquad i = 1, 2, \ldots, n \tag{21}$$

where the *Lagrangian* $L = T - U$ is the difference between the kinetic energy T and potential energy U. In the preceding example of a particle in three dimensions, suppose $F = -\nabla U$, where $U = U(\boldsymbol{q})$ depends only on $\boldsymbol{q}$. Then $L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = (m|\dot{\boldsymbol{q}}|^2/2) - U(\boldsymbol{q})$ and Eq. (21) is equivalent to $m\ \mathrm{d}\dot{q}/dt = F$. Quantities that are constant in time are called *integrals of motion.* For example, the *energy function* $h(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \Sigma_i\, \dot{q}_i\, \partial L/\partial \dot{q}_i - L$ is an integral of motion since Eq. (21) implies that $dh/dt = 0$. If enough integrals of motion can be found so that the relation between $\boldsymbol{q}$ and t can be expressed in terms of known functions, then the system is said to be *completely integrable* or to be a *classical integrable system.*

To obtain Hamilton's equations of motion, one introduces the *conjugate momenta* $\boldsymbol{p} = (p_1, \ldots, p_n)$ by $p_i = \partial L/\partial \dot{q}_i$. The *Hamiltonian* $H(\boldsymbol{q}, \boldsymbol{p})$ is the result of replacing $\dot{\boldsymbol{q}}$ by $\boldsymbol{p}$ in $h(\boldsymbol{q}, \dot{\boldsymbol{q}})$. One can show that Eq. (21) is

equivalent to Hamilton's equations:

$$\frac{dq_i}{dt} = \frac{\partial H(\boldsymbol{q}, \boldsymbol{p})}{\partial p_i}, \quad \frac{dp_i}{dt} = -\frac{\partial H(\boldsymbol{q}, \boldsymbol{p})}{\partial q_i} \tag{22}$$

For details see Ref. 13, pp. 340–342, and Ref. 14, p. 174. In the preceding example of a single particle, one has $H(\boldsymbol{q}, \boldsymbol{p}) = (|\boldsymbol{p}|^2/2m) + U(\boldsymbol{q})$. If $F = F(\boldsymbol{q}, \boldsymbol{p})$ is some physical quantity, then it follows from Eq. (22) that

$$\frac{dF}{dt} = \{F, H\} \tag{23}$$

where

$$\{F, H\} = \sum_i \left(\frac{\partial F}{\partial \boldsymbol{q_i}} \frac{\partial H}{\partial \boldsymbol{p_i}} - \frac{\partial F}{\partial \boldsymbol{p_i}} \frac{\partial H}{\partial \boldsymbol{q_i}} \right)$$

is the *Poisson bracket* of F and H. Thus F is conserved if $\{F, H\} = 0$. For simplicity, we assume that L and H are independent of t; the preceding equations can be generalized to the case where L and H depend on t. See Ref. 13, pp. 397, 405, and Ref. 14, p. 174 for details.

Quantization is the process of transforming a classical mechanical description of a system into a quantum mechanical one. In quantum mechanics observables such as coordinates, momenta, and energy are represented by self-adjoint operators acting on a Hilbert space H, and the state of the system is described by a vector $\boldsymbol{u}$ in H having length 1. Observables are usually not subject to precise measurement. Rather there is a probability of measuring a certain value for an observable if the system is in a particular state. The following fundamental principle of quantum mechanics makes this more precise.

Principle 1 (P1). If $\boldsymbol{u}$ is the state of the system and A is the operator corresponding to an observable, then the inner product $(\boldsymbol{u}, A\boldsymbol{u})$ represents the average of a series of measurements of the observable over an ensemble of systems that are all described by the state $\boldsymbol{u}$.

As we shall see, the only case when a measurement can be precise is when the state is an eigenvector of A, in which case the value of the observable is the corresponding eigenvalue. This is of particular importance for the energy operator $\mathscr{H}$ whose eigenvalues E_n and eigenvectors $\boldsymbol{u}_n$ satisfy

$$\mathscr{H}\boldsymbol{u}_n = E_n \boldsymbol{u}_n \tag{24}$$

The $\boldsymbol{u}_n$ are called *stationary states* because, as we shall see, they are time invariant. When the system is in a stationary state $\boldsymbol{u}_n$, a measurement of the energy results in the corresponding eigenvalue E_n. The correspondence between the classical mechanical variables and corresponding quantum mechanical operators should satisfy the next two fundamental principles.

Principle 2 (P2). If $F(\boldsymbol{q}, \boldsymbol{p})$ and $G(\boldsymbol{q}, \boldsymbol{p})$ are two functions of coordinates and momentum, with corresponding operators A and B, then the operator corresponding to the Poisson bracket $\{F, G\}$ of F and G should be $[A, B]/j$.

Principle 3 (P3). If $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ are classical variables with corresponding operators $A_1, \ldots, A_n$ that commute and $f(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ is any function of $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$, then $f(A_1, \ldots, A_n)$ should be the operator corresponding to $f(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$.

Here $= h/(2\pi)$ and h is Planck's constant and $[A, B] = AB - BA$ is the commutator of the operators A and B. The observables are said to commute if $[A, B] = 0$. Also, a function of commuting self-adjoint operators can

be defined in a manner similar to the case of one operator, as in the spectral theorem; see Ref. 15, p. 270. One implication of this is the following. Suppose $\boldsymbol{v}$ is an observable with operator A and let $f(\boldsymbol{v})$ be the function that is 1 in an interval I and 0 outside the interval. Then $(\boldsymbol{u}, f(A)\boldsymbol{u})$ coincides with the probability of the value of $\boldsymbol{v}$ being in the interval I if the state is u. However, by the spectral theorem this is equal to $\int^{\infty}_{-\infty} f(\lambda)\, d(\boldsymbol{u}, E(\lambda)\boldsymbol{u}) = \int_I d(\boldsymbol{u}, E(\lambda)\boldsymbol{u})$, where $\{E(\lambda)\}$ is the spectral family associated with A. Thus $d(\boldsymbol{u}, E(\lambda)\boldsymbol{u})$ corresponds to the probability distribution of observing a particular value of the observable when the state is $\boldsymbol{u}$; see Ref. 7, p. 201. Furthermore, $E(A) = E(A, \boldsymbol{u}) = \int^{\infty}_{-\infty} \lambda\, d(\boldsymbol{u}, E(\lambda)\boldsymbol{u}) = (\boldsymbol{u}, A\boldsymbol{u})$ is the mean of this distribution. The usual measure of the uncertainty in observations of the values of A is the standard deviation $\sigma(A)$ given by $\sigma(A) = \sigma(A, \boldsymbol{u}) = [\int^{\infty}_{-\infty} (\lambda - E(A))^2\, d(\boldsymbol{u}, E(\lambda)\boldsymbol{u})]^{1/2} = |(A - E(A)I)u|$. If $I = [\lambda, \lambda] = \{\lambda\}$ consists of a single number, then the probability of measuring the value λ is $(E(\lambda) - E(\lambda^-))\boldsymbol{u}$. This will be 0 unless λ is an eigenvalue. If λ is an eigenvalue, then this becomes $|P(\lambda)\boldsymbol{u}|^2$, where $P(\lambda) = E(\lambda) - E(\lambda^-)$ is the projection on the eigenspace corresponding to λ. In particular, if $\boldsymbol{u}$ is an eigenvector corresponding to λ, then the probability of observing the value λ for $\boldsymbol{u}$ is 1. Hence, as noted earlier, the only possible result of a precise measurement of an observable is one of the eigenvalues of the corresponding operator.

One of the fundamental principles of quantum mechanics, Heisenberg's uncertainty principle, is closely related to this. It says that for two observables A and B that do not commute it is not possible to find a state where the uncertainty of measuring each of these observables is arbitrarily small. More precisely, for any state $\boldsymbol{u}$ one has $2\sigma(A)\sigma(B) \geq |E(AB - BA)|$; for a proof see Ref. 7, pp. 230–247.

One implication of property (P2) is the following: Let $Q_1, \ldots, Q_n$ and $P_1, \ldots, P_n$ denote the operators corresponding to coordinates and momenta. Since $\{q_i, q_k\} = \{p_i, p_k\} = 0$ and $\{q_i, p_k\} = \delta_{ik}$, then

$$[Q_i, Q_k] = [P_i, P_k] = 0, \quad [Q_i, P_k] = j\hbar\delta_{ik} I \tag{25}$$

where $\delta_{ik} = 1$ if $i = k$ and $\delta_{ik} = 0$ if $i \neq k$. It follows from the uncertainty principle that for any state $\boldsymbol{u}$ one has $\sigma(Q_i)\sigma(P_i) \geq /2$ (i.e., the product of the uncertainties in the measurements of any position and corresponding momentum is at least $\hbar/2$).

There are two formulations of quantum mechanics. One of these, developed by Schrödinger, is called wave mechanics and is the most widely used, and a discussion of it is given first. The other, developed by Heisenberg, is called matrix mechanics and a discussion of it follows. In wave mechanics the operators are time independent and the state variable describing the system varies with time. For a system described by Hamiltonian $H(\boldsymbol{q}, \boldsymbol{p})$ the state propagates according to *Schrödinger's equation:*

$$j\hbar \frac{d\boldsymbol{u}}{dt} = \mathscr{H}\boldsymbol{u} \tag{26}$$

where $\mathscr{H} = H(Q, P)$ is the energy operator corresponding to $H(q, p)$. This is an equation of the form of Eq. (15) with $A = -j\mathscr{H}/$. Using the spectral theorem, one can construct the solution to this differential equation as

$$\boldsymbol{u}(t) = e^{-j\mathscr{H}t/\hbar}\boldsymbol{u}(0) = \int_{-\infty}^{\infty} e^{-j\lambda t/\hbar}\, dE(\lambda)\boldsymbol{u}(0) \tag{27}$$

where $\boldsymbol{u}(0)$ is the state when $t = 0$ and $\{E(\lambda)\}$ is the spectral family for $\mathscr{H}$. If the eigenvectors of $\mathscr{H}$ form a complete orthonormal set, then Eq. (27) takes the form similar to Eq. (18); that is,

$$\boldsymbol{u}(t) = \sum_n (\boldsymbol{u}(0), \boldsymbol{u}_n) e^{-j\lambda_n t/\hbar}\boldsymbol{u}_n \tag{28}$$

In the Schrödinger formulation the standard choice of the Hilbert space is $H = L^2(R^n)$ with

$$(Q_i \boldsymbol{u})(\boldsymbol{x}) = x_i \boldsymbol{u}(\boldsymbol{x}) \qquad P_i \boldsymbol{u} = -j\hbar \frac{\partial \boldsymbol{u}}{\partial x_i} \tag{29}$$

which satisfy Eq. (25). In this case one can show that $E(\lambda, Q_i)$ is given by $E(\lambda, Q_i)\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x})$ for $x_i \leq \lambda$ and $E(\lambda, Q_i)\boldsymbol{u}(\boldsymbol{x}) = 0$ for $x_i > \lambda$; see Ref. 7, p. 131. From this it follows that $|\boldsymbol{u}(\boldsymbol{x})|^2$ is the probability density function of finding the particle at position x if the state is u. In fact, many authors take this result as one of the basic assumptions of quantum mechanics; see Ref. 7, p. 198, and Ref. 14, p. 25. To get the spectral family for P_i one can use the fact that the Fourier transform transforms constant coefficient differential operators into multiplication [see Eq. (10)] so that $P_i/ = \mathscr{F}^{-1} Q_i \mathscr{F}$. Since $\mathscr{F}$ is a unitary operator, $E(\lambda, P_i/) = \mathscr{F}^{-1} E(\lambda, Q_i)\mathscr{F}$. Consequently, if $\boldsymbol{u}$ is the state of the system, then $|\mathscr{F}\boldsymbol{u}(\xi)|^2$ is the probability density of observing a value of ξ for $p/$.

When the Q_i and P_i are given by Eq. (29), Schrödinger's equation becomes

$$j\hbar \frac{d\boldsymbol{u}}{dt} = \mathrm{H}\left(q, -j\hbar \frac{\partial}{\partial x_i}\right) \boldsymbol{u} \tag{30}$$

In the case of a single particle in Cartesian coordinates acted on by a potential $U(\boldsymbol{q})$, this becomes

$$j\hbar \frac{\partial \boldsymbol{u}}{\partial t} = -\frac{\hbar^2}{2m} \Delta \boldsymbol{u} + U(\boldsymbol{q})\boldsymbol{u} \tag{31}$$

To make the formula of Eq. (27) for the solution more explicit, it is necessary to determine $E(\lambda, \mathscr{H})$ more precisely. For example, in the case of a free particle where $U(\boldsymbol{q}) \equiv 0$, one has $\mathscr{H} = \mathscr{F}^{-1} M \mathscr{F}$, where M is the multiplication operator by $^2|\xi|^2/2m$. It follows that $e^{-j\mathscr{H}t/} = \mathscr{F}^{-1} N \mathscr{F}$, where N is the multiplication operator by $\exp(-j|\xi|^2 t/2m)$.

For nonzero potentials $U(\boldsymbol{q})$ it is sometimes possible to find the eigenvalues and eigenfunctions of $\mathscr{H}$ in terms of familiar functions. If this is possible, one says that the system is a *quantum integrable system.* One important system that falls into this category is the one-dimensional harmonic oscillator where $U(\boldsymbol{q}) = K\boldsymbol{q}^2/2$. In this case it can be shown (see Ref. 14, p. 66) that the eigenvalues are $E_n = (n + 1/2)\hbar\omega$ for $n = 0, 1, 2, \ldots$ and $\omega = (K/m)^{1/2}$ and the eigenfunctions are given as $\boldsymbol{u}_n(x) = N_n H_n(\alpha x) \exp(-\alpha^2 x^2)$, where $N_n = (\alpha/\pi^{1/2} 2^n n!)^{1/2}$, $\alpha = (mK/\hbar^2)^{1/4}$, and $H_n(x)$ is the nth Hermite polynomial. These eigenfunctions are a complete orthonormal set. It is not always possible to find a complete orthonormal set consisting of eigenvectors of a given self-adjoint operator. However, often when this is not the case it is still possible to find generalized eigenfunctions and an integral representation that plays a similar role.

In Heisenberg's formulation of quantum mechanics the state vector $\boldsymbol{u}$ describing a system is time independent, and the operators associated with observables vary with time according to the equation

$$\frac{dA}{dt} = \frac{1}{j\hbar}[A, \mathscr{H}]$$

which is just the quantum mechanical analog of Eq. (23) under the correspondence (P2). It can be shown that this is equivalent to Schrodinger's equation; see Ref. 14, pp. 170–171.

Other Methods

In the preceding discussion the solution to the differential equation $d\boldsymbol{u}/dt = A\boldsymbol{u}$ for a self-adjoint operator A in a Hilbert space was given as $\boldsymbol{u}(t) = e^{tA}\boldsymbol{u}(0) = \int^{\infty}_{-\infty} e^{tr}\, dE(r)\boldsymbol{u}(0)$. This construction involving a spectral integral requires A to be self-adjoint. In 1948 E. Hille and K. Yosida constructed the solution to this equation under weaker assumptions, which allow application to a broader class of differential equations. They assumed $\mathrm{Re}(A\boldsymbol{u}, \boldsymbol{u}) \leq M\|\boldsymbol{u}\|^2$ for some constant M independent of u, and the equation $A\boldsymbol{u} - \lambda\boldsymbol{u} = \boldsymbol{f}$ has a solution $\boldsymbol{u}$ for all $\boldsymbol{f}$ in H if $\lambda > M$. (See Ref. 3, chap. 9; Ref. 8, chap. IX; and Ref. 16, chap. XII for details.) Their work has been extended to more general equations of evolution of the form $d\boldsymbol{u}/dt = A(t)\boldsymbol{u}(t) + \boldsymbol{f}(t)$, where the operators $A(t)$ may vary with t; see Ref. 8, chap. XIV.

The length of a vector $\|\boldsymbol{u}\| = (\boldsymbol{u}, \boldsymbol{u})^{1/2}$ expressed in terms of an inner product is only one example of a norm that gives a measure of the length or size of a vector. There are other useful norms in vector spaces that, when applied to linear operators, yield interesting results concerning the solution of differential equations. This forms the basis of the theory of Banach spaces and locally convex linear topological spaces that generalizes the theory of Hilbert spaces (2,3,4,8).

The preceding discussion has been concerned with the theory of linear operators and applies to linear partial differential equations. There is a corresponding theory of nonlinear operators in Hilbert and Banach spaces that applies to nonlinear partial differential equations. See Ref. 8, chap. XIV, and Ref. 16 for information on this.

Acknowledgment

Daoqi Yang would like to express his sincere gratitude to Nan Zhang for helpful conversations.

BIBLIOGRAPHY

1. J. Blank P. Exner M. Havlicek *Hilbert Space Operators in Quantum Physics*, New York: AIP Press, 1994.
2. N. Dunford J. T. Schwartz *Linear Operators, Part I: General Theory; Part 2: Spectral Theory*, New York: Interscience, 1958, 1963.
3. T. Kato *Perturbation Theory for Linear Operators*, 2nd ed., New York: Springer-Verlag, 1976.
4. A. W. Naylor G. R. Sell *Linear Operator Theory in Engineering and Science*, New York: Springer-Verlag, 1994.
5. F. Riesz B. Sz.-Nagy *Functional Analysis*, New York: Ungar, 1955.
6. M. Schechter *Operator Methods in Quantum Mechanics*, New York: Elsevier, 1981.
7. John von Neumann *Mathematical Foundations of Quantum Mechanics*, Princeton, NJ: Princeton Univ. Press, 1955.
8. K. Yosida *Functional Analysis*, 6th ed., New York: Springer-Verlag, 1980.
9. H. L. Royden *Real Analysis*, 2nd ed., New York: Macmillan, 1968.
10. R. Courant D. Hilbert *Methods of Mathematical Physics*, New York: Interscience, 1953, Vol. 1.
11. R. C. Buck *Advanced Calculus*, 2nd ed., New York: McGraw-Hill, 1965.
12. S. Mizohata *The Theory of Partial Differential Equations*, Cambridge, UK: Cambridge Univ. Press, 1973.
13. Herbert Goldstein *Classical Mechanics*, 2nd ed., Reading, MA: Addison-Wesley, 1980.
14. Leonard I. Schiff *Quantum Mechanics*, 3rd ed., New York: McGraw-Hill, 1968.
15. E. Prugovecki *Quantum Mechanics in Hilbert Space*, New York: Academic Press, 1971.

16. Eberhard Zeidler *Nonlinear Functional Analysis and Its Applications, Part I: Fixed Point Theorems; Part II/A: Linear Monotone Operators; Part II/B: Nonlinear Monotone Operators; Part III: Variational Methods and Optimization; Parts IV/V: Applications to Mathematical Physics*, New York: Springer-Verlag, 1986, 1990.

FRANK MASSEY
University of Michigan-Dearborn
DAOQI YANG
Wayne State University

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Hilbert Transforms
Standard Article
Robert E. Bogner[1]
[1]The University of Adelaide, South Australia
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2426
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (449K)

Abstract

The sections in this article are

Definition

Salient Properties

Brief List of Applications

Hilbert Transformer

Origins

Properties

Implementation of Hilbert Transformers

Applications

Further Reading

| | |

Copyright © 1999-2008 All Rights Reserved.

/ /

Homotopy Algorithm for Riccati Equations

Standard Article

Yuzhen Ge[1], Layne T. Watson[2], Dennis S. Bernstein[3], Emmanuel G. Collins, Jr.[4]

[1]Butler University, Indianapolis, IN

[2]Virginia Polytechnic Institute and State University, Blacksburg, VA

[3]University of Michigan, Ann Arbor, MI

[4]Florida A&M/Florida State University, Tallahassee, FL

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2448

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (367K)

Abstract

The sections in this article are

Riccati Equations

Homotopy Algorithm Based on Ly's Formulation

Homotopy Algorithm Based on Overparametrization Formulation

Numerical Algorithms

The Distributed Homotopy Algorithm

Numerical Results and Discussion

Acknowledgment

Keywords: Homotopy method; Riccati equation; LQG controller synthesis; H2/H∞ performance

| | |

Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

HOMOTOPY ALGORITHM FOR RICCATI EQUATIONS

In modern control an H^2 cost function is used to measure the response of a controlled system to wide band disturbances while an H^∞ cost function is used to measure the response to narrow band disturbances and can also be used to account for unstructured uncertainties in the design model. To take into account controller processor limitations, it is also desired to prespecify the order of the controller in the design process. Hence, an important problem in modern control design is the synthesis of fixed-order controllers that are optimized subject to these H^2 and H^∞ constraints. This synthesis problem requires the solution of a challenging optimization problem, and this article reviews a solution approach based on homotopy methods.

There are many approaches to solving both full- and reduced-order linear state equation, quadratic objective function, Gaussian noise (LQG) controller design problems with an H^∞ constraint on disturbance attenuation. The Riccati equation based approach enforces the H^∞ constraint by replacing the covariance Lyapunov equation by a Riccati equation whose solution gives an upper bound on H^2 performance. Numerical algorithms, based on homotopy theory, solve the necessary conditions for a minimum of the upper bound on H^2 performance subject to the H^∞ constraint given by the Riccati equation. A summary of the properties of Riccati equations and numerical algorithms for solving them is also included. The homotopy algorithms are based on a minimal parameter formulation: Ly, Bryson, and Cannon's 2×2 block parametrization. An overparametrization formulation is also proposed. Numerical experiments suggest that the combination of a globally convergent homotopy method and a minimal parameter formulation applied to the upper bound minimization gives excellent results for mixed-norm H^2/H^∞ synthesis. The nonmonotonicity of homotopy zero curves is demonstrated, proving that algorithms more sophisticated than standard continuation are necessary. To achieve high computational performance the homotopy algorithm is also parallelized to run in distributed environments such as a network of Unix workstations or an Intel Paragon parallel computer. Comparing results on the workstations with the results from the Intel Paragon, the study concludes that utilizing Unix workstations can be a very cost-effective approach to shorten computation time. Furthermore, this economical way to achieve high performance computation can easily be realized and incorporated in a practical industrial design environment.

The Riccati equation is central to modern linear-quadratic estimation and control design. Many problems in control analysis and synthesis can be formulated in terms of Riccati equations, with the H^2/H^∞ mixed-norm controller synthesis problem being one of them.

The H^2/H^∞ mixed-norm controller synthesis problem provides the means for simultaneously addressing H^2 and H^∞ performance objectives. In practice, such controllers provide both robust performance (via suboptimal H^2) and robust stability (via H^∞). Hence mixed-norm synthesis provides a technique for trading off performance and robustness, a fundamental objective in control design.

The H^2/H^∞ mixed-norm problem has been addressed in a variety of settings. The Riccati equation based approach was given in (1,2) which utilized an H^2 cost bound as the basis for an auxiliary minimization problem. Necessary conditions for optimality within a full- and reduced-order fixed-structure setting were then used to characterize feedback control gains. These necessary conditions have the form of coupled Riccati equations in both the full- and reduced-order cases. In related work (3,4), the H^2 cost bound in the case of equalized H^2 and

H^∞ performance weights was shown to be equal to an entropy cost functional. The centralized controller was then shown to yield a full-order controller that optimizes the entropy cost.

An additional treatment in (5) using a bounded power characterization of the H^2 norm obtained both necessary and sufficient conditions for optimality. Finally, a convex optimization approach was developed in (6) for the full-order problem.

The purpose of this article is to develop numerical algorithms for solving the Riccati equation based mixed-norm H^2/H^∞ problem addressed in (1,2). Basically, the modified cost function is optimized subject to constraints, including a coupled Riccati equation. The approach here is based upon homotopy methods which have been applied to related fixed-structure problems in (7,8,9,10,11). Using globally convergent homotopy techniques similar to those applied to the combined H^2/H^∞ model reduction problem (8,9,10,11), and using a controller parametrization suggested by Ly, Bryson, and Cannon, results are obtained for the combined H^2/H^∞ full- and reduced-order controller synthesis problems. Another parametrization, the input normal Riccati form, was used in (9) and its details will not be repeated here. However, such controller parametrizations, which use the minimum possible number of parameters, make structural assumptions about the optimal controller which may not be valid in a particular case. Invalidity of these assumptions manifests itself in numerical instability, and failure to converge. An over-parametrization formulation which does not make structural assumptions is also proposed. However, over-parametrization introduces singularity of the homotopy map at the solution and the algorithm may fail for a high dimensional system.

These homotopy methods utilize the solution of a related easily solved problem as the starting point. In the case of full-order H^2/H^∞ control with unequalized weights, the starting point is provided by the standard LQG solution. For the reduced-order problem, the starting point is obtained by constructing a low authority, nearly nonminimal LQG compensator (12).

The theoretical foundation of all probability-one globally convergent homotopy methods is given by the following definition and theorem from differential geometry.

Definition. *Let $U \subset \boldsymbol{R}^m$ and $V \subset \boldsymbol{R}^p$ be open sets, and let ρ $U \times [0, 1) \times V \to \boldsymbol{R}^p$ be a C^2 map. ρ is said to be transversal to zero if the Jacobian matrix $D\rho$ has full rank on $\rho^{-1}(0)$.*

Transversality Theorem. If $\rho(a, \lambda, x)$ is transversal to zero, then for almost all $a \in U$ the map

$$\rho_a(\lambda, x) = \rho(a, \lambda, x)$$

is also transversal to zero, that is, with probability one the Jacobian matrix $D\rho_a(\lambda, x)$ has full rank on ${\rho^{-1}}_a(0)$.

To solve the nonlinear system of equations

$$f(x) = 0$$

where f: $\mathbf{R}^p \to \mathbf{R}^p$ is a C^2 map, the general probability-one homotopy paradigm is to construct a C^2 homotopy map ρ: $U \times [0, 1) \times \mathbf{R}^p \to \mathbf{R}^p$ such that

(1) $\rho(a, \lambda, x)$ is transversal to zero, for each fixed $a \in U$,
(2) $\rho_a(0, x) = \rho(a, 0, x) = 0$ is trivial to solve and has a unique solution x_0,
(3) $\rho_a(1, x) = f(x)$,
(4) the connected component of ${\rho^{-1}}_a(0)$ containing $(0, x_0)$ is bounded.

Then (from the transversality theorem) for almost all $a \in U$ there exists a zero curve γ of ρ_a, along which the Jacobian matrix $D\rho_a$ has rank p, emanating from $(0, x_0)$ and reaching a zero $\bar{x}$ of f at $\lambda = 1$. This zero curve

γ has no bifurcations (i.e., γ is a smooth 1-manifold), and has finite arc length in every compact subset of $(0, 1) \times \mathbf{R}^p$. Furthermore, if $Df(\bar{x})$ is nonsingular, then γ has finite arc length. The complete homotopy paradigm is now apparent: Construct the homotopy map ρ_a and then track its zero curve γ from the known point $(0, x_0)$ to a solution $\bar{x}$ at $\lambda = 1$. ρ_a is called a probability-one homotopy because the conclusions hold almost surely with respect to a, that is, with probability one. Since the vector a, and indirectly the starting point x_0, are essentially arbitrary, an algorithm to follow the zero curve γ emanating from $(0, x_0)$ until a zero $\bar{x}$ of $f(x)$ is reached (at $\lambda = 1$) is legitimately called *globally convergent.*

There is considerable confusion in the control literature over the terms continuation, homotopy, and globally convergent. A careful discussion of the distinct meanings of these terms can be found in (13). Continuation refers to the standard classical technique of solving $\rho(\theta, \bar{\lambda} + \Delta\lambda) = 0$ with fixed $\Delta\lambda > 0$, given a solution $(\bar{\theta}, \bar{\lambda})$: $\rho(\bar{\theta}, \bar{\lambda}) = 0$. It is implicitly assumed that $\theta = \theta(\lambda)$, that is, the zero curve γ of $\rho(\theta, \lambda)$ being tracked in (θ, λ) space is monotone in λ. Other tacit assumptions are that γ does not bifurcate or otherwise contain singularities. The most general *homotopy* methods make no such assumptions, and include mechanisms to deal with bifurcations and turning points. In particular, homotopy methods do not assume that the zero curve γ is monotone in λ, that is, $\theta = \theta(\lambda)$. *Globally convergent* means that the zero curve γ reaches a solution $\bar{\theta}$, $\rho(\bar{\theta}, 1) = 0$, from an arbitrary starting point θ_0, $\rho(\theta_0, 0) = 0$. A continuation or homotopy algorithm is not a priori globally convergent. A particular class of homotopy methods, known as *probability-one homotopy methods,* are provably globally convergent under mild assumptions (14), and their zero curve γ is guaranteed to contain no singularities with probability one. The homotopy algorithms proposed here are examples of probability-one globally convergent homotopy methods; the matrices $A_0, B_0, \ldots$, and the starting point θ_0 defined later play the role of the parameter vector a in the probability-one homotopy theory (13). Computational results for the example in (1) clearly demonstrate the nonmonotonicity in λ and that standard continuation in λ would fail.

The LQG controller synthesis problem with an H^∞ performance bound can be stated as follows: given the nth order stabilizable and detectable plant

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + D_1 w(t) \\ y(t) &= Cx(t) + D_2 w(t) \end{aligned} \tag{1}$$

where $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{l\times n}$, $D_1 \in \mathbf{R}^{n\times p}$, $D_2 \in \mathbf{R}^{l\times p}$, $D_1 D^T{}_2 = 0$, and $w(t)$ is p-dimensional white noise, find a n_cth order dynamic compensator

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t) + B_c y(t) \\ u(t) &= C_c x_c(t) \end{aligned} \tag{2}$$

where $A_c \in \mathbf{R}^{n}c_{\times n}{}^{c}$, $B_c \in {}^{\mathbf{R}}{}_{n}c_{\times l}$, $C_c \in {}^{\mathbf{R}}m_{\times n}{}^{c}$, and $n_c \leq n$, which satisfies the following criteria:

(1) The closed-loop system of Eqs. (1) and (2) is asymptotically stable, that is

$$\tilde{A} = \begin{pmatrix} A & BC_c \\ B_c C & A_c \end{pmatrix}$$

is asymptotically stable;

(2) The $q_\infty \times p$ closed-loop transfer function

$$H(s) \equiv \tilde{E}_\infty (sI_{\tilde{n}} - \tilde{A})^{-1} \tilde{D}$$

from $w(t)$ to

$$z(t) = E_{1\infty}x(t) + E_{2\infty}u(t) \tag{3}$$

where $\tilde{C}_\infty = (E_{1\infty}\ E_{2\infty}C_c)$ ($E_{1\infty} \in R^{q_\infty \times n}$, $E_{2\infty} \in R^{q_\infty \times m}$, $E^T{}_{1\infty}E_{2\infty} = 0$), $\tilde{n} = n + n_c$, and

$$\tilde{D} = \begin{pmatrix} D_1 \\ B_c D_2 \end{pmatrix}$$

satisfies the constraint

$$\|H(s)\|_\infty \leq \gamma \tag{4}$$

where $\gamma > 0$ is a given constant; and

(3) The performance functional

$$J(A_c, B_c, C_c) \equiv \lim_{t\to\infty} \mathscr{E}\left[x^T(t)R_1x(t) + u^T(t)R_2u(t)\right] \tag{5}$$

is minimized, where $\mathscr{E}$ is the expected value, $R_1 = E^T{}_1E_1 \in \mathbf{R}^{n\times n}$ and $R_2 = E^T{}_2E_2 \in \mathbf{R}^{m\times m}$ ($E_1 \in \mathbf{R}^{q\times n}$, $E_2 \in \mathbf{R}^{q\times m}$, $E^T{}_1E_2 = 0$) are, respectively, symmetric positive semidefinite and symmetric positive definite weighting matrices.

The closed-loop system of Eqs. (1, 2, 3) can be written as the augmented system

$$\begin{aligned} \dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{D}w(t) \\ z(t) &= \tilde{E}_\infty \tilde{x} \end{aligned} \tag{6}$$

where

$$\tilde{x} = \begin{pmatrix} x \\ x_c \end{pmatrix}$$

Using this notation and under the condition that $\tilde{A}$ is asymptotically stable, for a given compensator the performance of Eq. (5) is given by

$$J(A_c, B_c, C_c) = \mathrm{tr}\,(\tilde{Q}\tilde{R}) \tag{7}$$

where

$$\tilde{R} = \begin{pmatrix} R_1 & 0 \\ 0 & C_c^T R_2 C_c \end{pmatrix}$$

and $\tilde{Q}$ satisfies the Lyapunov equation

$$\tilde{A}\tilde{Q} + \tilde{Q}\tilde{A}^T + \tilde{V} = 0 \tag{8}$$

with symmetric positive semidefinite $V_1 = D_1 D^T{}_1$, symmetric positive definite $V_2 = D_2 D^T{}_2$, and

$$\tilde{V} = \begin{pmatrix} V_1 & 0 \\ 0 & B_c V_2 B_c^T \end{pmatrix}$$

Lemma 1 1. *Let (A_c, B_c, C_c) be given and assume there exists $\hat{Q} \in \boldsymbol{R}^{\tilde{n}\times\tilde{n}}$ satisfying*

$$\mathscr{Q} \text{ is symmetric and nonnegative definite} \qquad (9)$$

and

$$\tilde{A}\mathscr{Q} + \mathscr{Q}\tilde{A}^T + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty\mathscr{Q} + \tilde{V} = 0 \qquad (10)$$

where

$$\tilde{R}_\infty = \begin{pmatrix} R_{1\infty} & 0 \\ 0 & C_c^T R_{2\infty} C_c \end{pmatrix}$$

$R_{1\infty} = E^T{}_{1\infty}E_{1\infty}$, and $R_{2\infty} = E^T{}_{2\infty}E_{2\infty}$ are symmetric positive semidefinite matrices. Then

$$(\tilde{A}, \tilde{D}) \text{ is stabilizable} \qquad (11)$$

if and only if

$$\tilde{A} \text{ is asymptotically stable}$$

In this case

$$\|H(s)\|_\infty \le \gamma$$
$$\tilde{Q} \le \mathscr{Q} \qquad (\mathscr{Q} - \tilde{Q} \text{ is nonnegative definite})$$

$$\operatorname{tr}\tilde{Q}\tilde{R} \equiv J(A_c, B_c, C_c) \le \mathscr{T}(A_c, B_c, C_c) \equiv \operatorname{tr}\mathscr{Q}\tilde{R}$$

Hence the satisfaction of Eqs. (9) and (10) along with the generic condition of Eq. (11) leads to: (1) closed-loop stability; (2) prespecified H^∞ attenuation; and (3) an upper bound for the H^2 performance criterion.

The auxiliary minimization problem is to determine (A_c, B_c, C_c) that minimizes $J(A_c, B_c, C_c)$ and thus provides a bound for the actual H^2 criterion $J(A_c, B_c, C_c)$.

$(A_c, B_c, C_c, \hat{Q})$ is restricted to the open set $S \equiv \{(A_c, B_c, C_c, \hat{Q})$: $\tilde{A}$ and $\tilde{A} + \gamma^{-2}\hat{Q}\tilde{R}$ are asymptotically stable, $\hat{Q}$ is symmetric positive definite, and (A_c, B_c, C_c) is controllable and observable$\}$.

Note that if $\tilde{A}$ and $\tilde{A} + \gamma^{-2}\hat{Q}\tilde{R}$ are asymptotically stable, Eq. (10) has a unique positive definite solution $\hat{Q}$. The condition on $\hat{Q}$ in the set S is stated for clarity. However, there are no special conditions imposed in the homotopy algorithm to force $\tilde{A}$ and $\tilde{A} + \gamma^{-2}\hat{Q}\tilde{R}$ to be asymptotically stable, nor are such conditions required.

Practical applications often lead to large dense systems of nonlinear equations which are time-consuming to solve on a serial computer. For these systems, parallel processing may be the only feasible means to achieving

solution algorithms with acceptable speed. One economical way of achieving parallelism is to utilize the aggregate power of a network of heterogeneous serial computers. In industrial environments where interactive design is often the practice, the parallel code can be easily incorporated into interactive software such as MATLAB or Mathematica with proper setup of the network computers. To the engineering users the design environment is identical. However, the parallel computations are faster.

The most expensive part of the H^2/H^∞ homotopy algorithm is the computation of the Jacobian matrix, which can be parallelized easily to run across an Ethernet network with little modification of the original sequential code, and which has relatively large task granularity. There is a trade-off between the programming effort and the speedup of the parallel program. To obtain a better speedup, other parts of the homotopy algorithm, such as finding the solution to the Riccati equations and the QR factorization (factorization of a matrix into an orthogonal matrix Q and an upper triangular matrix R) to compute the kernel of the Jacobian matrix, need to be parallelized as well.

In a later section, the homotopy algorithm for H^2/H^∞ controller synthesis is parallelized to run on a network of workstations using *PVM* (parallel virtual machine) and on an Intel Paragon parallel computer, under the philosophy that as few changes as possible are to be made to the sequential code while achieving an acceptable speedup. The parallelized computation is that of the Jacobian matrix, which is carried out in the master-slave paradigm by functional parallelism, that is, each machine computes a different column of the Jacobian matrix with its own data. Unless the Riccati equation solver is parallelized, there is a large amount of data needed for each slave process at each step of the homotopy algorithm. To avoid sending too many large messages across the network or among different nodes on the Intel Paragon, all slave processes repeat part of the computation done by the master process, which therefore decreases the speedup of the parallel computation.

The speedups of the parallel code are compared as the number of workstations increases or as the number of nodes increases on an Intel Paragon and as the size of the problem varies. A reasonable speedup can be achieved using an existing network of workstations compared to that of using an expensive parallel machine, the Intel Paragon. It is demonstrated that for a large problem, the approach of using a network of workstations to achieve parallelism is feasible and practical, and provides an efficient and economical computational method to parallelize a homotopy based algorithm for H^2/H^∞ controller synthesis in a workstation-based interactive design environment.

Riccati Equations

Equation (10) is a Riccati equation. In the numerical algorithm described in the later sections, Riccati Eq. (10) needs to be solved. Some of the known results about Riccati equations are summarized next.

A generalized algebraic Riccati equation can be written as

$$A^T XW + W^T XA - W^T XRXW + V = 0 \qquad (12)$$

where X is the unknown matrix, A, W, R, and V are real square matrices, and V and R are also assumed to be symmetric, with R being positive semidefinite, and W being nonsingular. For some of the applications, V is also assumed to be positive semidefinite.

Since Riccati equations are central to modern control analysis and synthesis, their theoretical properties have been thoroughly studied (15,16,17,18). Conditions that guarantee the existence of a unique symmetric solution may be also found in (19,20,21,22). Several numerical solution techniques have been developed for Riccati equations including eigenvalue methods (23,24,25,26,27,28,29,30), the Chandrasekhar algorithm (31,

32,33), and the matrix sign function technique (34). Software for Riccati equations is widely available and is included in numerous control-design packages for MATLAB and *Mathematica.*

For the given Riccati Eq. (12), the Hamiltonian pencil associated with it is defined as

$$H - \lambda L = \begin{pmatrix} A & -R \\ -V & -A^T \end{pmatrix} - \lambda \begin{pmatrix} W & 0 \\ 0 & W^T \end{pmatrix}$$

Some useful existence conditions are summarized next.

- If V is positive semidefinite, (A, R) is stabilizable, and (A, V) is observable, then there exists a unique symmetric positive semidefinite solution X.
- If V is indefinite, (A, R) is controllable, and the Hamiltonian pencil associated with the equation has no eigenvalues on the imaginary axis, then a unique symmetric solution exists.
- If both V and R are possibly indefinite and the associated Hamiltonian pencil has no pure imaginary eigenvalues, then a unique symmetric solution exists.

The homotopy algorithms described in the following sections require the solution of Eq. (10) at each point along the homotopy curve. Therefore, efficiently solving a Riccati equation is important to achieving high computational speed for the homotopy algorithm. The Schur method and an implementation by Laub (25) is used in our code.

Homotopy Algorithm Based on Ly's Formulation

In optimizing performance with respect to stabilizing controllers of fixed order n_c, it is desirable to consider controller realizations of a specified structure. In this regard there exist a variety of realizations that involve fewer than the $n_c(n_c + m + l)$ parameters appearing in a fully populated parameterization (35,36,37,38,39, 40,41,42). However, as discussed in (35,36,38), realizations that involve a minimal number of independent parameters cannot provide a smooth, global parameterization of all *MIMO* (multiple input multiple output) systems. Specialized parameterizations are also useful for realizing transfer functions of specified classes (40).

In this article we employ the Ly, Bryson, and Cannon parameterization proposed in (42). The following result characterizes the particular class of transfer functions G realized by the Ly, Bryson, and Cannon parameterization.

Proposition 1. *Suppose that G has the minimal realization (A_c, B_c, C_c), where the matrix A_c is similar to a 2×2 block-diagonal matrix where each 2×2 block has distinct eigenvalues, violation of which will lead to ill-conditioned transformation from the given basis to the Ly form (10). There is an additional 1×1 block if n_c is odd. Then there exists a state space basis with respect to which G has a Ly–Bryson–Cannon realization.*

Although this parameterization does not provide a global representation of all transfer functions even in the *SISO* (single input single output) case, it does provide a generic representation which is particularly suited for parametric optimization. Although the Ly–Bryson–Cannon form implicitly assumes somewhat more than that A_c be diagonalizable, computing the two-dimensional invariant subspaces is better conditioned than computing the eigenvectors, which algorithms that assume diagonalizability attempt to do. When the transformation to the Ly–Bryson–Cannon form is not ill conditioned, this particular representation turns out to be very efficient computationally.

Ly et al. (42) introduced a canonical form with $n_c m + n_c l$ parameters. The compensator is represented with respect to a basis such that A_c is a 2×2 block-diagonal matrix (2×2 blocks with an additional 1×1

block if n_c is odd) with 2×2 blocks in the form

$$\begin{pmatrix} 0 & 1 \\ * & * \end{pmatrix}$$

B_c is a full matrix, and

$$C_c = \begin{pmatrix} (C_c)_1 & (C_c)_2 \cdots (C_c)_r \end{pmatrix}$$

where

$$(C_c)_i = \begin{pmatrix} 1 & * & \cdots & * \\ 0 & * & \cdots & * \end{pmatrix}^T$$

$(C_c)_r = (1 * \cdots *)^T$ if n_c is odd.

It is assumed that (A_c, B_c, C_c) is in Ly's form. Let $\mathscr{I}$ be the set of indices of those elements of A_c which are parameters, that is

$$\mathscr{I} \equiv \{(2, 1), (2, 2), \ldots, (n_c, n_c)\}$$

To optimize $J(A_c, B_c, C_c)$ over the open set S under the constraint that symmetric positive definite $\hat{Q}$ satisfies Eq. (10), and (A_c, B_c, C_c) is in Ly's form, the following Lagrangian is formed:

$$\mathscr{L}(A_c, B_c, C_c, \mathscr{P}, \mathscr{Q}) \equiv \operatorname{tr}\left[\mathscr{Q}\tilde{R} + (\tilde{A}\mathscr{Q} + \mathscr{Q}\tilde{A}^T + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty\mathscr{Q} + \tilde{V})\mathscr{P}\right]$$

where $P \in \mathbf{R}^{\tilde{n}\times\tilde{n}}$ is a Lagrange multiplier. Setting $\partial\mathscr{L}/\partial\hat{Q} = 0$ yields

$$0 = (\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)^T\mathscr{P} + \mathscr{P}(\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty) + \tilde{R} \qquad (13)$$

Partition $\hat{Q}, P \in \mathbf{R}^{\tilde{n}\times\tilde{n}}$ as

$$\mathscr{Q} = \begin{pmatrix} \mathscr{Q}_1 & \mathscr{Q}_{12} \\ \mathscr{Q}_{12}^T & \mathscr{Q}_2 \end{pmatrix}, \qquad \mathscr{P} = \begin{pmatrix} \mathscr{P}_1 & \mathscr{P}_{12} \\ \mathscr{P}_{12}^T & \mathscr{P}_2 \end{pmatrix} \qquad (14)$$

The partial derivatives of $\mathscr{L}$ can be computed as

$$\frac{\partial\mathscr{L}}{\partial(A_c)_{ij}} = 2(\mathscr{P}_{12}^T\mathscr{Q}_{12} + \mathscr{P}_2\mathscr{Q}_2)_{ij}, \quad (i, j) \in \mathscr{I}$$

$$\frac{\partial\mathscr{L}}{\partial B_c} = 2\mathscr{P}_2 B_c V_2 + 2(\mathscr{P}_{12}^T\mathscr{Q}_1 + \mathscr{P}_2\mathscr{Q}_{12}^T)C^T$$

$$\frac{\partial\mathscr{L}}{\partial(C_c)_{ij}} = \big(2R_2C_c\mathscr{Q}_2 + 2B^T(\mathscr{P}_1\mathscr{Q}_{12} + \mathscr{P}_{12}\mathscr{Q}_2) + \gamma^{-2}R_{2\infty}C_c$$
$$[(\mathscr{Q}_{12}^T\mathscr{P}_1 + \mathscr{Q}_2\mathscr{P}_{12}^T)\mathscr{Q}_{12}(\mathscr{Q}_{12}^T\mathscr{P}_{12} + \mathscr{Q}_2\mathscr{P}_2)\mathscr{Q}_2]\big)_{ij} \quad i > 1$$

Let $A_f, B_f, C_f, \gamma_f, R_{1f}, R_{2f}, R_{1\infty f}, R_{2\infty f}, V_{1f}$, and V_{2f} denote $A, B, C, \lambda, R_1, R_2, R_{1\infty}, R_{2\infty}, V_1$, and V_2 in the last expression and define $A(\lambda), B(\lambda), C(\lambda), \gamma(\lambda), R_1(\lambda), R_2(\lambda), R_{1\infty}(\lambda), R_{2\infty}(\lambda), V_1(\lambda), V_2(\lambda)$ as

$$\begin{aligned}
A(\lambda) &= A_0 + \lambda(A_f - A_0), \quad B(\lambda) = B_0 + \lambda(B_f - B_0) \\
C(\lambda) &= C_0 + \lambda(C_f - C_0), \quad \gamma(\lambda) = \gamma_0 + \lambda(\gamma_f - \gamma_0) \\
R_1(\lambda) &= R_{1,0} + \lambda(R_{1f} - R_{1,0}), \quad R_2(\lambda) = R_{2,0} + \lambda(R_{2f} - R_{2,0}) \\
R_1\infty(\lambda) &= R_{1\infty,0} + \lambda(R_{1\infty f} - R_{1\infty,0}), \\
R_{2\infty}(\lambda) &= R_{2\infty,0} + \lambda(R_{2\infty f} - R_{2\infty,0}) \\
V_1(\lambda) &= V_{1,0} + \lambda(V_{1f} - V_{1,0}), \quad V_2(\lambda) = V_{2,0} + \lambda(V_{2f} - V_{2,0})
\end{aligned} \tag{15}$$

and denote them by $A, B, C, \gamma, R_1, R_2, R_{1\infty}, R_{2\infty}, V_1$, and V_2 respectively in the following. Let

$$\begin{aligned}
H_{A_c}(\theta, \lambda) &= \frac{1}{2}\frac{\partial \mathscr{L}}{\partial A_c} = \mathscr{P}_{12}^T \mathscr{Q}_{12} + \mathscr{P}_2 \mathscr{Q}_2 \\
H_{B_c}(\theta, \lambda) &= \frac{1}{2}\frac{\partial \mathscr{L}}{\partial \mathscr{B}_c} = \mathscr{P}_2 B_c V_2 + (\mathscr{P}_{12}^T \mathscr{Q}_1 + \mathscr{P}_2 \mathscr{Q}_{12}^T) C^T \\
H_{C_c}(\theta, \lambda) &= \frac{1}{2}\frac{\partial \mathscr{L}}{\partial C_c} = R_2 C_c \mathscr{Q}_2 + B^T(\mathscr{P}_1 \mathscr{Q}_{12} + \mathscr{P}_{12} \mathscr{Q}_2) \\
&\quad + \frac{1}{2}\gamma^{-2} R_{2\infty} C_c [(\mathscr{Q}_{12}^T \mathscr{P}_1 + \mathscr{Q}_2 \mathscr{P}_{12}^T)\mathscr{Q}_{12} \\
&\quad + (\mathscr{Q}_{12}^T \mathscr{P}_{12} + \mathscr{Q}_2 \mathscr{P}_2)\mathscr{Q}_2]
\end{aligned} \tag{16}$$

where in H_{Ac} only those elements corresponding to the parameter elements of A_c are of interest and

$$\theta \equiv \begin{pmatrix} (A_c)_{\mathscr{I}} \\ \mathbf{Vec}\,(B_c) \\ \mathbf{Vec}\,(C_c)_{\mathscr{J}} \end{pmatrix} \tag{17}$$

denotes the independent variables, $\hat{Q}$ and P satisfy respectively Eqs. (10) and (13), $(A_c)_{\mathscr{I}}$ is a vector consisting of those elements in A_c with indices in the set I, that is

$$(A_c)_{\mathscr{I}} = \left((A_c)_{21}, (A_c)_{22}, \ldots, (A_c)_{n_c n_c}\right)^T$$

and $(C_c)\,T$ is the matrix obtained from rows $P = \{2, \ldots, m\}$ of C_c. Vec(P) for a matrix $P \in \mathbf{R}^{p\times q}$ is the concatenation of its columns:

$$\mathbf{Vec}(P) \equiv \begin{pmatrix} P_{\cdot 1} \\ P_{\cdot 2} \\ \vdots \\ P_{\cdot q} \end{pmatrix} \in \mathbf{R}^{pq}$$

The most natural homotopy map, given the embeddings in Eq. (15), is defined as

$$\rho(\theta,\lambda)=\begin{pmatrix}[H_{A_c}(\theta,\lambda)]_{\mathscr{V}}\\ \mathbf{Vec}\,[H_{B_c}(\theta,\lambda)]\\ \mathbf{Vec}\,[H_{C_c}(\theta,\lambda)]_{\mathscr{W}}\end{pmatrix} \tag{18}$$

and its Jacobian matrix is

$$D\rho(\theta,\lambda)=\big(D_\theta\rho(\theta,\lambda),D_\lambda\rho(\theta,\lambda)\big)$$

In practice it may be difficult to find the initial point θ_0 such that $\rho(\theta_0,0)=0$. A somewhat more artificial (lacking a physical interpretation) homotopy then, letting θ_0 be the chosen initial point, is the Newton homotopy map defined as

$$\overline{\rho}(\theta,\lambda)=\rho(\theta,\lambda)-(1-\lambda)\rho(\theta_0,0)$$

which will give rise to an extra term $\rho(\theta_0,0)$ in $D_\lambda\bar{\rho}(\theta,\lambda)$. To guarantee a full rank Jacobian matrix along the whole homotopy zero curve except possibly at the solution corresponding to $\lambda=1$, define the homotopy map to be

$$\hat{\rho}(\theta,\lambda)=\lambda\rho(\theta,\lambda)+(1-\lambda)(\theta-\theta_0) \tag{19}$$

The Jacobian matrix of $\bar{\rho}$ is given by

$$\begin{aligned}D\hat{\rho}(\theta,\lambda)=\big(&\lambda D_\theta\rho(\theta,\lambda)+(1-\lambda)I,\rho(\theta,\lambda)\\&+\lambda D_\lambda\rho(\theta,\lambda)-(\theta-\theta_0)\big)\end{aligned}$$

In the following, the homotopy map of Eq. (18) is assumed for the full-order problem and Eq. (19) is assumed for the reduced-order case since the reduced-order initialization scheme produces a singular starting point if Eq. (18) is used. Define

$$\begin{aligned}\hat{H}_{A_c}(\mathscr{P}^{(j)},\mathscr{Q}^{(j)})=&\mathscr{P}_{12}^{T(j)}\mathscr{Q}_{12}+\mathscr{P}_{12}^{T}\mathscr{Q}_{12}^{(j)}+\mathscr{P}_2^{(j)}\mathscr{Q}_2+\mathscr{P}_2\mathscr{Q}_2^{(j)}\\
\hat{H}_{B_c}(\mathscr{P}^{(j)},\mathscr{Q}^{(j)})=&\mathscr{P}_2^{(j)}B_cV_2+(\mathscr{P}_{12}^{T(j)}\mathscr{Q}_1+\mathscr{P}_{12}^{T}\mathscr{Q}_1^{(j)}\\
&+\mathscr{P}_2^{(j)}\mathscr{Q}_{12}^{T}+\mathscr{P}_2\mathscr{Q}_{12}^{T(j)})C^T\\
\hat{H}_{C_c}(\mathscr{P}^{(j)},\mathscr{Q}^{(j)})=&R_2C_c\mathscr{Q}_2^{(j)}+B^T(\mathscr{P}_1^{(j)}\mathscr{Q}_{12}+\mathscr{P}_1\mathscr{Q}_{12}^{(j)}\\
&+\mathscr{P}_{12}^{(j)}\mathscr{Q}_2+\mathscr{P}_{12}\mathscr{Q}_2^{(j)})\\
&+\frac{\gamma^{-2}}{2}R_{2\infty}C_c\big[(\mathscr{Q}_{12}^{T(j)}\mathscr{P}_1+\mathscr{Q}_{12}^{T}\mathscr{P}_1^{(j)}\\
&+\mathscr{Q}_2^{(j)}\mathscr{P}_{12}^{T}+\mathscr{Q}_2\mathscr{P}_{12}^{T(j)})\mathscr{Q}_{12}\\
&+(\mathscr{Q}_{12}^{T(j)}\mathscr{P}_{12}+\mathscr{Q}_{12}^{T}\mathscr{P}_{12}^{(j)}+\mathscr{Q}_2^{(j)}\mathscr{P}_2\\
&+\mathscr{Q}_2\mathscr{P}_2^{(j)})\mathscr{Q}_2+(\mathscr{Q}_{12}^{T}\mathscr{P}_1\\
&+\mathscr{Q}_2\mathscr{P}_{12}^{T})\mathscr{Q}_{12}^{(j)}+(\mathscr{Q}_{12}^{T}\mathscr{P}_{12}+\mathscr{Q}_2\mathscr{P}_2)\mathscr{Q}_2^{(j)}\big]\end{aligned} \tag{20}$$

where the superscript (j) means $\partial/\partial\theta_j$, Using these definitions, we have for $\theta_j = (A_c)_{kl}$, where $(k,l) \in \mathcal{I}$,

$$\begin{aligned}
\frac{\partial H_{A_c}}{\partial (A_c)_{kl}} &= \hat{H}_{A_c}(\mathcal{P}^{(j)}, \mathcal{Q}^{(j)}) \\
\frac{\partial H_{B_c}}{\partial (A_c)_{kl}} &= \hat{H}_{B_c}(\mathcal{P}^{(j)}, \mathcal{Q}^{(j)}) \\
\frac{\partial H_{C_c}}{\partial (A_c)_{kl}} &= \hat{H}_{C_c}(\mathcal{P}^{(j)}, \mathcal{Q}^{(j)})
\end{aligned} \tag{21}$$

for $\theta_j = (B_c)_{kl}$,

$$\begin{aligned}
\frac{\partial H_{A_c}}{\partial (B_c)_{kl}} &= \hat{H}_{A_c}(\mathcal{P}^{(j)}, \mathcal{Q}^{(j)}) \\
\frac{\partial H_{B_c}}{\partial (B_c)_{kl}} &= \hat{H}_{B_c}(\mathcal{P}^{(j)}, \mathcal{Q}^{(j)}) + \mathcal{P}_2 E^{(k,l)} V_2 \\
\frac{\partial H_{C_c}}{\partial (B_c)_{kl}} &= \hat{H}_{C_c}(\mathcal{P}^{(j)}, \mathcal{Q}^{(j)})
\end{aligned} \tag{22}$$

and for $\theta_j = (C_c)_{kl}$, where $k > 1$,

$$\begin{aligned}
\frac{\partial H_{A_c}}{\partial (C_c)_{kl}} &= \hat{H}_{A_c}(\mathcal{P}^{(j)}, \mathcal{Q}^{(j)}) \\
\frac{\partial H_{B_c}}{\partial (C_c)_{kl}} &= \hat{H}_{B_c}(\mathcal{P}^{(j)}, \mathcal{Q}^{(j)}) \\
\frac{\partial H_{C_c}}{\partial (C_c)_{kl}} &= \hat{H}_{C_c}(\mathcal{P}^{(j)}, \mathcal{Q}^{(j)}) + R_2 E^{(k,l)} \mathcal{Q}_2 + \frac{\gamma^{-2}}{2} R_{2\infty} E^{(k,l)} Y
\end{aligned} \tag{23}$$

where

$$Y = (\mathcal{Q}_{12}^T \mathcal{P}_1 + \mathcal{Q}_2 \mathcal{P}_{12}^T)\mathcal{Q}_{12} + (\mathcal{Q}_{12}^T \mathcal{P}_{12} + \mathcal{Q}_2 \mathcal{P}_2)\mathcal{Q}_2 \tag{24}$$

and $E^{(k,l)}$ is a matrix of the appropriate dimension of which the only nonzero element is $e_{kl} = 1$. $P^{(j)}$ and $\hat{Q}^{(j)}$ can be obtained by solving the Lyapunov equations

$$\begin{aligned}
0 &= (\tilde{A} + \gamma^{-2}\mathcal{Q}\tilde{R}_\infty)\mathcal{Q}^{(j)} + \mathcal{Q}^{(j)}(\tilde{A} + \gamma^{-2}\mathcal{Q}\tilde{R}_\infty)^T + \tilde{V}^{(j)} \\
&\quad + \tilde{A}^{(j)}\mathcal{Q} + \mathcal{Q}\tilde{A}^{T(j)} + \gamma^{-2}\mathcal{Q}\tilde{R}_\infty^{(j)}\mathcal{Q} \\
0 &= (\tilde{A} + \gamma^{-2}\mathcal{Q}\tilde{R}_\infty)^T\mathcal{P}^{(j)} + \mathcal{P}^{(j)}(\tilde{A} + \gamma^{-2}\mathcal{Q}\tilde{R}_\infty) + \tilde{R}^{(j)} \\
&\quad + (\tilde{A}^{(j)} + \gamma^{-2}\mathcal{Q}^{(j)}\tilde{R}_\infty \\
&\quad + \gamma^{-2}\mathcal{Q}\tilde{R}_\infty^{(j)})^T\mathcal{P} + \mathcal{P}(\tilde{A}^{(j)} + \gamma^{-2}\mathcal{Q}^{(j)}\tilde{R}_\infty + \gamma^{-2}\mathcal{Q}\tilde{R}_\infty^{(j)})
\end{aligned} \tag{25}$$

Similarly for λ, using a dot to denote $\partial/\partial\lambda$,

$$\begin{aligned}
\frac{\partial H_{A_c}}{\partial \lambda} &= \hat{H}_{A_c}(\dot{\mathscr{P}}, \dot{\mathscr{Q}}) \\
\frac{\partial H_{B_c}}{\partial \lambda} &= \hat{H}_{B_c}(\dot{\mathscr{P}}, \dot{\mathscr{Q}}) + \mathscr{P}_2 B_c \dot{V}_2 + (\mathscr{P}_{12}^T \mathscr{Q}_1 + \mathscr{P}_2 \mathscr{Q}_{12}^T)\dot{C}^T \\
\frac{\partial H_{C_c}}{\partial \lambda} &= \hat{H}_{C_c}(\dot{\mathscr{P}}, \dot{\mathscr{Q}}) + R_2 C_c \mathscr{Q}_2 + \dot{B}^T(\mathscr{P}_1 \mathscr{Q}_{12} + \mathscr{P}_{12}\mathscr{Q}_2) \\
&\quad + \frac{1}{2}\gamma^{-2}\dot{R}_{2\infty} C_c Y - \gamma^{-3}\dot{\gamma} R_{2\infty} C_c Y
\end{aligned} \tag{26}$$

where $\dot{P}$ and $\dot{Q}$ are obtained by solving

$$\begin{aligned}
0 &= (\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)\dot{\mathscr{Q}} + \dot{\mathscr{Q}}(\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)^T + \dot{\tilde{V}} + \dot{\tilde{A}}\mathscr{Q} \\
&\quad + \mathscr{Q}\dot{\tilde{A}}^T + \gamma^{-2}\mathscr{Q}\dot{\tilde{R}}_\infty\mathscr{Q} - 2\gamma^{-3}\dot{\gamma}\mathscr{Q}\tilde{R}_\infty\mathscr{Q} \\
0 &= (\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty)^T\dot{\mathscr{P}} + \dot{\mathscr{P}}(\tilde{A} + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty) + \dot{\tilde{R}} \\
&\quad + (\dot{\tilde{A}} + \gamma^{-2}\dot{\mathscr{Q}}\tilde{R}_\infty + \gamma^{-2}\mathscr{Q}\dot{\tilde{R}} + \infty - 2\gamma^{-3}\dot{\gamma}\mathscr{Q}\tilde{R}_\infty)^T\mathscr{P} \\
&\quad + \mathscr{P}(\dot{\tilde{A}} + \gamma^{-2}\dot{\mathscr{Q}}\tilde{R}_\infty + \gamma^{-2}\mathscr{Q}\dot{\tilde{R}}_\infty - 2\gamma^{-3}\dot{\gamma}\mathscr{Q}\tilde{R}_\infty)
\end{aligned} \tag{27}$$

Homotopy Algorithm Based on Overparametrization Formulation

The parametrization in the previous section, the Ly form, is minimal in the sense that it uses the minimum possible number of parameters, $n_c(m + l)$, to describe the controller. However, it also assumes a particular structure for the controller that may not be satisfied by the optimal controller, or even if it is, that structure may be ill conditioned near the optimum. The formulation in this section makes no assumptions whatsoever on the controller structure, treating all the components of (A_c, B_c, C_c) as independent variables.

To optimize $T(A_c, B_c, C_c)$ over the open set S under the constraint that symmetric positive definite $\hat{Q}$ satisfies Eq. (10), the following Lagrangian is formed:

$$\mathscr{L}(A_c, B_c, C_c, \mathscr{P}, \mathscr{Q}) \equiv \operatorname{tr}\left[\mathscr{Q}\tilde{R} + (\tilde{A}\mathscr{Q} + \mathscr{Q}\tilde{A}^T + \gamma^{-2}\mathscr{Q}\tilde{R}_\infty\mathscr{Q} + \tilde{V})\mathscr{P}\right]$$

where $P \in \mathbf{R}^{\tilde{n}\times\tilde{n}}$ is a Lagrange multiplier. Setting $\partial\mathscr{L}/\partial\hat{Q} = 0$ yields Eq. (13). Partition $\hat{Q}, P \in \mathbf{R}^{\tilde{n}\times\tilde{n}}$ as in Eq. (14). The partial derivatives of $\mathscr{L}$ can be computed as

$$\begin{aligned}
\frac{\partial \mathscr{L}}{\partial A_c} &= 2(\mathscr{P}_{12}^T\mathscr{Q}_{12} + \mathscr{P}_2\mathscr{Q}_2) \\
\frac{\partial \mathscr{L}}{\partial B_c} &= 2\mathscr{P}_2 B_c V_2 + 2(\mathscr{P}_{12}^T\mathscr{Q}_1 + \mathscr{P}_2\mathscr{Q}_{12}^T)C^T \\
\frac{\partial \mathscr{L}}{\partial C_c} &= 2R_2 C_c \mathscr{Q}_2 + 2B^T(\mathscr{P}_1\mathscr{Q}_{12} + \mathscr{P}_{12}\mathscr{Q}_2) \\
&\quad + \gamma^{-2}R_{2\infty}C_c\left[(\mathscr{Q}_{12}^T\mathscr{P}_1 + \mathscr{Q}_2\mathscr{P}_{12}^T)\mathscr{Q}_{12} + (\mathscr{Q}_{12}^T\mathscr{P}_{12} + \mathscr{Q}_2\mathscr{P}_2)\mathscr{Q}_2\right]
\end{aligned}$$

Let A_f, B_f, Cf, γ_f, R_{1f}, R_{2f}, $R_{1\infty f}$, $R_{2\infty f}$, V_{1f}, and V_{2f} denote A, B, C, λ, R_1, R_2, $R_{1\infty}$, $R_{2\infty}$, V_1, and V_2 in the last expression and define $A(\lambda)$, $B(\lambda)$, $C(\lambda)$, $\gamma(\lambda)$, $R_1(\lambda)$, $R_2(\lambda)$, $R_{1\infty}(\lambda)$, $R_{2\infty}(\lambda)$, $V_1(\lambda)$, and $V_2(\lambda)$ as in Eq. (15) and denote them by A, B, C, γ, R_1, R_2, $R_{1\infty}$, $R_{2\infty}$, V_1, and V_2 respectively in the following.

Define $H_{Ac}(\theta, \lambda)$, $H_{Bc}(\theta, \lambda)$, and $H_{Cc}(\theta, \lambda)$ as in Eq. (16) where

$$\theta \equiv \begin{pmatrix} \mathbf{Vec}\ (A_c) \\ \mathbf{Vec}\ (B_c) \\ \mathbf{Vec}\ (C_c) \end{pmatrix} \tag{28}$$

denotes the independent variables, and $\hat{Q}$ and$P$satisfy respectively Eq. (10) and Eq. (13). Vec applied to a matrix is a column vector obtained by concatenating the columns of the matrix.

Define

$$\rho(\theta, \lambda) = \begin{pmatrix} \mathbf{Vec}\ [H_{A_c}(\theta, \lambda)] \\ \mathbf{Vec}\ [H_{B_c}(\theta, \lambda)] \\ \mathbf{Vec}\ [H_{C_c}(\theta, \lambda)] \end{pmatrix} \tag{29}$$

whose Jacobian matrix is

$$D\rho(\theta, \lambda) = \big(D_\theta \rho(\theta, \lambda), D_\lambda \rho(\theta, \lambda)\big)$$

Note that θ in Eq. (29) has ${n^2}_c + n_c m + n_c l$ components, more than the minimal number $n_c m + n_c l$ needed. Because of this over-parametrization, the Jacobian matrix of ρ is seriously rank deficient. To remedy this severe rank deficiency, the homotopy map is defined as

$$\hat{\rho}(\theta, \lambda) = \lambda \rho(\theta, \lambda) + (\mathbf{1} - \lambda)(\theta - \theta_0) \tag{30}$$

which guarantees a full rank Jacobian matrix along the entire homotopy zero curve except possibly at the solution (corresponding to $\lambda = 1$). The Jacobian matrix of $\bar{\rho}$ is given by

$$\begin{aligned} &D\hat{\rho}(\theta, \lambda) \\ &\quad = \big(\lambda D_\theta \rho(\theta, \lambda) + (\mathbf{1} - \lambda) I,\ \rho(\theta, \lambda) + \lambda D_\lambda \rho(\theta, \lambda) - (\theta - \theta_0)\big) \end{aligned} \tag{31}$$

To find $D_\theta \rho(\theta, \lambda)$, define the auxiliary matrices $\hat{H}_{Ac}(P^{(j)}, \hat{Q}^{(j)})$, $\hat{H}_{Bc}(P^{(j)}, \hat{Q}^{(j)})$, and $\hat{H}_{Cc}(P^{(j)}, \hat{Q}^{(j)})$ as in Eq. (20). Using these definitions, we have Eqs. (21), (22), (23) for $\theta_j = (A_c)_{kl}$, $\theta_j = (B_c)_{kl}$, and $\theta_j = (C_c)_{kl}$ respectively. $P^{(j)}$ and $\hat{Q}^{(j)}$ can be obtained by solving the Lyapunov Eq. (25). Similarly we have Eq. (26) for λ and $\dot{p}$ and $\hat{Q}$ are obtained by solving Eq. (27).

Numerical Algorithms

Choose r_0, the initial γ, so that ${\gamma^{-2}}_0$ is approximately zero. The initial point $(\theta, \lambda) = (\theta_0, 0)$ is chosen so that it satisfies $\rho(\theta_0, 0) = 0$ and the triple $((A_c)_0, (B_c)_0, (C_c)_0)$ is in the respective form for each homotopy.

It is well known that the full-order LQG compensator of Eq. (2) for the plant Eq. (1) minimizing the steady-state quadratic performance functional Eq. (5) is given by:

$$A_c = A - \Sigma P - Q\bar{\Sigma} \tag{32}$$

$$B_c = QC^T V_2^{-1}, \quad C_c = -R_2^{-1} B^t P \tag{33}$$

where $\Sigma \equiv BR^{-1}{}_2 B^T$, $\Sigma \equiv C^T V^{-1}{}_2 C$, and P and Q are the unique, symmetric, positive semidefinite solutions respectively, of

$$\begin{aligned} 0 &= A^T P + PA + R_1 - P\Sigma P \\ 0 &= AQ + QA^T + V_1 - Q\bar{\Sigma}Q \end{aligned} \tag{34}$$

Full-Order Initialization. The initial point for the full-order problem can be chosen as follows:

(1) Solve for Q and P from Eq. (34) and obtain $((\hat{A}_c)_0, (\hat{B}_c)_0, (\hat{C}_c)_0)$ from Eqs. (32) and (33).
(2) Transform the triple $((\hat{A}_c)_0, (\hat{B}_c)_0, (\hat{C}_c)_0)$ to Ly's form for the Ly form homotopy, and build θ_0 as described in Eq. (17) and Eq. (28) for the respective homotopies.

Reduced-Order Initization. The initialization scheme for the reduced-order problem is more complicated since a closed form expression for the reduced-order H^2 LQG compensator does not exist. For a given system $(\bar{A}, \bar{B}, \bar{C})$, and matrices $\bar{R}_1, \bar{R}_2, \bar{R}_{1\infty}, \bar{R}_{2\infty}, \bar{V}_1$, and $\bar{V}_2$, the reduced order starting point is chosen using a method in (12) which can be summarized as:

(1) Compute the real Schur decomposition of $\bar{A}$ so that $\bar{A} = UAU^t$,

$$A \equiv \begin{pmatrix} A_1 & 0 \\ A_{21} & A_2 \end{pmatrix}$$

where $A_1 \in R^{n_c \times n_c}$, and transform $\bar{B}, \bar{C}, \bar{R}_1, \bar{V}_1, \bar{R}_{1\infty}$ so that $B = U\bar{B}$, $C = \bar{C}U^t$, $R_1 = U\bar{R}_1 U^t$, $V_1 = U\bar{V}_1 U^t$, $R_{1\infty} = U\bar{R}_{1\infty}U^t$ and let $R_2 = \bar{R}_2$, $V_2 = \bar{V}_2$, $R_{2\infty} = \bar{R}_{2\infty}$.
(2) If A is not asymptotically stable, modify either diagonal elements or 2×2 diagonal blocks of A so that it is asymptotically stable and call this modified matrix A_0. Note that this step can always be done easily. For example, a diagonal matrix can be added to A_0 until the sum is asymptotically stable.
(3) Take $B_0 = B$, $C_0 = C$, $R_{2,0} = R_{2f} \equiv R_2$, $R_{2\infty,0} = R_{2\infty \mathrm{f}} \equiv R_{2\infty}$, $V_{1,0} = V_{1f} \equiv V_1$, $V_{2,0} = \beta V_{2f} \equiv \beta V_2$, $\beta \gg 0$, and

$$R_{1,0} = \begin{pmatrix} (R_1)_1 & 0 \\ 0 & 0 \end{pmatrix}$$

where $(R_1)_1$ is the leading $n_c \times n_c$ block of $R_{1f} \equiv R_1$.
(4) Solve

$$\begin{aligned} 0 &= A_0^T P + PA_0 + R_{1,0} - P\Sigma_0 P \\ 0 &= A_0 Q + QA_0^T + V_{1,0} - Q\bar{\Sigma}_0 Q \end{aligned}$$

for symmetric, positive semidefinite P and Q, where $\Sigma_0 \equiv B_0 R^{-1}{}_{2,0} B^T{}_0$, and $\bar{\Sigma}_0 \equiv C^T{}_0 V^{-1}{}_{2,0} C_0$.

(5) Obtain (A_c, B_c, C_c) from $A_c = A_0 - \Sigma_0 P - Q\bar{\Sigma}_0$, $B_c = QC^T{}_0 V^{-1}{}_{2,0}$, $C_c = -R^{-1}{}_{2,0} B^T{}_0 P$.

(6) Solve

$$0 = (A_0 - Q\bar{\Sigma}_0)^T \hat{P} + \hat{P}(A_0 - Q\bar{\Sigma}_0) + P\Sigma_0 P$$
$$0 = (A_0 - \Sigma_0 P)\hat{Q} + \hat{Q}(A_0 - \Sigma_0 P)^T + Q\bar{\Sigma}_0 Q$$

for symmetric, positive semidefinite $\hat{P}$ and $\hat{Q}$.

Following (43), obtain the reduced-order compensator starting point from (A_c, B_c, C_c), $\hat{P}$, $\hat{Q}$, and n_c, as follows:

(1) Compute the Cholesky decomposition of (assumed positive definite) $\hat{P}$ and $\hat{Q}$,that is $\hat{P} = L_{\hat{P}} L_{\hat{P}}^T$, $\hat{Q} = L_{\hat{Q}} L_{\hat{Q}}^T$.

(2) Compute the singular value decomposition of $L_{\hat{P}}^T L_{\hat{Q}}$, that is, $L_{\hat{P}}^T L_{\hat{Q}} = U\Omega V^T$.

(3) Let $T = L_{\hat{Q}} V \Omega^{-1/2}$, $T^{-1} = \Omega^{-1/2} U^T L_{\hat{P}}^T$.

(4) Let $\bar{A}_c = T^{-1} A_c T$, $\bar{B}_c = T^{-1} B_c$, and $\bar{C}_c = C_c T$ so that

$$\bar{A}_c = {}^{n_c}\{\overbrace{\begin{pmatrix} (\bar{A}_c)_1 & (\bar{A}_c)_{12} \\ (\bar{A}_c)_{21} & (\bar{A}_c)_2 \end{pmatrix}}^{n_c} \quad \bar{B}_c = {}^{n_c}\{\begin{pmatrix} (\bar{B}_c)_1 \\ (\bar{B}_c)_2 \end{pmatrix}$$
$$\bar{C}_c = \overbrace{\begin{pmatrix} (\bar{C}_c)_1 & (\bar{C}_c)_2 \end{pmatrix}}^{n_c}$$

The starting point θ_0 for the reduced order problem is chosen using $((\bar{A}_c)_1, \bar{B}_c)_1, (\bar{C}_c)_1)$, with the construction in Eq. (17) and Eq. (28) for the respective homotopies.

The main idea of choosing the initial point is to find an approximate H^2 solution as the initial point so that the corresponding γ is very big. As λ increases to 1, γ goes to the given value. Computationally, if choosing $\gamma_0 = 10^5$ and $\gamma_0 = 10^6$ lead to the same solution, then the initial γ_0 can be chosen as 10^5. If γ is too small for the existence of a solution, the Riccati solver fails. In other words, it is impossible to obtain a symmetric and positive definite solution $\hat{Q}$ from Eq. (10) in this situation.

Homotopy Zero Curve Tracking. Once the initial point is chosen, the rest of the computation is as follows:

(1) Set $\lambda := 0$, $\theta := \theta_0$.

(2) Calculate $\tilde{R}$, $\tilde{R}_\infty$, $\tilde{V}$, and compute $\hat{Q}$ and$P$according to Eqs. (10) and (13).

(3) Evaluate the homotopy map $\rho(\theta, \lambda)$ or $\hat{\rho}(\theta, \lambda)$ and the Jacobian of the homotopy map $D\rho(\theta, \lambda)$ or $D\hat{\rho}(\theta, \lambda)$.

(4) Predict the next point $Z^{(0)} = (\theta^{(0)}, \lambda^{(0)})$ on the homotopy zero curve using, for example, a Hermite cubic interpolant.

(5) For k := 0, 1, 2, ··· until convergence do

$$Z^{(k+1)} = Z^{(k)} - \left[D\rho\left(Z^{(k)}\right)\right]^{\dagger} \rho\left(Z^{(k)}\right)$$

where $[D\rho(Z)]^{\dagger}$ is the Moore–Penrose inverse of $D\rho(Z)$. Let $(\theta_1, \lambda_1) = \lim_{k\to\infty} Z^{(k)}$.

(6) If $\lambda_1 < 1$, then set $\theta := \theta_1$, $\lambda := \lambda_1$, and so to step 2.

(7) If $\lambda_1 \geq 1$, compute the solution $\bar{\theta}$ at $\lambda = 1$.

For the over-parametrization formulation homotopy, because of the singularity at $\lambda = 1$, step 7 is replaced by:

(1) If $\lambda_1 \geq 1$, use the last point $(\tilde{\lambda}, \tilde{\lambda})$ with $\tilde{\lambda} < 1$ to redefine the homotopy map with $\theta_0 = \tilde{\lambda}$.

(2) Redo steps 1–6 until $\lambda \geq 1$.

(3) Use Hermite polynomial interpolation to obtain the solution at $\lambda = 1$.

The Distributed Homotopy Algorithm

In the preceding algorithm, step 2 involves solving one Riccati equation and one Lyapunov equation. The Riccati equation is solved using Laub's Schur method (25). The algorithm of Bartels and Stewart (44) is applied to solve the Lyapunov equation. Although both algorithms are $O((n + n_c)^3)$, the Riccati equation, being more complicated, takes much more *CPU* (central processing unit) time to solve. Once $\hat{Q}$ and P are obtained, the homotopy map $\bar{\rho}$ is formed by matrix multiplication operations.

The major part of the computation in step 3 is that of the Jacobian matrix. The number of variables including λ in this formulation is $n_c(m + l) + 1$. Each column of the Jacobian matrix corresponds to the derivative of the homotopy map with respect to one variable and requires the solution of two Lyapunov equations (9). Therefore, the time complexity of the Jacobian matrix computation is $O(n_c(m + l)(n + n_c)^3)$. The Bartels and Stewart algorithm finds the real Schur form of $\tilde{A}$ or $\tilde{A}^T$ depending on the Lyapunov equation. At each step along the homotopy path unnecessary factorization can be avoided if the previous factorization results from the computation of $\bar{\rho}$ and $D_\lambda \hat{\rho}$ are used.

Our goal of distributed computation is to make use of the existing code and to achieve reasonable parallel efficiency economically. The only part of the algorithm that is parallelized is the Jacobian matrix computation in step 3. To utilize existing computer resources such as a network of workstations, the software package PVM (parallel virtual machine) is used to provide the distributed computing capabilities.

The parallel algorithm follows the master-slave paradigm. The master sends the index of the column of the Jacobian matrix to be computed to a slave. The slave computes the corresponding column of the Jacobian, sends the column back to the master, and waits for the next index from the master to arrive. After receiving a column of the Jacobian, the master sends another index to the idle slave. In the implementation for the Intel Paragon, asynchronous send, which sends a message without waiting for completion, is used whenever possible to speed up the communication.

When the algorithm is implemented on a network of workstations, the modification to the original sequential source code consists of three parts: the first one is to spawn slave processes and set up the communication links between the master and the slaves; the second is to extract a slave program from the original code and at the same time simplify the master program; the last is to add a mechanism to guarantee correct communication between master and slaves. The first part consists of standard PVM operations, while the second is more problem oriented. To decrease communication, each slave process repeats part of the computation of $\bar{\rho}$ and $D_\lambda \hat{\rho}$ so that $\hat{Q}$ and P are not sent through the network. There is no loss of efficiency since the master is also computing the same quantities. The slave program consists of mainly the original subroutines with additional code for communication.

For the implementation on the Intel Paragon, the modification of the original code is even simpler. There is no need for a separate slave program if control statements use node identification properly. The parent process,

which runs on an Intel Paragon, always gets node number 0, while other nodes are numbered 1 and higher. The statement *if node_number == 0* precedes the code that is to be executed by the master, and an *else* following the previous master code will precede the code to be executed by the slave. The remaining modification to the original code is similar to the implementation using PVM. Asynchronous send is used whenever possible. A *wait* is used later when the data is needed, to ensure correct communication between the master and the slaves.

Numerical Results and Discussion

The following systems are solved by the homotopy algorithms discussed in the previous sections. The homotopy curve tracking was done with HOMPACK (14).

The first system, formulated in Ref. 45 and studied in Ref. 1, is given by

$$A = \begin{pmatrix} -0.161 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6.004 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -0.5822 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -9.9835 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -0.4073 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -3.982 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ 0 \\ 0.0064 \\ 0.00235 \\ 0.0713 \\ 1.0002 \\ 0.1045 \\ 0.9955 \end{pmatrix} \qquad C^T = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$E_1 = 10^{-3} \begin{pmatrix} 0 & 0 & 0 & 0 & 0.55 & 11 & 1.32 & 18 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad E_{1\infty} = E_1$$

$$E_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad E_{2\infty} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad D_1 = (B \quad 0), \quad D_2 = (0 \quad 1)$$

For a given initial (A_c, B_c, C_c), γ is lowered until a solution cannot be found anymore. The smallest γ for which a solution can still be found is $\gamma_{\min}$. For the full-order (i.e., 8th order) problem, the solutions of the auxiliary minimization problem are obtained for $\gamma \geq \gamma_{\min} \equiv 0.481$ using the Ly form homotopy approach. For $\gamma < \gamma_{\min} = 0.481$, the Riccati equation solver fails and therefore no solution can be found. In Fig. 1, $\|H(s)\|_\infty$ is plotted against J. The ratio of $\|H(s)\|_\infty$ at $\gamma = \gamma_{\min}$ to that at $\gamma = \infty$ is 0.33, which indicates that there is about 67% improvement in the H^∞ performance of the compensator with $\gamma = \gamma_{\min}$ over the compensator without the H^∞ constraint.

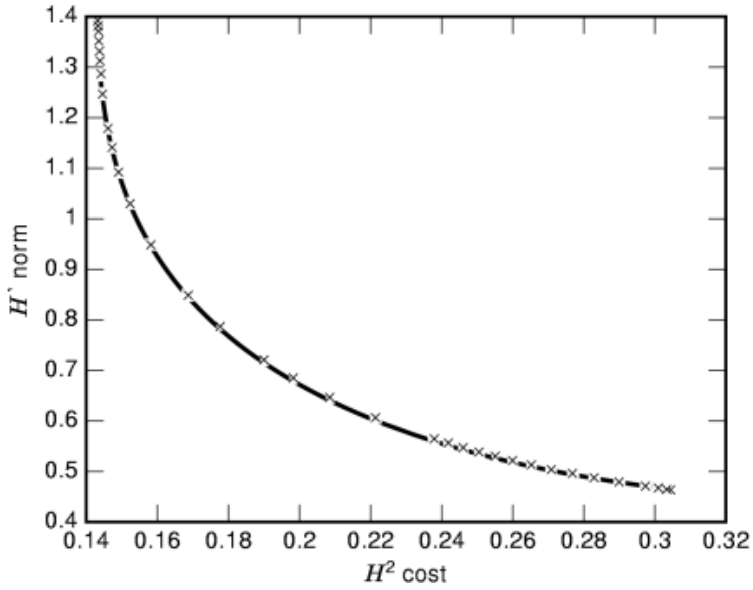


Fig. 1. $\|H(s)\|_\infty$ versus J for $n_c = n$.

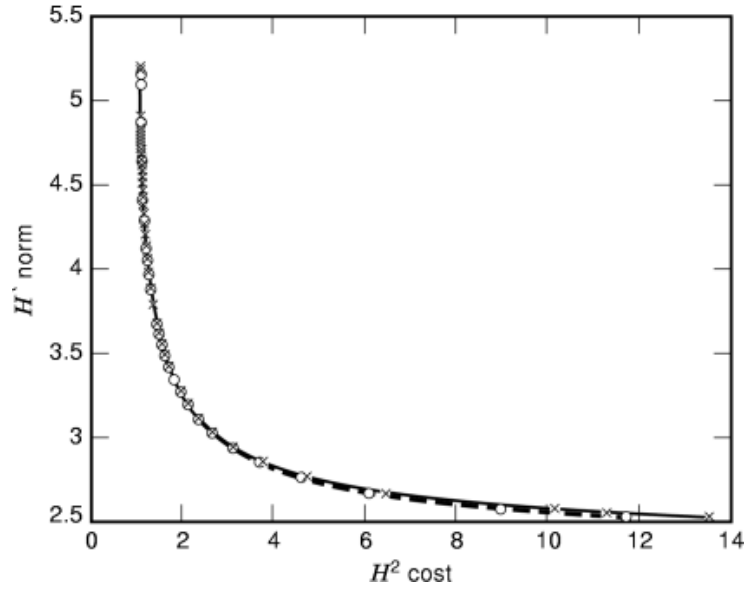


Fig. 2. $\|H(s)\|_\infty$ versus $J$for $n_c = 4, 6$.

For $n_c = 4, 6$, the solutions of the auxiliary minimization problem are obtained for $\gamma \geq 2.55$ using the Ly form homotopy approach. In Fig. 2, $\|H(s)\|_\infty$ is plotted against J for $n_c = 4$ (solid line with "x" indicating the data points) and $n_c = 6$ (dashed line with "o" indicating the data points). For both $n_c = 4$ and $n_c = 6$, the ratio of $\|H(s)\|_\infty$ at $\gamma = 2.55$ to that at $\gamma = \infty$ is 0.49, which indicates that there is about 51% improvement in the

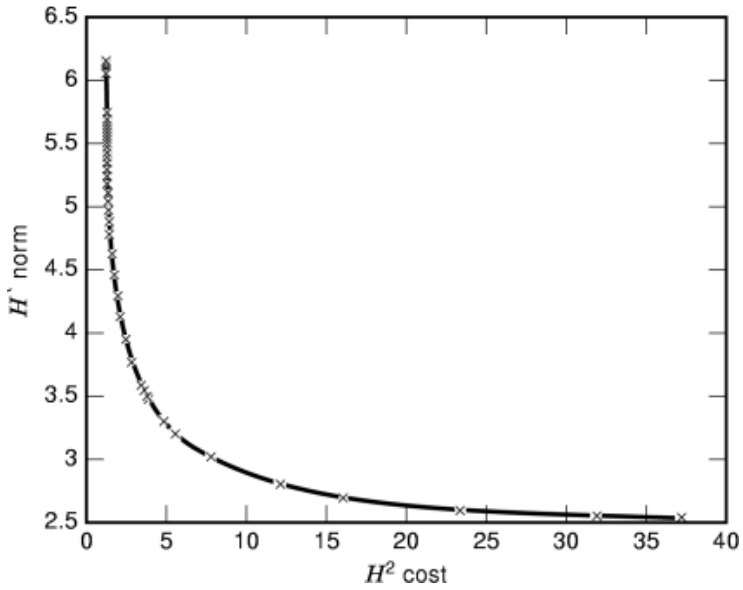


Fig. 3. $\|H(s)\|_\infty$ versus J for $n_c = 2$.

H^∞ performance of the compensator with $\gamma = 2.55$ over the compensator without the H^∞ constraint. For this example, the 4th-order and 6th-order compensators have almost the same H^2 and H^∞ performance.

For $n_c = 2$, two different sets of solutions are obtained by varying β in the initialization step. Different β correspond to different initial (A_c, B_c, C_c), and therefore different homotopy curves. In this case, these two different homotopy curves lead to different solutions which have different minimum H^∞ upper bounds $\gamma_{\min}$.

The trade-off curves are shown in Fig. 3 ($\beta = 100$) and Fig. 4 ($\beta = 1$). The first set of solutions (shown in Fig. 3) is obtained for $\gamma \geq 2.54$, while the second set (shown in Fig. 4) is obtained for $\gamma \geq 9.5$. It can be seen that the first set of solutions has lower H^2 cost and better H^∞ performance. It was verified by sampling in a neighborhood that all the points in both Figs. 3 and 4 are local minima of the auxiliary cost J.

The homotopy algorithms proposed here are examples of probability-one globally convergent homotopy methods; the matrices A_0, B_0, ..., and the starting point θ_0 here play the role of the parameter vector a in the probability-one homotopy theory (13). Figure 5 shows a portion of γ for the previous example, clearly demonstrating the nonmonotonicity in λ and that standard continuation in λ would fail.

As a second example, consider the system given by

$$A = \begin{pmatrix} 0 & 1.0000 \\ -9.8696 & -0.0001 \end{pmatrix},$$

$$C = (10.0637 \quad -9.9363), \quad B = C^T$$

$$E_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad E_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad E_{1\infty} = E_1, \quad E_{2\infty} = E_2$$

$$D_1 = \begin{pmatrix} 0 & 0 \\ 0.0482 & 0 \end{pmatrix}, \quad D_2 = (0 \quad 1)$$

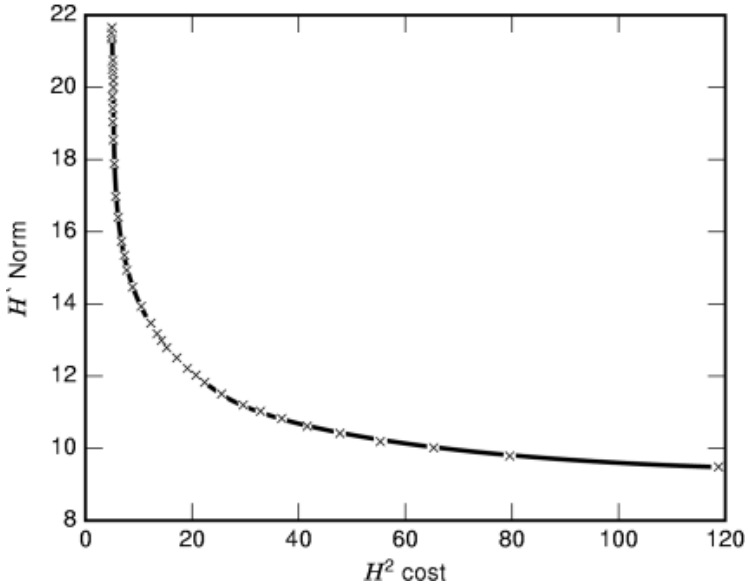


Fig. 4. $\|H(s)\|_\infty$ versus J for $n_c = 2$.

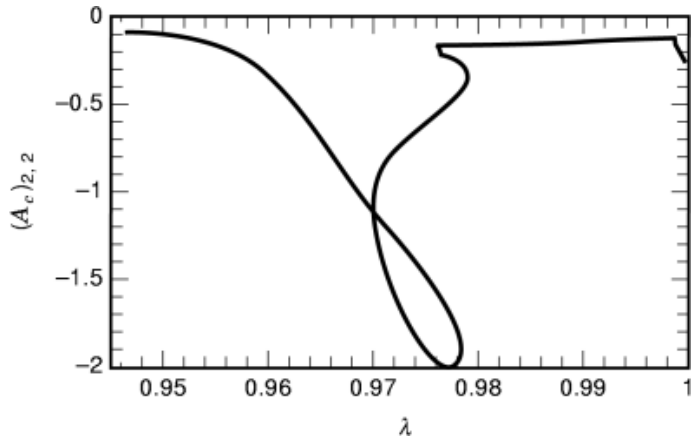


Fig. 5. $(A_c)_{2,2}$ versus λ for $\gamma = 3.8$ and $n_c = 2$.

The trade-off curve is shown in Fig. 6. The solutions of the auxiliary minimization problem are obtained for $\gamma \geq 0.032$. The ratio of $\|H(s)\|_\infty$ at $\gamma = 0.032$ to that at $\gamma = \infty$ is 0.69, which indicates that there is about 31% improvement in the H^∞ performance of the compensator with $\gamma = 0.032$ over the compensator without the H^∞ constraint.

The relative performance of homotopies based on the Ly parametrization, and overparametrization for the combined H^2/H^∞ model order reduction problem was reported in detail in (8,9,10,11). The Ly's formulation is very efficient but can fail to exist or lead to ill conditioning and it is conceivable that it will fail for some problems. This failure of existence in general is related to the insistence on using the minimal number of parameters $n_c m + n_c l$.

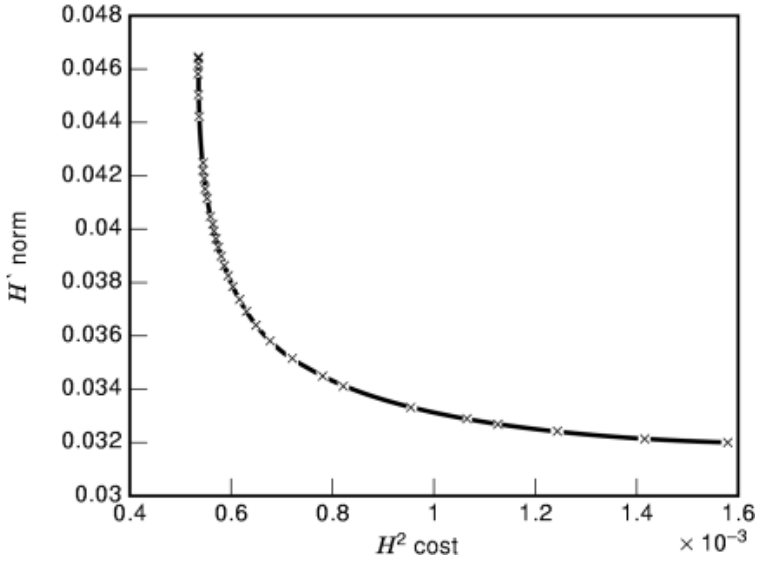


Fig. 6. $\|H(s)\|_\infty$ versus J for the second example.

By using $n_c(n_c + m + l)$ parameters, the over-parametrization formulation solves the ill-conditioning issue related to existence, but introduces a very high order singularity at the solution. It is doubtful whether the Hermite interpolation used here can handle a large problem with a singularity of order 100. A pragmatic suggestion is to try Ly's form first and then the over-parametrization form, switching if ill conditioning or failure occurs. The optimal projection equations formulation of (1) does not make structural assumptions (in fact is completely basis independent), but the optimal projection equations are very difficult and expensive to solve numerically. This cost can be reduced by exploiting tensor product structure and assuming monotonicity in λ of the homotopy zero curves, but Fig. 5 here shows that assumption is not tenable. The over-parameterization formulation makes no structural assumptions and is cheaper computationally than the optimal projection equations, but it is inherently singular at the solution with rank deficiency ${n^2}_c$, which will ultimately overwhelm the numerical linear algebra (8), (11).

The distributed code using PVM was run on a network of seven SGI Indigo^2 workstations. The data came from a control problem for suppressing vibrations in a string under transverse loading from a time varying disturbance force. For dimensions $n = 12, 20, 28, 36$ reduced order controllers of dimensions $n_c = 10, 18, 26, 34$ are sought, respectively.

The speedups versus the number of workstations are shown in Fig. 7 and Fig. 8 ($n = 36, 28, 20, 12$, top to bottom). Figure 7 shows the speedup versus the number of workstations when the master process and the slave processes are run on different machines, while Fig. 8 corresponds to the situation where the master process and a slave process with lower priority are run on one machine and the rest of the slaves are on other machines. For fair comparison all the speedups are computed relative to the results of the best optimized sequential code. In Fig. 7 two workstations correspond to the master process on one machine and the only slave on the other.

As shown in the figures, the speedup increases as the dimension of the problem increases, or as the number of workstations increases for a sufficiently large problem. The speedups from three scenarios (solid line—master and slaves on different machines; dash-dot line—master and a low priority slave on one machine and the rest of the slaves on others; dashed line—master and a slave with the same priority as the master on one machine) for $n = 20$ are plotted against the number of workstations in Fig. 9. If the number of workstations is < 4 it is better to use the second scenario. When the number of workstations is > 4, the speedup is higher if

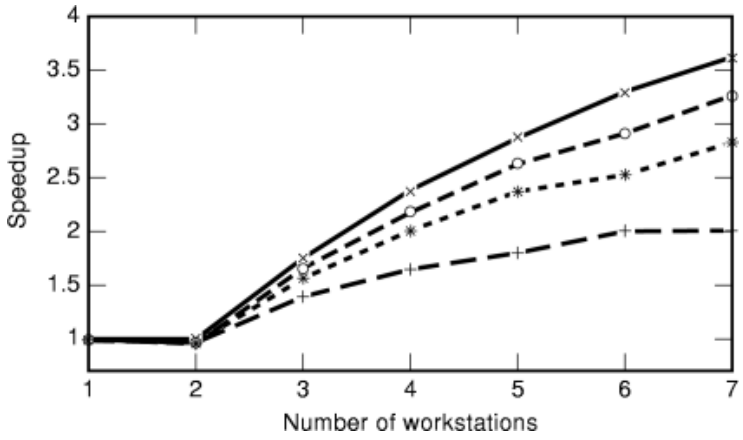


Fig. 7. Speedup with master and slaves on different machines.

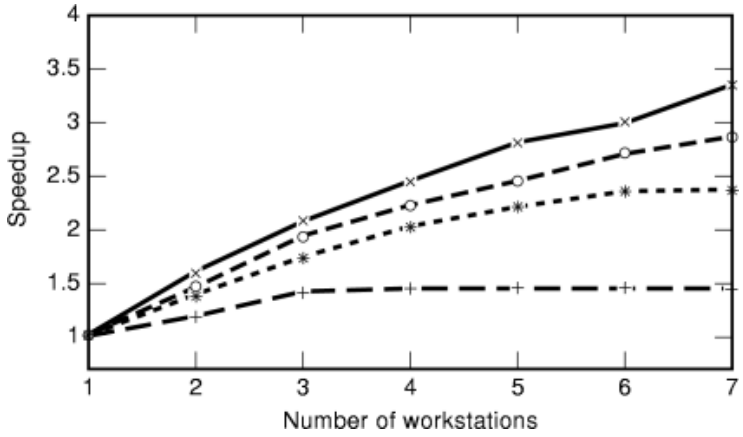


Fig. 8. Speedup when one slave is on the master machine.

all the processes including the master and the slaves are run on different machines. Similar results obtain for large n.

The same algorithm is implemented using the system function calls of an Intel Paragon and run on one with 28 processors at Virginia Polytechnic Institute and State University. Figure 10 shows the results obtained from the run for $n = 12, 20, 28$. The number of nodes varies up to 25. The highest curve corresponds to the speedup when $n = 28$ and the lowest corresponds to that when $n = 12$. As n increases, the advantage of parallel processing also increases. The highest speedup achieved for $n = 28$ using seven SGI Indigo^2 workstations is about 3.3 while the highest speedup using 25 nodes on an Intel Paragon is about 5.1. Comparing speedups is meaningful since the performance of a single SGI Indigo^2 processor is roughly comparable to that of a single i860XP Paragon node (actually, depending on the task, the 100 MHz R4000 Indigo^2 is faster by a factor of 2). However, the cost of the Intel Paragon is a factor of three times the cost of the SGI Unix workstation network. Much higher speedups are potentially possible with the Paragon, but not without considerable programming effort for this controller design problem.

The above methodology can be easily generalized to industrial design environments where software packages like MATLAB or *Mathematica* are often used. The sequential program for mixed-norm H^2/H^∞ LQG

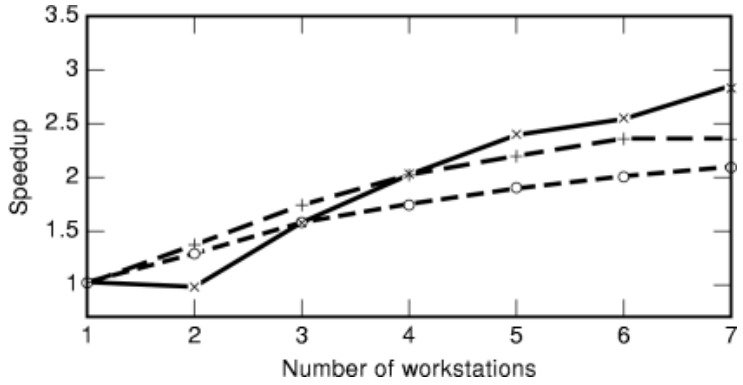


Fig. 9. Comparison of speedups.

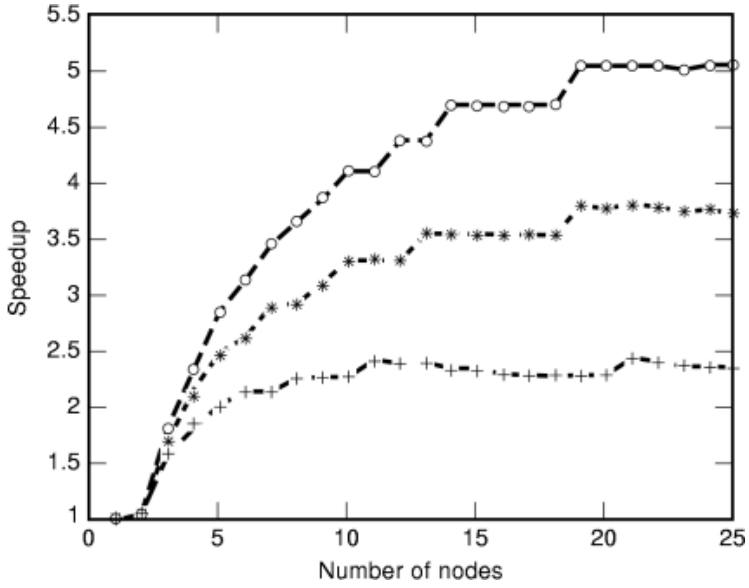


Fig. 10. Results for Intel Paragon XPE-28.

controller synthesis has been developed into a MATLAB package. It is easy to include this distributed implementation into the MATLAB package. Installation requires two steps: the first one is to install PVM on the network of workstations, and the second is to create a file in which all the worker machines on the network are listed (46). The execution of the distributed program from within an interactive design environment, for example, MATLAB, can be done by using a MATLAB function defined in a MATLAB .m file, in which Unix shell commands will start the PVM daemons if they have not already been started, and will then execute the distributed code.

Acknowledgment

The work of Yuzhen Ge and Emmanuel G. Collins, Jr. was supported in part by Air Force Office of Scientific Research grant F49620-95-1-0244 and the work of Layne T. Watson was supported in part by Air Force Office of Scientific Research grant F49620-92-J-0236 and Department of Energy grant DE-FG05-88ER25068/A004.

BIBLIOGRAPHY

1. D. S. Bernstein W. M. Haddad LQG control with an H_∞ performance bound: A Riccati equation approach, *IEEE Trans. Autom. Control.*, **AC-34**: 293–305, 1989.
2. W. M. Haddad D. S. Bernstein Generalized Riccati equations for the full- and reduced-order mixed-norm H_2/H_∞ standard problem, *Syst. Control Lett.*, **14**: 185–197, 1990.
3. K. Glover D. Mustafa Derivation of the maximum entropy H^∞ controller and a state space formula for its entropy, *Int. J. Control*, **50**: 899–916, 1989.
4. D. Mustafa Relations between maximum entropy/H^∞ control and combined H^∞/LQG control, *Syst. Control Lett.*, **12**: 193–203, 1989.
5. K. Zhou *et al.* Mixed H^2 and H^∞ control, *Proc. Amer. Control Conf.*, 1013–1017, 1990.
6. P. P. Khargonekar M. A. Rotea Mixed H^2/H^∞ control: A convex optimization approach, *IEEE Trans. Autom. Control*, **36**: 824–837, 1991.
7. D. Žigić *et al.* Homotopy approaches to the H_2 reduced order model problem, *J. Math. Syst., Estimation, Control*, **3**: 173–205, 1993.
8. Y. Ge Homotopy algorithms for the H^2 and the combined H^2/H^∞ model order reduction problems, *M. S. thesis*, Dept. Computer Sci., Virginia Polytechnic Inst. & State Univ., Blacksburg, VA, 1993.
9. Y. Ge *et al.* Probability-one homotopy algorithms for full and reduced order H^2/H^∞ controller synthesis, *Optimal Control Appl. Methods*, **17**: 187–208, 1996.
10. Y. Ge L. T. Watson E. G. Collins, Jr. A comparison of homotopies for alternative formulations of the L^2 optimal model order reduction problem, *J. Computational Appl. Math.*, **69**: 215–241, 1996.
11. Y. Ge *et al.* Globally convergent homotopy algorithms for the combined H^2/H^∞ model reduction problem, *J. Math. Syst., Estimation, Control*, **7**: 1997.
12. E. G. Collins, Jr. W. M. Haddad S. S. Ying Construction of low authority, nearly non-minimal LQG compensator for reduced-order control design, *preprint*, October, 1993.
13. L. T. Watson R. T. Haftka Modern homotopy methods in optimization, *Comput. Methods Appl. Mech. Eng.*, **74**: 289–305, 1989.
14. L. T. Watson S. C. Billups A. P. Morgan HOMPACK: A suite of codes for globally convergent homotopy algorithms, *ACM Trans. Math. Software*, **13**: 281–310, 1987.
15. M. Jamshidi An overview of the solutions of the algebraic matrix Riccati equation and related problems, *Large Scale Syst.*, **1**: 167–192, 1980.
16. M. A. Shayman Geometry of the algebraic Riccati equation, *SIAM J. Control Optim.*, **21**: 375–409, 1983.
17. I. Gohberg P. Lancaster L. Rodman On Hermitian solutions of the symmetric algebraic Riccati equation, *SIAM J. Control Optim.*, **24**: 1323–1334, 1986.
18. I. Gohberg P. Lancaster L. Rodman *Invariant Subspaces of Matrices with Applications*, New York: Wiley, 545, 1986.
19. V. Kučera A contribution to matrix quadratic equations, *IEEE Trans. Automat. Control*, **17**: 344–347, 1972.
20. P. Lancaster L. Rodman Existence and uniqueness theorems for the algebraic Riccati equation, *Int. J. Control*, **32**: 285–309, 1980.
21. C. Kenney A. Laub E. Jonckheere Positive and negative solutions of dual Riccati equations by matrix sign function iteration, *Syst. Control Lett.*, **13**: 109–116, 1989.
22. B. P. Molinari The time-invariant linear-quadratic optimal control problem, *Automatica*, **31**: 347–357, 1977.
23. J. E. Potter Matrix quadratic solutions, *SIAM J. Appl. Math.*, **14**: 496–501, 1966.
24. D. L. Kleinman On an iterative technique for Riccati equation computations, *IEEE Trans. Autom. Control*, **AC-13**: 114–115, 1968.

25. A. J. Laub A Schur method for solving algebraic Riccati equations, *IEEE Trans. Automat. Control*, **AC-24**: 913–921, 1979.
26. A. J. Laub Numerical linear algebra aspects of control design computations, *IEEE Trans. Automat. Control*, **AC-30**: 97–108, 1985.
27. A. J. Laub Invariant subspace methods for the numerical solutions of Riccati equations, in S. Bittanti, A. J. Laub, and J. C. Willems (eds.), *The Riccati Equations*, New York: Springer-Verlag, 163–195, 1991.
28. P. Van Dooren A generalized eigenvalue approach for solving Riccati equations, *SIAM J. Sci. Stat. Comp.*, **2**: 121–135, 1981.
29. W. F. Arnold A. J. Laub Generalized eigenproblem algorithms and software for algebraic Riccati equations, *Proc. IEEE*, **72**: 1746–1754, 1984.
30. T. Pappas A. J. Laub N. R. Sandell, Jr. On the numerical solution of the discrete-time algebraic Riccati equation, *IEEE Trans. Automat. Control*, **AC-25**: 631–641, 1980.
31. S. S. L. Chang T. K. C. Peng Adaptive guaranteed cost control of systems with uncertain parameters, *IEEE Trans. Automat. Control*, **AC-17**: 474–483, 1972.
32. T. Kailath Some Chandrasekhar-type algorithms for quadratic regulators, *Proc. IEEE Conf. Decis. Control*, New Orleans, LA, 219–223,1972.
33. K. Ito R. K. Powers Chandrasekhar equations for infinite dimensional systems, *SIAM J. Control Optim.*, **25**: 596–611, 1987.
34. R. Byers A Hamiltonian QR algorithm, *SIAM J. Sci. Stat. Comp.*, **7**(1): 212–229, 1986.
35. M. Hazewinkel R. E. Kalman On invariants, canonical forms and moduli for linear, constant, finite dimensional, dynamical systems, *Proc. Math. Syst. Theory*, Udine, Italy, 1975.
36. K. Glover J. C. Willems Parameterization of linear dynamical systems: Canonical forms and identifiability, *IEEE Trans. Autom. Control*, **AC-19**: 640–651, 1974.
37. A. C. Antoulas On canonical forms for linear constant systems, *Int. J. Control*, **33**: 95–122, 1981.
38. G. O. Correa K. Glover Pseudo-canonical forms, identifiable parameterizations and simple parameter estimation for linear multivariable systems: Input-output models, *Automatica*, **20**: 429–442, 1984.
39. P. K. Kabamba Balanced forms: Canonicity and parametrization, *IEEE Trans. Autom. Control*, **30**: 1106–1109, 1985.
40. R. Ober Balanced parametrization of classes of linear systems, *SIAM J. Control Optim.*, **29**: 1251–1287, 1988.
41. L. D. Davis E. G. Collins, Jr. S. A. Hodel A parametrization of minimal plants, *Proc. 1992 Amer. Control Conf.*, Chicago, IL, 355–356, June 1992.
42. U.-L. Ly A. E. Bryson R. H. Cannon Design of low-order compensators using parameter optimization, *Automatica*, **21**: 315–318, 1985.
43. A. Yousuff R. E. Skelton A note on balanced controller reduction, *IEEE Trans. Autom. Control*, **AC-29**: 254–257, 1984.
44. R. H. Bartels G. W. Stewart Solution of matrix equation $AX + XB = C$, *Comm. ACM*, **15**: 820–826, 1972.
45. R. H. Cannon, Jr. D. E. Rosenthal Experiments in control of flexible structures with noncolocated sensors and actuators, *AIAA J. Guid. Control Dynam.*, **7**: 546–553, 1984.
46. A. Geist *et al.* PVM 3 User's Guide and Reference Manual, *Oak Ridge National Laboratory*, ORNL/TM-12187, Oak Ridge, TN, 1993.

YUZHEN GE
Butler University
LAYNE T. WATSON
Virginia Polytechnic Institute and State University
DENNIS S. BERNSTEIN
University of Michigan
EMMANUEL G. COLLINS, JR.
Florida A&M/Florida State University

/ /

Horn Clauses

Standard Article

John M. Jeffrey[1], Jorge Lobo[1], Tadao Murata[2]

[1]Elmhurst College, Elmhurst, IL

[2]University of Illinois at Chicago, Chicago, IL

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2427

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (211K)

Abstract

The sections in this article are

Horn Clause Syntax and Terminology

Informal Horn Clause Semantics

A More Formal Semantics

Summary

| | |

Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

HORN CLAUSES

This article introduces Horn clause logic, a subset, yet expressive, form of first-order logic. The syntax of Horn clauses along with the semantics and the related inference rule are presented. Background on how Horn clause logic is related to general first-order logic is covered. The subject matter is informally introduced through an example, which is then followed by a more rigorous description with additional examples. Horn clauses, named after Alfred Horn, constitute a language subset of first-order clauses, which represents a canonical, normal form for expressing first-order predicate calculus formulas. Although Horn clauses are syntactically restricted clauses, they provide sufficient expressiveness and semantics that are easier to implement on several computing platforms and are central to the study of logic programming languages and knowledge-based systems. In the early 1970s, through research and experiments with various formulations of problem-solving tasks, Horn clauses were found to be very useful and expressive (1). Lloyd (2) shows that all computable functions can be expressed in terms of a formal deduction system, the inference rule of which is SLD resolution and the language syntax of which consists of definite program clauses, a subset of Horn clauses. *SLD* resolution stands for a restricted resolution inference rule with a Selection function restricted to Linear resolution and Definite clauses. The history of the naming and definition of this inference rule is discussed in detail in this article.

Horn Clause Syntax and Terminology

Before proceeding with an introductory Horn clause example, some essential definitions are informally given. A clause is often written in the form

$$\forall x_1 \cdots \forall x_z A_1 \vee A_2 \vee \cdots \vee A_m \leftarrow B_1 \wedge B_2 \wedge \cdots \wedge B_n$$

where $m \geq 0$, $n \geq 0$, $\vee$ is logical-OR, $\wedge$ is logical-AND, $\leftarrow$ is logical-IMPLIES, the A_i's and B_j's are atoms, $x_1, \ldots, x_z$ are all the variables that occur in A_i's and B_j's, and $\forall$ represents universal quantification (for all). An *atom* has the form of $P(t_1, t_2, \ldots, t_n)$, where P is a predicate symbol and t_i's are terms. A *term* is a constant, variable, or a function f applied to terms, t_i, notated as $f(t_1, t_2, \ldots, t_n)$. Terms with no variables are called *ground terms*. Similarly, atoms with only ground terms are called *ground atoms*.

The above clause is read as: "A_1 or A_2 or $\ldots$ or A_m is/are true if B_1 and B_2 and $\ldots$ B_n is/are true." Note that a clause may be empty, that is, $m = 0$ and $n = 0$; the truth assignment for an empty clause is always false and is considered to represent logic contradiction. The empty clause is denoted as $\square$.

When formally describing semantics for clauses it is convenient to write them using alternative syntax, but logically equivalent forms. One form is

$$\forall x_1 \cdots \forall x_z A_1 \vee A_2 \vee \cdots \vee A_m \vee \neg B_1 \vee \neg B_2 \vee \cdots \vee \neg B_n$$

The ¬ represents logical-NOT. Both negated atoms and nonnegated atoms are called negative and positive literals, respectively. The notation $|L|$ represents the atom in a literal with the possible negation symbol removed.

For the purpose of defining resolution and other related concepts, it is convenient to represent a clause as a set of literals with universal quantification of variables assumed:

$$\{A_1, A_2, \ldots, A_m, \neg B_1, \neg B_2, \ldots, \neg B_n\}$$

A *Horn clause* has at most one atom to the left of the implication arrow or, equivalently, at most one positive literal; that is, $0 \leq m \leq 1$. If $m = 1$, a Horn clause has the form

$$A \leftarrow B_1 \wedge B_2 \wedge \cdots \wedge B_n$$

The above is read as: "A is true if B_1 and B_2 and ... and B_n is/are true." This type of Horn clause is called a *definite program clause* in logic programming contexts. A is called the *head* of the clause and the B_i's collectively are called the *body* of the clause. If $n = 0$, the clause represents a statement that atom A is unconditionally true, that is, a *fact.* A fact is also called a *unit clause,* a special kind of definite program clause. Unit clauses are often written with the implication arrow omitted. A *Horn program* is a collection of definite program clauses. If $m = 0$ and $n \geq 1$, the clause is called a *definite goal clause,* a *query,* or, simply a *goal clause* in logic programming contexts.

In the following examples, strings composed of lowercase letters and Arabic digits represent constants and upper case letters represent variables. Also, the logical-AND operator ($\wedge$) is replaced by a comma (,). Truth values are represented as TRUE and FALSE. (Note that when clauses are represented in set notation, the comma delimits the literals and no longer represents logical-AND.)

Relationship Horn Program Example. The following example illustrates the use of Horn clauses for expressing the relationships of father to child. The atom *Father*(F, C) represents the statement that F is a father of child C and *Parent*(F, C) represents the statement "F is a parent of C."

Father(F, C) ← *Parent*(F, C), *Male*(F)
Parent(*joe, dave*)
Parent(*joe, john*)
Parent(*mary, john*)
Male(*joe*)
Male(*john*)
Male(*dave*)
Male(*mark*)
Female(*mary*)

Informal Horn Clause Semantics

Informal Model-Theoretic Semantics of Relationship Example. The first clause is read as: "F is a father of C if F is a parent of C and F is male." The remaining clauses are facts, each assumed to be true assertions. For example, *Parent*(*joe, dave*) asserts that "joe is a parent of dave." Informally, the *Parent* predicate symbol is intended to represent a parent relation over a set of people, called the universe of discourse. The universe of discourse might represent the set of people in a particular country, state, organization, etc.

The assignment of terms to elements in the universe of discourse and the assignment of relations to predicate symbols is an interpretation. The set of names {*joe, dave, john, mark, mary*} is intended to represent particular people in a universe of discourse; the representation is an interpretation. The binary relation {(*joe,*

dave), (*joe, john*), (*mary, john*)} is assigned to predicate symbol *Parent.* The set {*joe, john, dave, mark*} is a unary relation assigned to predicate symbol *Male.* The set {*mary*} is assigned to predicate symbol *Female.* These predicate symbol assignments complete the interpretation.

By substituting the variables with all possible name values chosen from a universe of names, one is able to "compute" the relation *Father* by noting which names satisfy or make the rule TRUE according to the usual semantics associated with logical-AND and logical-IMPLIES semantics. For example, given the substitutions that replace F with *joe* and C with *john,* the rule is TRUE; so the tuple (*joe, john*) is in the relation assigned to *Father.* However, when any name other than *joe* is assigned to variable F, the rule is not necessarily TRUE since either the name is not a left-hand member of the *Parent* relation or is not in the *Male* unary relation, as assigned in the interpretation. Choosing the universe of names, assigning the relations to all predicate symbols (even those occurring only in the head of a Horn clause), and evaluating the truth values of each clause is one approach to giving a semantics to sets of clauses and Horn clauses in general. Each universe and assignment comprises an interpretation. The interpretations that satisfy or make all clauses true for all substitutions of variables are models.

Informal Procedural Semantics of Relationship Example. Procedural semantics are based on a formal deduction system, which comprises a language (Horn clauses in this case), a set of proper axioms (the set of clauses), and an inference rule. The procedural semantics describe how to construct proofs. Constructing a proof involves applying an inference rule to a set of clausal axioms to deduce more clauses, which, in turn, are added to the set of axioms. A sequence of inferences starting with a set of axioms is a proof of the last clause in the sequence.

Logicians are interested in inference rules that are sound and complete. An inference rule is characterized as being *sound* if and only if it yields only clauses that are satisfied by the model-theoretic semantics, that is, they are TRUE in every model of the theory. An inference rule is *complete* if and only if a proof can be constructed for any clause satisfied by the model-theoretic semantics.

Resolution is a sound and complete inference rule for general clauses. Readers interested in further details should consult the original work on resolution by Robinson (3) or textbooks such as those by Chang and Lee (4) or Loveland (5). Resolution and related concepts are briefly introduced later in this article for the purpose of providing a context for SLD resolution, which is a more restricted resolution inference rule for Horn clauses.

Informally, resolution involves two general steps. The first step involves unification of two atoms, each belonging to two distinct clauses. One of the atoms appears negated in a clause and the other occurs as a positive literal in the other clause. Unification is a substitution or an assignment of variables to (not necessarily ground) terms that make the atoms syntactically match. A substitution of a variable X with a term t is often denoted X/t. A unifier is denoted as a set of these variable assignments.

The second step involves applying the substitution for all variables in each clause and canceling these matching atoms and leaving the remaining atoms. The remaining atoms form a new clause, called the *resolvent,* that is added to the set of clauses that potentially could be used in a resolution step.

A proof in a resolution-based deduction system is one based on reaching a contradiction—a proof by contradiction. In a system composed only of Horn clauses, one begins with the goal or query clause and, through resolution at each step, attempts to arrive at a contradiction, which is represented by the empty clause, $\square$. The sequence of steps that begins with a goal clause and that arrives at a contradiction is called a *refutation.* Because there may be several choices in clauses to use in a resolution step, the construction of a refutation is tantamount to a search problem.

The preceding description to finding a refutation is called a top-down procedure. In contrast there is a bottom-up approach that starts with the unit (fact) clauses and works "up" to the goal clause. For this example, we utilize the top-down procedure to illustrate a refutation. We give a more formal description later. For now, we give an overview.

4 HORN CLAUSES

Suppose that we attempt to prove that "joe is a father of dave." Toward a contradiction, one negates the ground atom *Father*(*joe, dave*) and expresses it as a goal clause:

$$\neg Father(joe, dave)$$

This unifies with the head of the Father rule with unifier {*F*/*joe*, *C*/*dave*}. Applying this unifier to the other literals in the Father rule, we are left with the resolvent:

$$\neg Parent(joe, dave) \vee \neg Male(joe)$$

This clause is now added to the collection of clauses. In general resolution-based deduction systems, there are many choices of clauses to use in each resolution step and, moreover, there may be more than one choice in the literals that unifies. The choices implemented in such a reasoning system define the operational or procedural semantics. In Horn clause–based deduction systems, the computational model or operational semantics requires that one of the clauses to be used in each resolution step be the most recently added goal clause. This strategy is one component of SLD resolution, a restricted resolution rule used for Horn clause–based systems. Using this strategy, we continue with the proof example.

Although we require that one of the clauses be a goal clause, we need to select a literal in a goal clause. Defining a rule or function that selects a literal in a clause is the other component of SLD resolution.

In this example, we choose the leftmost literal ¬*Parent*(*joe, dave*) instead of ¬*Male*(*joe*). The choice is arbitrary here. This unifies with the unit clause *Parent*(*joe, dave*) with the empty unifier, which requires no variable substitutions. We are left with the (goal clause) resolvent:

$$\neg Male(joe)$$

Finally, ¬*Male*(*joe*) unifies with the unit clause *Male*(*joe*) with the empty unifier. The resolvent is the empty clause, which represents contradiction. With this, we have shown a proof that *Father*(*joe, dave*) is TRUE. That is, we negated what we were trying to prove and, through a sequence of resolution steps, arrived at a contradiction. This sequence is a refutation.

Note that since one and only one of the clauses used in SLD resolution must be a goal clause, and a goal clause contains only negative literals, the negative literal used in a unification step must unify with the head of a definite clause. We do not need to look at the clause bodies when we apply the unification step. In general, there may be more than one clause head to unify with. Choosing which goal clause literal and then which definite clause head is specified in the computational model used to construct refutations. If at any step there is no match between the selected literal and any definite clause head, backtracking to the previous goal clause and a different program clause is chosen at that step. This process continues until a contradiction is reached or backtracking goes back to the original goal clause and we do not have more choices. In that case the operational semantics are defined to give an answer: no. Notice that there is no need to backtrack over the goal literal selected, since in order to obtain the empty clause all literals in the goal must be resolved away. If a refutation exists we will find one independently of the literal selected.

When variables are present in the original goal clause, the unifiers used at each resolution step are composed and the substitutions for the original goal clause are called the *computed answer substitutions*. Because there may be more than one clause to resolve with the goal clause at each step of a refutation, there is potentially more than one computed answer. How this is efficiently implemented is another issue in Horn clause–based logic programming systems, which is beyond the scope of this article.

For example, if the goal clause ¬*Father*(*X*, *dave*) is given, the computed answer would be {*X*/*joe*}. This would similarly follow the steps given in the previously shown refutation. If the goal clause ¬*Father*(*X*, *Y*) were given, two computed answers would be given:

(1) $\{X/joe, Y/dave\}$
(2) $\{X/joe, Y/john\}$.

In this second query, two refutations are constructed via SLD resolution and backtracking. The procedure of computing all possible answers, starting with the query, terminates when no more refutations can be constructed with SLD resolution.

A More Formal Semantics

The study of logic, including syntactic and semantic aspects, can be broadly viewed as one approach to reasoning about abstract and concrete entities and the relationships between these entities; that is, it is one approach to a knowledge representation scheme or paradigm.

One may classify a logic according to various attributes that reflect some assumptions made regarding the relationships between these entities within a universe of discourse that the logic represents. For example, one taxonomy of logic may be based upon the nature of the range of the variables, if any, of the logic language. Within a zero-order (propositional) logic language, there are no variables; the propositional symbols represent statements (propositions) about particular entities. Higher-order logics introduce variables that range not only over these universe entities, but also the relationships between the entities (second-order) the relationships among relationships between entities (third-order), etc. For each of the orders, each class of variables may be quantified. Horn clauses are a syntactically restricted form of first-order logic, the language of which includes variables that range over the fundamental entities or elements of the universe of discourse.

There is a multitude of other attributes that can be used to classify a logic. We state just a few here to indicate our assumptions. For example, one may utilize different underlying (possibly partially ordered) sets of truth values. Related to this is the classification of a logic based upon the assumption made about the "law of the excluded middle," which informally means that all statements in a logic are either true or false. Another classification can be based upon whether a logic is monotonic or not. Within a nonmonotonic logic the assignment of a truth value of one statement may alter the truth value of another; within a monotonic logic truth values are not altered. Only two-valued, monotonic logic and the law of the excluded middle are utilized within Horn clause logic.

There are two primary approaches to assigning semantics or meaning to sets of clauses and Horn clauses. These are model-theoretic and procedural semantics. Each of these are considered using the relationship program example as a means for discussion and presentation.

Model-Theoretic Semantics. The model-theoretic semantics of first-order logic defines an assignment of the truth values of TRUE or FALSE to every formula in the logic. This assignment is done in several steps. First, every constant in the logic is identified with an element in a particular domain, for example, the rational numbers. Then every function symbol in the logic is identified with a function over the same domain. Then the interpretation is extended to the predicate symbols by assigning to each predicate a relation, again using the same domain. Finally, based on these assignments, the truth value of formulas is defined inductively by first assigning truth values to atomic formulas and then to more complex formulas. In the case of Horn clauses this process can be highly simplified. Interpretations are restricted to a very special class called *Herbrand interpretations.* A Herbrand interpretation is simply a set of ground atomic formulas, that is, atomic formulas with no variables. The meaning that these interpretations assign to formulas is as follows. The truth value of any ground atomic formula in the set is TRUE, and for any other ground formula its truth value assignment if FALSE. Before we say how clauses are interpreted let us see an example.

From our family relationship program, a Herbrand interpretation could be the set $\{Father(mary, john), Parent(joe, david), Male(david), Father(mark, mary)\}$. Note that this interpretation is not a good interpretation

since the program does not seem to imply that *mary* is the father of *john*. Good interpretations are called Herbrand models, or simple models, and are defined as follows.

Let $Ground(\Phi)$ be the ground instances of all the clauses in a program Φ. That is, take any clause C in Φ and replace all the variables in C with ground terms; the resulting clause must be in $Ground(\Phi)$. A Herbrand interpretation M is said to be the intended model of the program φ if (1) for every clause in $Ground(\Phi)$ whose atoms in the body are in M, its head is also in M and (2) there is no proper subset of M with property (1).

Continuing with our family example, the set {*Parent*(*joe, dave*), *Parent*(*joe, john*), *Parent*(*mary, john*), *Male*(*joe*), *Male*(*john*), *Male*(*dave*), *Male*(*mark*), *Female*(*mary*), *Father*(*joe*, *dave*), *Father*(*joe*, *john*)} is the intended model of our program, and coincides with our intuition about the meaning of the program.

It can be shown that for any Horn program this model always exists and is unique. Let us look at another example.

$arc(a, b)$
$arc(b, c)$
$arc(b, d)$
$path(X, Y) \leftarrow arc(X, Y)$
$path(X, Y) \leftarrow path(X, Z), path(Z, Y)$

The program is intended to represent the arcs in a graph and paths between nodes in the graph. The meaning of the first three clauses is obvious. The last two clauses are more interesting. The fourth clause says that there is a path from X to Y if there is an arc from X to Y. The last rule says that there can also be a path from X to Y if there is a path from X to an intermediate point Z and then a path from Z to Y. With a careful analysis the reader can notice that the intended model for this program is

$$\{arc(a, b), arc(b, c), arc(b, d), path(a, b), path(a, c), path(a, d), path(b, c), path(b, d)\}$$

With the definition of the intended model of a Horn program we can define the *answers* to a goal. Recall that a goal clause is a clause of the form $\not B_1 \vee \cdots \vee \not B_n$, also written as $\leftarrow B_1, \ldots, B_n$. Answers are defined using substitutions. Formally, a substitution θ is an assignment of terms to variable names. A substitution is denoted as follows: $\theta = \{X_1/t_1, \ldots, X_n/t_n\}$. The notation $E\theta$ indicates the application of a substitution θ to E, where E represents a term, atom, literal, clause, or set of clauses. $E\theta$ represents E with all variables replaced with the terms. For example, if $E = \{p(X,Y), q(X,Z)\}$ and θ $\{X/a, Y/Z\}$, then $E\theta = \{p(a, Z), q(a, Z)\}$. An answer for a goal clause G in a Horn program Φ is a substitution θ such that any ground instance of an atom in $G\theta$ is a member of the intended model of Φ. From our last example the goal $\leftarrow path(a,X)$ has three answers, $\theta_1 = \{X/b\}$, $\theta_2 = \{X/c\}$, and $\theta_3 = \{X/d\}$.

There are many properties connected Herbrand models and answers to first-order logic. Details can be found in Refs. 2 and 6.

Procedural Semantics. The following subsections present a general procedure to compute answers. First, unification and general resolution is presented to provide the context for a restricted resolution-based procedure, called the SLD resolution, which is central to the procedural semantics of Horn clauses.

Unification. A key component in the procedure to compute answers is called unification. Unification is the process of finding a substitution that makes two terms or atoms syntactically identical by applying the substitution to the terms or atoms.

There may be more than one unifier. There are some unifiers that have less term-to-variable assignments than others; the unifiers with less are referred to as the most general. Since any of them could be used as a general unifier, one is chosen as a representative and called the most general unifier (*mgu*).

Table 1. Unification Examples

	Expression 1 (E1)	Expression 2 (E2)	(an) mgu(E1, E2)
(1)	$f(X, a)$	$f(g(Y), Z)$	$\{X/g(Y), Z/a\}$
(2)	$f(X, a)$	$f(g(Y), b)$	fail
(3)	$p(f(X), X, g(b))$	$p(f(Y), Z, g(X))$	$\{X/b, Y/b, Z/b\}$
(4)	$p(f(X), X, g(b))$	$p(f(Y), Z, g(X_1))$	$\{X/Y, Z/Y, X_1/b\}$
(5)	$p(X, X)$	$p(Y, f(Z))$	$\{X/f(Z), Y/f(Z)\}$
(6)	$p(X, X)$	$p(Y, f(Y))$	fail

Given that E_1 and E_2 are both terms or atoms, hereafter called expressions, the notation mgu(E_1, E_2) represents an mgu of expressions E_1 and E_2. Shown in Table 1 are examples of applying the mgu function to expressions.

Properties about unification and an algorithm to compute most general unifiers can be found in Ref. 7.

Binary Resolution. The second component of the procedure to compute answers is *binary resolution.* The general resolution introduced by Robinson (8) is an inference rule that deals with general clauses and has two steps: binary resolution and factoring. Factoring is presented to be complete in the discussion; however, it is not necessary when using SLD resolution and Horn clauses.

Before describing binary resolution, some convenient terminology is introduced. Two clauses C_1 and C_2 are *renamed apart* when variables in the clauses are renamed so that the set of variables that appear in C_1 is disjoint from those appearing in C_2. If at least one variable is renamed in a clause C and the resultant clause is C', then C' is also called a *variant* of C.

Binary resolution can be defined as follows. Let C_1 and C_2 be two clauses renamed apart. A clause C is a *resolvent* of C_1 and C_2 if and only if

(1) there exists a literal $L_1 \in C_1$ and a literal $\not L_2 \in C_2$ such that $|L_1|$ and $|L_2|$ are unifiable with mgu($|L_1|$, $|L_2|$) $= \theta$, and
(2) $C = C_1\theta - \{L_1\theta\} \cup C_2\theta - \{L_2\theta\}$.

We say C_1 and C_2 are *parent* clauses of resolvent C. We also say that L_1 and L_2 are the *literals resolved upon.* An example of resolvents is as follows. Let

$$C_1 = \{p(X_1), \neg q(f(Y_1))\} \quad \text{and} \quad C_2 = \{\neg p(a), q(X_2), r(g(X_2))\}$$

C' and C'' are two possible resolvents of parents C_1 and C_2:

(1) Resolving upon literals $p(X_1)$ and $\not p(a)$ from clauses C_1 and C_2, respectively, we have

$$mgu(p(X_1), p(a)) = \{X_1/a\} \quad \text{and}$$
$$C' = \{\neg q(f(Y_1)), q(X_2), r(g(X_2))\}$$

(2) Resolving upon literals $q(f(Y_1))$ and $q(X_2)$ from clause C_1 and C_2, respectively, we have

$$mgu(q(f(Y_1)), q(X_2)) = \{X_2/f(Y_1))\} \quad \text{and}$$
$$C'' = \{p(X_1), p(a), r(g(f(Y_1)))\}$$

A refutation system that has only the binary resolution inference rule is not refutation complete. We consider the following classic example (see Ref. 8). Let

$$C_1 = \{p(X_1), p(Y_1)\} \quad \text{and} \quad C_2 = \{\neg p(X_2), \neg p(Y_2)\}$$

Clearly, the set $\Gamma = \{C_1, C_2\}$ is unsatisfiable; that is, it has no models. However, one needs more than just the binary resolution inference rule to refute Γ. All the resolvents of C_1 and C_2 are variants of the clause $\{p(X), p(Y)\}$.

Let $C = \{L_1, \ldots, L_n\}$ be a clause for which there exists two (same sign) literals L_i and L_j $(i = j)$ that are unifiable with a unifier θ. A *factor* of C is the clause $C\theta$.

In the following, the most general unifier will be used in factoring. We give three examples of factoring:

(1) Factoring C_1 and C_2 as given above, we get, respectively,

$$\{p(X)\} \quad \text{and} \quad \{\neg p(X)\}$$

Clearly, the resolvent of these two clauses is $\square$. Thus, through factoring and then the resolving of those factors, we have a refutation for $\Gamma = \{C_1, C_2\}$.

(2) $C'' = \{p(X_1), p(a), r(g(f(Y_1)))\}$ as shown above in the resolvent example can be factored as

$$\{p(a), r(g(f(Y_1)))\}$$

$C' = \{q(f(Y_1)), q(X_2), r(g(X_2))\}$ as shown above in the resolvent example cannot be factored.

(3) Suppose clause $C = \{p(X), q(X), q(f(Y))\}$. A factor of C is $\{p(f(Y)), q(f(Y))\}$.

Let Γ be a set of clauses. A *resolution deduction* (or *resolution derivation*) of clause C from Γ is a deduction $C_1, C_2, \ldots, C_n$ such that C is exactly C_n and each C_i $(1 \leq i \leq n)$ is a clause for which

(1) $C_i \in \Gamma$,

(2) C_i is a factor of a clause C_j $(j < i)$, or

(3) C_i is a resolvent of preceding clauses $C_1, C_2, \ldots, C_{i-1}$.

The resolution derivation is denoted: $\Gamma \vdash C$.

A *resolution refutation* of a set of clauses Γ is a resolution deduction of $\square$ from Γ and is denoted $\Gamma \vdash \square$.

Resolution with Horn Clauses. Since searching for refutations in a general resolution system can be quite expensive in time and space, naturally one investigates useful subclasses of clauses and restricted resolution inference rules. Kowalski and Kuehner (9) indicate that an inference system should satisfy the following criteria:

(1) "It should admit few redundant derivations and limit those which are irrelevant to the proof."

(2) "It should admit simple proofs."

(3) "It should determine a search space which is amenable to a variety of methods of heuristic searches."

It has been shown that SLD resolution meets the above criteria. The refutation procedure based upon SLD resolution was first described by Kowalski (10). As stated by Lloyd (2) along with Apt and van Emden

(11), the "SLD" in SLD resolution is an abbreviation for "linear resolution with a selection function for definite clauses." We note that there are other assumptions made about what "SLD" represents. In Ref. 12 Ringwood gives a brief history of the confusion about the SLD abbreviation; we leave it to the interested reader to consult that article for details.

SLD resolution is a further restriction to other restricted forms of resolution, namely linear-input resolution. Linear and input resolutions are briefly described as follows; a more detailed description can be found in Ref. 4.

- *Linear resolution* requires that both parents of a resolvent must be a previous resolvent; the first resolvent's parents, of course, must be the input clauses (which are the proper axioms).
- *Input resolution* requires that one of the two parents be an input clause. (The input clauses can be thought of as proper axioms.)
- *Linear-input resolution* naturally combines the previous two restrictions. It can be shown that linear-input resolution is only refutation complete for Horn clauses (12).

To describe SLD resolution, several definitions are presented first.

Let G be a goal clause $\leftarrow A_1, \ldots, A_i, \ldots, A_m$. Let C be a definite clause, $A \leftarrow B_1, \ldots, B_n$. A goal clause G', also called a resolvent, is derived from G and C using mgu θ if the following constraints hold:

(1) A_i is an atom from G called the selected atom.
(2) θ is the mgu of A_i and A, the head atom of clause C.
(3) G'' is the resolvent goal clause $(\leftarrow A_1, \ldots, A_{i-1}, B_1, \ldots, B_n, A_{i+1}, \ldots A_m)\theta$.

An SLD derivation is a restricted resolution derivation as follows. Let Φ be a Horn program and G_0 a goal clause. An SLD derivation of $\Phi \cup \{G_0\}$ is a (not necessarily finite) sequence of

(1) Goals $G_0, G_1, \ldots$
(2) Definite clauses from Φ, $C_1, C_2, \ldots$
(3) mgus $\theta_1, \theta_2, \ldots$

such that G_{i+1} is derived from G_i and a (renamed apart from G_i) clause C_{i+1} using θ_{i+1}.

It should be noted that the variables in each C_{i+1} may need to be renamed so that there are no common names with the corresponding goal G_i. Such a renamed clause is also referred to as a variant of a clause. Each clause variant C_i is called an input clause of the derivation.

Let Φ be a definite program and G_0 a goal clause. A finite SLD derivation of $\Phi \cup \{G_0\}$, whose goal sequence is $G_0, G_1, \ldots, G_n = \square$, is called an SLD refutation of $\Phi \cup \{G_0\}$ of length n.

An SLD derivation, like all general resolution-based deductions, can be *finite* or *infinite.* For finite SLD derivations, they are partitioned into *successful* or *failed* ones. A *successful* SLD derivation is simply another name for an SLD refutation.

A *failed* SLD derivation is one in which the last clause in the derivation contains a selected atom that does not unify with any program clause head. It is also shown that SLD resolution is sound and complete with respect to Horn clause logic (2). The *success set* of a definite program Φ is the set of all ground atoms A such that $\Phi \cup \{\leftarrow A\}$ has an SLD refutation. It can be shown that the success set of a program Φ is exactly the intended Herbrand model of Φ.

Let Φ be a Horn program and G a goal clause. A *computed answer substitution* θ for $\Phi \cup \{G\}$ is the resultant substitution of restricting the composition of $\theta_1 \ldots \theta_n$ to the variables in the goal G, where $\theta_1 \ldots \theta_n$ is the sequence of mgu's used in an SLD refutation of $\Phi \cup \{G\}$.

Table 2. SLD Refutation with Ground Goal Clause Example

	Goal Clause	Definite Clause	mgu
G_0	← Father(joe, dave)		
	⇓	Father(F, C) ← Parent(F, C), Male(F)	θ_1 = {F/joe, C/dave}
G_1	← Parent(joe, dave), Male(joe)		
	⇓	Parent(joe, dave)	θ_2 = { }
G_2	← Male(joe)		
	⇓	Male(joe)	θ_3 = { }
G_3	□		

Examples of Refutations with the Relationship Horn Program. Let Φ be set of definite clauses that represent a family of people as given earlier.

Father(F, C) ← *Parent*(F, C), *Male*(F)
Parent(*joe, dave*)
Parent(*joe, john*)
Parent(*mary, john*)
Male(*joe*)
Male(*john*)
Male(*dave*)
Male(*mark*)
Female(*mary*)

We show an SLD refutation and illustrate how a computed answer is constructed. Suppose we wish to prove that "joe is a father of dave." One negates the ground atom *Father*(*joe, dave*) and expresses it as a goal clause:

$$G_0 = \neg Father(joe, dave)$$

A refutation for $\Phi \cup \{G_0\}$ is shown in Table 2. By constructing an SLD refutation, we have shown that "joe is a father of dave."

When implementing SLD resolution, one must commit to a selection function. This requires a well-defined procedure for selecting one and only one of the literals in the goal clause at each step in an SLD derivation (or refutation). In the above example, at step 1, the leftmost literal, ~~*Parent*~~(*joe, dave*) was chosen; however, ~~*Male*~~(*joe*) could have been chosen. It can be shown that if a refutation exists, it may be found, independent of the literal selected.

Another aspect of implementing SLD resolution involves the order of potential clause heads tried when attempting to unify the selected literal. This is where backtracking is employed. The reader should consult Ref. 2 for details and a starting point for investigating other sources that present machine implementation details.

Suppose one wants to compute the answers to "Who is the father of whom?" We illustrate this using the same set of definite clauses that represent a family of people. An example of computing one of the answers is shown in Table 3. Note that the definite clauses from Φ have to be renamed apart from the goal at some steps. The computed answer is {*F*/*joe*, *C*/*dave*}. That is, "joe is a father of dave" as shown earlier. This answer is constructed by composing the mgu's used at each step as follows:

$$\{F/joe, C/dave\} = \theta_1\theta_2\theta_3 = \{F/F_1, C/C_1\}\{F_1/joe, C_1/dave\}\{\}$$

Table 3. SLD Refutation and Computed Answer Example

	Goal Clause	Definite Clause	mgu
G_0	$\leftarrow$ Father(F, C)		
	$\Downarrow$	Father(F_1, C_1) $\leftarrow$ Parent(F_1, C_1), Male(F_1)	θ_1 = {F/F_1, C/C_1}
G_1	$\leftarrow$ Parent(F_1, C_1), Male(F_1)		
	$\Downarrow$	Parent(joe, dave)	θ_2 = {F_1/joe, C_1/dave}
G_2	$\leftarrow$ Male(joe)		
	$\Downarrow$	Male(joe)	θ_3 = { }
G_3	$\square$		

Table 4. Alternate SLD Refutation to Table 3 Example

	Goal Clause	Definite Clause	mgu
G_0	$\leftarrow$ Father(F, C)		
	$\Downarrow$	Father(F_1, C_1) $\leftarrow$ Parent(F_1, C_1), Male(F_1)	θ_1 = {F/F_1, C/C_1}
G_1	$\leftarrow$ Parent(F_1, C_1), Male(F_1)		
	$\Downarrow$	Parent(joe, john)	θ_2 = {F_1/joe, C_1/john}
G_2	$\leftarrow$ Male(joe)		
	$\Downarrow$	Male(joe)	θ_3 = { }
G_3	$\square$		

Yet another SLD refutation for the same program and goal clause is given in Table 4. This computed answer shows that "joe is a father of john." At step 1 the leftmost literal is chosen again, as given previously, but a different clause is used. If a different selection function were used, one may chose the rightmost literal at step 1. Careful analysis reveals that the same answers would be computed as the two preceding examples illustrate.

Data Structures in Horn Programs. In this section we will show how function symbols are used in Horn programs to build data structures. The following examples all implement different operations over lists of terms. A list of terms is a finite sequence of zero or more terms. Three examples of lists are: (a, *joe,* b), ($f(a)$, $g(a, a)$, a), and (b, b, b, b). We will use the following terms to represent lists:

(1) The special constant *nil* is a list, representing the empty list.
(2) If t is a term and L is a list then $list(t, L)$ is also a list.

The following program will test or define lists:

$$IsaList(nil)$$
$$IsaList(list(E, L)) \leftarrow IsaList(L)$$

The definition of first and last element of list is

$$First(list(F, L), F) \leftarrow IsaList(L)$$
$$Last(list(L, nil), L)$$
$$Last(list(E, Y), L) \leftarrow Last(L, Y)$$

Table 5. Representing Data Structures

Goal Clause	Definite Clause	mgu
$\leftarrow Last(list(a, list(c, nil))), X)$		
$\downarrow$	$Last(list(E_1, L_1), X_1 \leftarrow Last(L_1, X_1)$	$\sigma_1 = \{E_1/a, L_1/list(b, list(c, nil)), X_1/X\}$
$\leftarrow Last(list(b, list(c, nil)), X)$		
$\downarrow$	$Last(list(E_2, L_2), X_2 \leftarrow Last(L_2, X_2)$	$\sigma_2 = \{E_2/b, L_2/list(c, nil), X_2/X\}$
$\leftarrow Last(list(c, nil), X)$		
$\downarrow$	$Last(list(X_3, nil), X_3)$	$\sigma_3 = \{X_3/c, X/c\}$
$\square$		

Suppose one wants to compute the last element of the list (a, b, c). This is translated into the goal clause $\leftarrow Last(list(a, list(b, list(c, nil))), X)$. The refutation for this goal is shown in Table 5. The computed answer is $\{X/c\}$. Our last example shows the definition of member:

$$Member(E, list(E, L)) \leftarrow IsaList(L)$$
$$Member(E, list(X, L)) \leftarrow Member(E, L)$$

In this program if one asks the query $\leftarrow Member(E, list(a, list(b, list(c, nil))))$, there will be, as expected, three computed answer substitutions: $\{E/a\}$, $\{E/b\}$, and $\{E/c\}$.

Summary

The discovery of Horn clauses has made programming in logic a reality. Perhaps the most evident contribution to computer science is the logic programming language *Prolog* (1,13). Logic programming has matured into a field in its own right and has found applications in such diverse fields as program verification, computational biology, and databases. Deductive databases are a direct descendent of Horn logic and have had a major impact on database technology. There are two major conferences dedicated to the presentation of research work in the area: the *North American Conference on Logic Programming* and the *International Conference of Logic Programming.* There is also a journal, the *Journal of Logic Programming,* which recently celebrated its 25th anniversary. One can also regularly find papers in the major journals of databases, programming languages, and artificial intelligence founded in logic programming concepts.

Logic programs are not restricted to Horn clauses and most logic programming environments include full clausal syntax and many variations of semantics (14). Some of the variations include allowing negation of atoms in the body of a clause, allowing a disjunction of atoms in a rule head, adding constraints to clauses, and allowing concurrency to play an integral part of the implementation of constructing computed answers. We invite the reader to browse over the conference proceedings of the logic programming conferences to find more about the recent direction of the field or to visit the *Association of Logic Programming* web site at http://www.cwi.nl/projects/alp/.

BIBLIOGRAPHY

1. J. Cohen, A view of the origins and development of Prolog, *Commun. ACM*, **31** (1): 26–36, 1988.
2. J. W. Lloyd, *Foundations of Logic Programming*, 2nd ed., New York: Springer-Verlag, 1987.
3. J. A. Robinson, A machine-oriented logic based on the resolution rule, *J. ACM*, **12** (1): 23–41, 1965.

4. C.-L. Chang, R. C.-T. Lee, *Symbolic Logic and Mechanical Theorem Proving*, New York: Academic Press, 1973.
5. D. Loveland, *Automated Theorem Proving: A Logical Basis*, New York: North-Holland, 1978.
6. K. R. Apt, Logic programming, in J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*, New York: North-Holland, 1990, Vol. B, pp. 493–574.
7. J.-L. Lassez, M. J. Maher, K. Marriott, Unification Revisited, in J. Minker (ed.), *Foundations of Deductive Databases and Logic Programming*, San Mateo, CA: Morgan Kaufmann, 1988, pp. 587–626.
8. L. Wos, *et al.*, *Automated Reasoning: Introduction and Applications*, Englewood Cliffs, NJ: Prentice-Hall, 1984.
9. R. Kowalski, D. Kuehner, Linear resolution with selection function, *Artif. Intell.*, **2**: 227–260, 1971.
10. R. Kowalski, Predicate Logic as a Programming Language, in *Proc. Inf. Proc. '74*, Stockholm: North-Holland, 1974, pp. 569–574.
11. K. R. Apt, M. H. van Emden, Contributions to the theory of logic programming, *J. ACM*, **29** (3): 841–862, 1982.
12. G. A. Ringwood, SLD: A folk acronym?, *ACM Sigplan Notices*, **24** (5): 71–75, 1989.
13. L. S. Sterling, E. Y. Shapiro, *The Art of Prolog*, Cambridge, MA: MIT Press, 1986.
14. J. Lobo, J. Minker, A. Rajasekar, *Foundations of Disjunctive Logic Programming*, Cambridge, MA: MIT Press, 1990.

JOHN M. JEFFREY
JORGE LOBO
Elmhurst College
TADAO MURATA
University of Illinois at Chicago

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Integral Equations
Standard Article
S. M. Rao[1] and G. K. Gothard[1]
[1]Auburn University, Auburn, AL
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2428
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (180K)

Abstract

The sections in this article are

Integral Equations

Method of Moments Solution

Sparse Matrix Methods

| | |

Copyright © 1999-2008 All Rights Reserved.

INTEGRAL EQUATIONS

Mathematics play a very important role in all the areas of electrical engineering. Whenever we are asked to develop a system or address a problem, the first thing we need to do is to develop a simple model. Many a times, this simple model turns out to be a mathematical model. The mathematical model lets us study many important aspects of the problem thoroughly and in an inexpensive manner.

In this article, we deal with an area of mathematics known as *integral equations.* We define an equation as an integral equation when the unknown quantity, i.e., the quantity to be determined, is under an integral sign. Integral equations are usually formulated when it is required to obtain the driving mechanism (input) of a physical system, given the description of the system along with the response function (output). For electrical engineers, the physical system may be an electrical circuit, an electrical machine, or, sometimes, a complex structure such as a fighter aircraft whose electromagnetic signature is the quantity of interest. Similarly, in many situations in electrical engineering, the response function may, simply, be the voltage at some given terminals or the current flowing in a wire.

There are several methods to solve integral equations (1) using complex mathematics. However, in many practical situations, these methods are inadequate and, quite often, we need to resort to numerical methods to solve these equations. In the following section, we formally introduce integral equations using simple mathematical language. We also introduce standard terminology to describe such equations and describe various types of integral equations. In the second section, we describe a general numerical method, known as method of moments, to solve these equations. In the third section, we present a new technique which makes the method of moments technique computationally more efficient along with a set of numerical results. Note that, although the topic of integral

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

equations is really a mathematical subject, we develop the subject by using examples from electrical engineering and in fact, from electromagnetic theory. It must be clearly understood that this way of treatment of the subject does not necessarily preclude the application of the techniques discussed in this article into other areas of engineering.

INTEGRAL EQUATIONS

Mathematically speaking, an equation involving the integral of an unknown function of one or more variables is known as integral equation. One of the most common integral equations encountered in electrical engineering is the *convolution integral* given by

$$\int X(\tau)H(t,\tau)d\tau = Y(t) \tag{1}$$

In Eq. (1), we note that the response function $Y(t)$ and the system function $H(t, \tau)$ is known and we need to determine the input $X(\tau)$. Of course, if $X(\tau)$ and $H(t, \tau)$ are known and we need to determine $Y(t)$, then Eq. (1) simply represents an integral relationship which can be performed in a straightforward manner. We further note that $H(t, \tau)$ is also commonly known as *impulse response* if Eq. (1) represents the system response of a linear system. In general, in mathematics and in engineering literature, $H(t, \tau)$ is known as *Green's function* or *kernel function*. We also acknowledge that, for some other physical systems, $Y(t)$ and $X(t)$ may represent the driving force and response functions, respectively.

Next, we note that Eq. (1) is known as integral equation of first kind. We also have another type of integral equation given by

$$C_1X(t) + C_2\int X(\tau)H(t,\tau)d\tau = Y(t) \tag{2}$$

where C_1 and C_2 are constants.

In Eq. (2), we note that the unknown function $X(t)$ appears both inside and outside the integral sign. Such equation is known as the integral equation of second kind. Further, we also see in electrical engineering yet another type of integral equation given by

$$C_1\int X(\tau)H(t,\tau)d\tau + C_2X(t) + C_3\frac{dX(t)}{dt} = Y(t) \tag{3}$$

which is known as *integro-differential equation.*

It may be noted that for a limited number of kernel and response functions, in Eqs. (1–3), it is possible to obtain the solution using analytical methods. Several textbooks have been written to discuss the mathematical aspects of the integral equations from an analytical point of view (2–4). However, for a majority of practical problems, these equations can be solved using numerical methods only. Fortunately, in this day and age, we can obtain very accurate numerical solutions owing to the availability of fast digital computers. In the following section, we discuss a general numerical technique, popularly known as *method of moments,* to solve the integral Eqs. (1–3).

METHOD OF MOMENTS SOLUTION

The method of moments (MoM) solution procedure was first applied to electromagnetic scattering problems by Harrington (5). Consider a linear operator equation given by

$$AX = Y \tag{4}$$

where A represents the integral operator, Y is the known excitation function, and X is the unknown response function to be determined. Now, let X be represented by a set of known functions, termed as basis functions or expansion functions, $(p_1, p_2, p_3, \ldots,)$ in the domain of A as a linear combination:

$$X = \sum_{i=1}^{N}\alpha_i p_i \tag{5}$$

where α_i values are scalars to be determined. Substituting Eq. (5) into Eq. (4), and using the linearity of A, we have

$$\sum_{i=1}^{N}\alpha_i Ap_i = Y \tag{6}$$

where the equality is usually approximate. Let $(q_1, q_2, q_3, \ldots)$ define a set of testing functions in the range of A. Now, multiplying Eq. (6) with each q_j and using the linearity property of the inner product, we obtain

$$\sum_{i=1}^{N}\alpha_i\langle q_j, Ap_i\rangle = \langle q_j, Y\rangle \tag{7}$$

for $j = 1, 2, \ldots, N$. The set of linear equations represented by Eq. (7) may be solved using simple matrix methods to obtain the unknown coefficients α_i.

The simplicity of the method lies in choosing the proper set of expansion and testing functions to solve the problem at hand. Further, the method provides a most accurate result if properly applied. However, for the integral equation operators, the method generates a dense matrix which may be expensive in terms of computer storage requirements when complex systems are involved. In the following subsections, we discuss the application of the method of moments to some commonly used integral equations in engineering and science.

Integral Equations without Derivatives

In this section, we develop simple numerical methods to solve integral equations (both first and second kind) applying the method of moments. Further, we restrict our treatment to integral equations with single independent variable (one-dimension) only. The extension to multiple variables is straightforward and hence is not considered here. The numerical methods are general methods, and thus applicable to a variety of practical problems.

Consider an integral equation given by

$$\int_{x'=-w}^{w} u(x')g(x,x')dx' = f(x) \qquad x \in (-w, w) \tag{8}$$

in which $u(x)$ is the unknown function to be determined. For the method of moments analysis of such problems, we develop a numerical scheme known as collocation method, subdomain

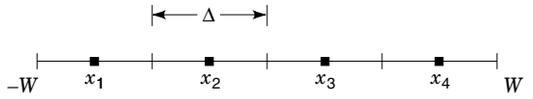


Figure 1. Match points for the integral equation.

method, or point matching method. For this procedure, we first divide the interval $-w$ to w into N equal segments of width Δ as shown in Fig. 1.

The segment center points are given by

$$x_i = -w + 0.5(i-1)\Delta \qquad i = 1, 2, \ldots, N \tag{9}$$

Note that while defining Eq. (9), we have divided the interval $-w$ to w into equal segments, although this need not be the case in general.

The next step in the method of moments solution procedure is to define a suitable set of basis and testing functions. Our research shows that, for this type of problem, i.e., the integral equations with no derivatives, the most convenient and simple set of functions are pulse functions with unit amplitude as basis functions and Dirac delta distributions (functions) as testing functions. In the following, we formally define these functions, as shown in Fig. 2, given by

$$p_i(x) = \begin{cases} 1 & x_i - \dfrac{\Delta}{2} \le x \le x_i + \dfrac{\Delta}{2} \\ 0 & \text{Otherwise} \end{cases} \tag{10}$$

and

$$q_j(x) = \delta(x - x_j) \tag{11}$$

Here, we emphasize that Eqs. (10) and (11) are by no means the only set of functions used in practice. It is quite possible to define a completely different set of functions as long as these functions satisfy a certain set of conditions (6–8). Further, it is also possible to carry-out an entirely different scheme in which the expansion and testing functions are defined over the whole interval without ever dividing the solution region into subsections. Such numerical schemes are known as *entire domain methods.* Entire domain methods are known to be mathematically unstable (5), which may be overcome by a suitable choice of testing and basis functions or a combination of subdomain/entire domain functions (9). However, we will not present the numerical treatment with entire domain functions in this work since the subject is still in research stage.

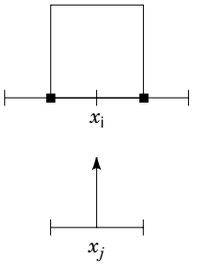


Figure 2. Pulse function and delta function.

First, we shall consider the testing procedure. Here, we multiply the Eq. (8) by the testing function q_j and integrate over the whole interval to obtain a set of equations given by

$$\int_{x'=-w}^{w} u(x')g(x_j, x')dx' = f(x_j) \qquad j = 1, 2, \ldots N \tag{12}$$

Observe that, while evaluating Eq. (12), we made use of the well-known properties of the delta distribution (function). Also note that Eq. (12) is actually a set of N equations for each j, and x_j represents the value of the independent variable at the center of the jth subdomain. Further, observe that we are matching the left and right hand sides of Eq. (12) at points x_j for $j = 1, 2, \ldots, N$. Thus, these points are known also as *match points.*

Next we consider the expansion procedure. By using the basis functions defined in Eq. (10), the unknown quantity $u(x)$ may be written as

$$u(x) = \sum_{i=1}^{N} \alpha_i p_i \tag{13}$$

where α's represent the unknown scalar coefficients. Substituting Eq. (13) into Eq. (12), we have

$$\sum_{i=1}^{N} \alpha_i \int_{x'=x_i-\Delta/2}^{x_i+\Delta/2} g(x_j, x')dx' = f(x_j) \qquad j = 1, 2, \ldots N \tag{14}$$

Note that, Eq. (14) may be written as a matrix equation, given by

$$[Z][I] = [V] \tag{15}$$

where

$$Z_{ji} = \int_{x'=x_i-\Delta/2}^{x_i+\Delta/2} g(x_j, x')dx' \tag{16}$$

$$V_j = f(x_j) \tag{17}$$

and the column vector $[I]$ contains unknown coefficients α's. Except for certain special cases, the matrix $[Z]$ is a well-conditioned matrix and hence the solution of Eq. (15) is straightforward. Also, the integrations involved in Eq. (16) may be either performed analytically or numerically depending on the exact nature of the kernel function.

Lastly, the numerical method described so far is also known as pulse expansion and point matching method. In the following, we present an example problem based on the procedure described so far.

Example. Consider an infinitely long conducting strip of width of 0.1 m located symmetrically at the origin as shown

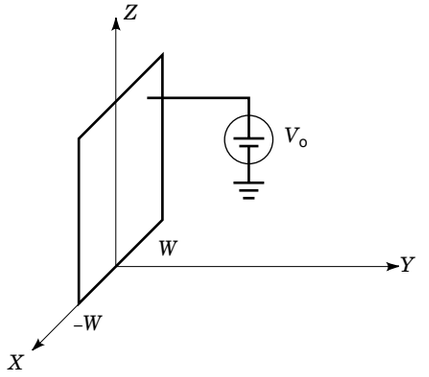


Figure 3. Infinite strip raised to 1 V potential.

in Fig. 3. The strip is raised to a potential of 1 V. Note that the reference point (i.e., $V = 0$) is at $x = 1$ m. Calculate the charge distribution on the strip.

SOLUTION. Following the basic principles of electrostatics, an integral equation may be developed, given by

$$\int_{x'=-0.05}^{0.05} q_s(x') \ln|x - x'| dx' = 2\pi\epsilon_0 \qquad x \in (-0.05, 0.05) \quad (18)$$

where $\epsilon_0 = 8.854e - 12$ is the permittivity of the surrounding medium. Following the numerical procedures described so far, we obtain the elements of the $[Z]$-matrix given by

$$\begin{aligned} Z_{ji} &= \int_{x'=x_i-\Delta/2}^{x_i+\Delta/2} \ln|x_j - x'| dx' \\ &= \Delta - \frac{\Delta}{2} \ln|(x_j - x_i)^2 - (\Delta/2)^2| \\ &\quad - (x_j - x_i) \ln \frac{|x_j - x_i + \Delta/2|}{|x_j - x_i - \Delta/2|} \end{aligned} \quad (19)$$

and the elements of $[V]$-matrix are

$$V_j = 2\pi\epsilon_0 \quad (20)$$

In Fig. 4, we present the charge distribution for N equal to 10, 50, and 100 obtained by solving the integral Eq. (18).

Integral Equations with Derivatives

In this section, we develop simple numerical methods to solve integrodifferential equations, i.e., integral equations with derivative operators, applying the method of moments. As before, we restrict our treatment to integral equations with a single independent varaible (one-dimension) only. The extension to multiple variables is straightforward and hence is not considered here. The numerical methods are general methods, and thus applicable to a variety of practical problems.

We consider two cases in this section: the first-order integrodifferential equation, and the second-order integrodifferential equation. Obviously, higher order derivatives may be handled in a similar manner.

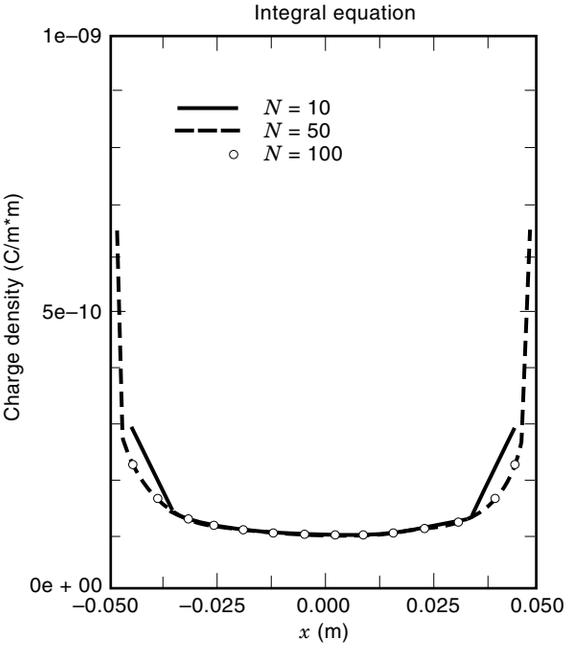


Figure 4. Charge density distribution on the infinite strip.

First Order Integrodifferential Equation. Consider a first-order integrodifferential equation given by

$$\frac{\partial}{\partial x} \int_{x'=-w}^{w} u(x') g(x, x') dx' = f(x) \qquad x \in (-w, w) \quad (21)$$

subject to

$$\int_{x=-w}^{w} u(x) dx = 0 \quad (22)$$

The Eq. (22) is also known as a *constraining equation.* In a variety of situations, constraining equations can be implicitly enforced by a proper choice of basis or testing functions. This necessitates a more elaborate construction of basis/testing functions which, although it seems to be complicated, results in an efficient numerical solution. It is quite easy to see that a straightforward application of the method discussed in the previous section, i.e., pulse-expansion and point matching method, results in $N \times N$ matrix. However, the application of the constraint equation adds one more column to the $[Z]$-matrix, thus making the problem over-determined system. Further, other numerical problems, such as stability and nonuniqueness, set in when MoM is applied blindly. Thus, we develop the following numerical procedure for this case.

As before, the interval $(-w, w)$ is divided into N equal segments. But for this case, the match points are labled in the following way for mathematical convenience as shown in Fig. 5.

$$x_i = -w + i \times \Delta \qquad i = 1, 2, \ldots, N-1 \quad (23)$$

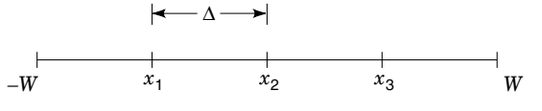


Figure 5. Match points for integrodifferential equation.

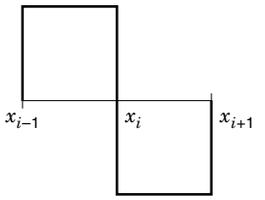


Figure 6. Pulse-doublet function.

In order to enforce the constraining Eq. (22), we let the basis function to overlap over two subdomains with positive unit height in the first subdomain and negative unit pulse in the second subdomain, as shown in Fig. 6.

Thus, mathematically, we define the basis function as

$$p_i(x) = \begin{cases} 1 & x_{i-1} \le x \le x_i \\ -1 & x_i \le x \le x_{i+1} \\ 0 & \text{Otherwise} \end{cases} \tag{24}$$

and express the unknown quantity $u(x)$ as

$$u(x) = \sum_{i=1}^{N-1} \alpha_i p_i \tag{25}$$

Notice that, by defining basis functions as in Eq. (24), Eq. (22) is automatically satisfied, which can be proved as

$$\begin{aligned} \int_{x=-w}^{w} u(x)dx &= \sum_{i=1}^{N} \alpha_i \int p_i dx \\ &= \sum_{i=1}^{N} \alpha_i \left[\int_{x_{i-1}}^{x_i} dx - \int_{x_i}^{x_{i+1}} dx \right] \\ &= 0 \end{aligned} \tag{26}$$

The functions defined by Eq. (24) are known as *pulse doublet functions.*

Next, we define the testing procedure for this case. Notice that we have one derivative on the integral sign. By simple mathematical manipulation, we transform the derivative operator onto the testing function q_j. By using a compact notation

$$\langle f, g \rangle = \int fg dx \tag{27}$$

we can write the integrodifferential Eq. (21) as

$$\left\langle \frac{\partial v}{\partial x}, q_j \right\rangle = \langle f(x), q_j \rangle \tag{28}$$

where

$$v(x) = \int_{x'=-w}^{w} u(x')g(x, x')dx' \tag{29}$$

Then, we have

$$\begin{aligned} \left\langle \frac{\partial v}{\partial x}, q_j \right\rangle &= \int \frac{\partial v}{\partial x} q_j dx \\ &= [q_j v] - \int \frac{\partial q_j}{\partial x} v dx \end{aligned} \tag{30}$$

The first term in Eq. (30) can be set to zero if $q_j = 0$ at the ends of the subdomain.

Keeping this procedure in mind, we select the testing functions in such a way that when the derivative is transformed onto the testing function the result must be a delta distribution (function). A unit pulse function, as shown in Fig. 7, has this property whose derivative happens to be two delta distributions on either end of the pulse.

Thus, for first-order integrodifferential equations, we choose the testing function q_j as

$$q_j(x) = \begin{cases} 1 & x_j - \dfrac{\Delta}{2} \le x \le x_j + \dfrac{\Delta}{2} \\ 0 & \text{Otherwise} \end{cases} \tag{31}$$

The numerical procedure may be best illustrated by the following example.

Example. Consider that an infinitely long conducting strip of width 1 m, as shown in Fig. 3, is immersed in an electrostatic field. Calculate the charge distribution on the strip.

SOLUTION. Following the basic principles of electrostatics, and applying the electric field boundary condition on perfect conducting bodies, an integral equation may be developed, given by

$$\frac{\partial}{\partial x} \int_{x'=-w}^{w} q_s(x') \ln|x - x'| dx' = 2\pi\epsilon \boldsymbol{a}_x \cdot \boldsymbol{E}^{\mathrm{i}} \qquad x \in (-w, w) \tag{32}$$

subject to

$$\int_{x=-w}^{w} q_s(x)\, dx = 0 \tag{33}$$

where $\boldsymbol{E}^i$, q_s, and $\boldsymbol{a}_x$ are the impressed electric field, charge density, and the x-directed unit vector, respectively. For the numerical solution, we divide the interval $(-w, w)$ into N subdomains of width Δ and label the match points as shown in Fig. 5. Notice that when the interval is divided into N divisions, we actually have $N - 1$ match points.

Defining the testing functions by Eq. (31), and carrying out the mathematical steps outlined in Eq. (30), we get

$$\begin{aligned} \int_{x'=-w}^{w} q_s(x') \ln\left| x_j + \frac{\Delta}{2} - x' \right| dx' \\ - \int_{x'=-w}^{w} q_s(x') \ln\left| x_j - \frac{\Delta}{2} - x' \right| dx' \\ = 2\pi\epsilon \Delta \boldsymbol{a}_x \cdot \boldsymbol{E}^{\mathrm{i}}(x_j) \end{aligned} \tag{34}$$

for $j = 1, 2, \ldots, N - 1$.

Next, we apply the expansion procedure. By selecting the basis functions as described in Eq. (25), the constraining Eq.

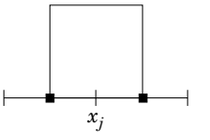


Figure 7. Pulse testing function.

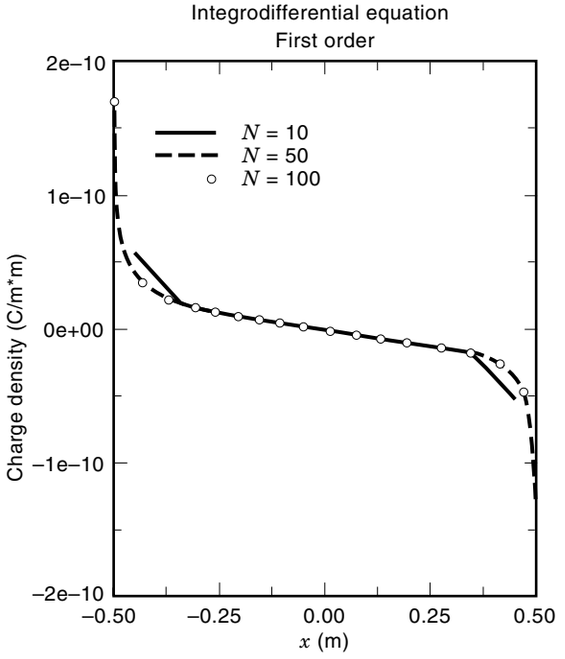


Figure 8. Charge density distribution on the infinite strip immersed in electric field $\boldsymbol{E}^i = \boldsymbol{a}_x$.

(33) is automatically enforced. Thus, applying the method of moments procedure, we obtain $[Z][I] = [V]$, where

$$\begin{aligned} Z_{ji} = &\int_{x_{i-1}}^{x_i} \ln\left|x_j + \frac{\Delta}{2} - x'\right| dx' \\ &- \int_{x_i}^{x_{i+1}} \ln\left|x_j + \frac{\Delta}{2} - x'\right| dx' \\ &- \int_{x_{i-1}}^{x_i} \ln\left|x_j - \frac{\Delta}{2} - x'\right| dx' \\ &+ \int_{x_i}^{x_{i+1}} \ln\left|x_j - \frac{\Delta}{2} - x'\right| dx' \end{aligned} \tag{35}$$

and

$$V_j = 2\pi\epsilon\Delta\boldsymbol{a}_x \cdot \boldsymbol{E}^i(x_j) \tag{36}$$

In Fig. 8, we present the charge distribution for N equal to 10, 50, and 100 obtained by solving the integrodifferential Eq. (32). Notice that, in this procedure, the dimension of the system matrix is $N - 1$.

Second-Order Integrodifferential Equation. In this section, we consider techniques for solving the integrodifferential equation

$$\frac{\partial^2}{\partial x^2}\int_{x'=-w}^{w} u(x')g(x,x')\,dx' = f(x) \qquad x \in (-w, w) \tag{37}$$

where the unknown function $u(x)$ must satisfy the boundary conditions

$$u(w) = u(-w) = 0$$

These types of integral equations usually appear in electromagnetic and acoustic scattering problems, the most common being the dipole antenna problem in antenna engineering. Further, the treatment of second-order integrodifferential equation, coupled with the treatment of first-order derivatives, provides a solution procedure for handling higher order derivatives.

We begin our analysis by rewriting the integrodifferential Eq. (37) in the following form:

$$\frac{\partial}{\partial x}\int_{x'=-w}^{w} u(x')\frac{\partial g(x,x')}{\partial x}\,dx' = f(x) \qquad x \in (-w, w) \tag{38}$$

For almost all mathematical problems in engineering, there exists a definite relationship between $\partial g/\partial x$ and $\partial g/\partial x'$. In fact, for electromagnetic (EM) and acoustic scattering problems, we have $\partial g/\partial x = -\partial g/\partial x'$. Using this relationship, we can write Eq. (38), at least for EM and acoustic problems, as

$$\frac{\partial}{\partial x}\int_{x'=-w}^{w} \frac{\partial u(x')}{\partial x'} g(x,x')\,dx' = f(x) \qquad x \in (-w, w) \tag{39}$$

Now, we have an integrodifferential equation of first order which we already know how to handle. At first, we divide the interval $(-w, w)$ into N segments and label $N - 1$ match points as shown in Fig. 5. The definition of testing functions and the testing procedure is identical to the case of first-order integrodifferential equation and hence need not be repeated again. However, we need to look more closely at the basis functions.

Note that, for the case of first-order integrodifferential equations, we defined the pulse doublet as the expansion function and obtained the solution for the unknown function. In the present case we can do the same thing, if we define the antiderivative of pulse doublet as the expansion function. Following this logic, we define the basis functions for the solutions of second-order integrodifferential equation as

$$p_i(x) = \begin{cases} 1 - \dfrac{x_i - x}{\Delta} & x_{i-1} \le x \le x_i \\ 1 + \dfrac{x_i - x}{\Delta} & x_i \le x \le x_{i+1} \\ 0 & \text{Otherwise} \end{cases} \tag{40}$$

The functions described in Eq. (40), and shown in Fig. 9, are popularly known as *Triangle functions,* which are linear piece-wise.

Thus, for the solution of second-order integrodifferential equations, we employ triangle function expansion and pulse function testing. We describe the numerical procedure using the following example.

Example. Consider a finite-length straight wire, radius $a = 0.001\lambda$, and length $2h = 0.5\lambda$ illuminated by an electromag-

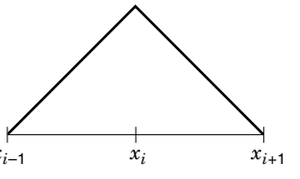


Figure 9. Triangle basis function.

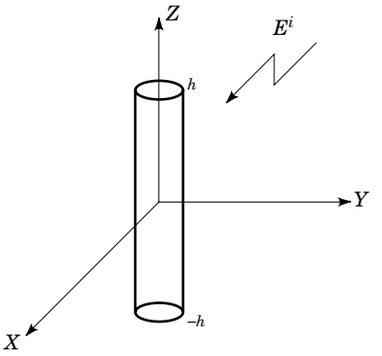


Figure 10. Straight wire illuminated by a plane wave.

netic plane wave (wave length λ) as shown in Fig. 10. Calculate the current induced on the wire.

SOLUTION. Since the radius a is very small compared to λ and h we can use the thin-wire theory (10) to formulate the integrodifferential equation. Following the mathematical procedures described in (11), we derive the following integral equation, given by

$$\frac{\partial}{\partial z}\int_{z'=-h}^{h}\frac{\partial I(z')}{\partial z'}G(z-z')\,dz' + k^2\int_{z'=-h}^{h} I(z')G(z-z')dz' = -j\frac{4\pi k}{\eta}E_z^i(z) \qquad z\in(-h,h) \tag{41}$$

where

$$G(z-z') = \frac{e^{-jkR}}{R} \tag{42}$$

and

$$R = \sqrt{(z-z')^2+a^2} \tag{43}$$

In Eqs. (41–43), I is the unknown current induced on the wire, $E_z^i(z)$ is the z-component of the incident plane wave, $k = 2\pi/\lambda$ is the wave number, and η is the wave impedance of the surrounding medium.

First of all, divide the wire region $(-h, h)$ into N equal segments labeling $N-1$ match points as shown in Fig. 5. Next, for this problem, we choose the expansion functions p_i defined in Eq. (40) to express the unknown current I and the testing functions q_j defined in Eq. (31).

Thus, we have

$$I = \sum_{i=1}^{N-1}\alpha_i p_i \tag{44}$$

Next, we consider the testing procedure. By following the same procedures of the previous section on first-order integrodifferential equations, the testing procedure yields,

$$\int_{z'=-h}^{h}\frac{\partial I(z')}{\partial z'}G\left(z_j+\frac{\Delta}{2}-z'\right)dz' - \int_{z'=-h}^{h}\frac{\partial I(z')}{\partial z'}G\left(z_j-\frac{\Delta}{2}-z'\right)dz' + \Delta k^2\int_{z'=-h}^{h} I(z')G(z_j-z')dz' = -j\frac{4\pi k\Delta}{\eta}E_z^i(z_j) \tag{45}$$

for $j = 1, 2, \ldots, N-1$. Notice that, in Eq. (45), the integrations on the second term and the right hand side of the Eq. (41) are approximated by a simple one-point rule.

Substituting the expansion Eq. (44) into Eq. (45), we obtain the matrix equation $[(1/\Delta)[Z^a] + (k^2\Delta)[Z^b]]\,[I] = [V]$ where the matrix elements are:

$$\begin{aligned} Z_{ji}^a = &\int_{z_{i-1}}^{z_i} G\left(z_j+\frac{\Delta}{2}-z'\right)dz' \\ &-\int_{z_i}^{z_{i+1}} G\left(z_j+\frac{\Delta}{2}-z'\right)dz' \\ &-\int_{z_{i-1}}^{z_i} G\left(z_j-\frac{\Delta}{2}-z'\right)dz' \\ &+\int_{z_i}^{z_{i+1}} G\left(z_j-\frac{\Delta}{2}-z'\right)dz' \end{aligned} \tag{46}$$

$$\begin{aligned} Z_{ji}^b = &\int_{z_{i-1}}^{z_i}\left\{1-\frac{z_i-z}{\Delta}\right\}G(z_j-z')dz' \\ &+\int_{z_i}^{z_{i+1}}\left\{1+\frac{z_i-z}{\Delta}\right\}G(z_j-z')dz' \end{aligned} \tag{47}$$

and

$$v_j = -j\frac{4\pi k\Delta}{\eta}E_z^i(z_j) \tag{48}$$

The integrations involved in Eqs. (46) and (47) may be carried out using the methods discussed in (12).

In Fig. 11, we present the current induced on a half-wave dipole wire scatterer due to a unit-amplitude, normally incident plane wave for N equal to 20 and 50 divisions obtained by using Eqs. (46–48).

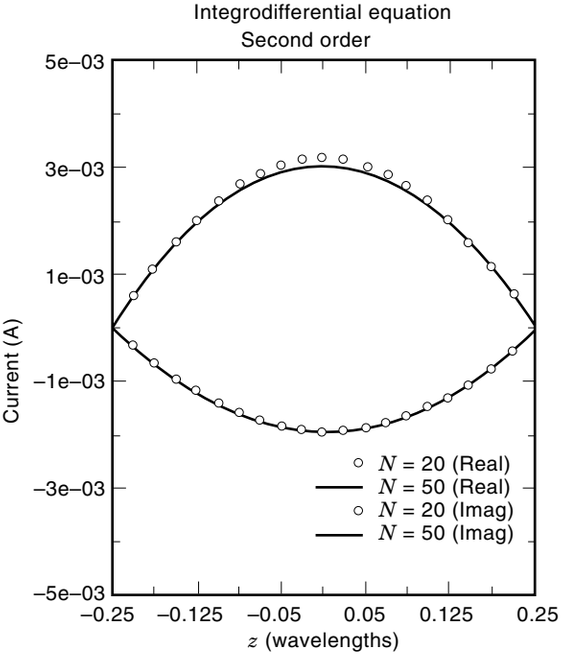


Figure 11. Current induced on the wire scatterer.

Integral Equations with More Variables

In the previous subsection, we discussed numerical methods applying the method of moments to handle integral and integrodifferential equations with one independent variable. Extension to multivariable case is straightforward and follows the same numerical procedures discussed so far. For example, for two-variable case, i.e. $x - y$ plane, the solution region may be divided into square or rectangular cells and one can construct the basis and testing functions using the methods discussed in the previous section (13). For a more general situation, the solution region can be divided into triangular subdomains along with suitable basis and testing functions (14,15). Efficient solutions have been obtained for very complex problems using these methods in electromagnetics and acoustics (16–19) and it is quite possible that these methods found applications in other areas of engineering. Lastly, we mostly discussed only boundary-value problems in this work but solutions have also been obtained for initial value problems (20–28) using the same methods. An extensive application of method of moments to electromagnetic scattering problems may be obtained in (29).

SPARSE MATRIX METHODS

One major problem with MoM is the generation of a dense matrix and for complex problems, the dimension of this matrix can be prohibitively large. Usually, for electromagnetic and acoustic scattering problems, it is necessary to divide the solution region into small enough subdomains in order to obtain accurate results. By "small enough," we mean about 200 to 300 subdomains per square wavelength. In usual practice, we may typically solve for several thousand unknowns for large, complex problems. Quickly, this requirement becomes expensive in terms of computational resources and may even become impossible to handle. Hence, we look for alternate schemes to reduce the computational resources by generating a sparse matrix instead of a full matrix.

The generation of a sparse matrix in the method of moment solution procedure may be achieved in two ways: (a) by defining a special set of basis functions to represent the unknown quantity or (b) by handling the influence of the kernel function in a novel way. The usage of well-known, wavelet-type basis functions to provide the required sparsity belongs to the former category (30) and the application of fast multipole method (FMM) belongs to the latter category (31). So far, the wavelet-type basis functions have been applied to integral equations with one variable only, and it remains to be seen how these functions can be utilized for two or more variable cases. In contrast, in the FMM scheme, the matrix-vector product is carried out in a novel way and seems to work well for more complex problems. Unfortunately, the FMM is a complicated scheme and any reasonable summary of the method is beyond the scope of the present article.

There is yet another scheme, known as impedance matrix localization (IML), which achieves modest sparsity for simple problems (32). Notice that the kernel function is, in general, a decaying function with respect to the distance between the source and observation points. Thus, with increasing distances, the influence of a given source becomes negligible at a sufficiently distant observation point and may be actually set to zero. The IML scheme cleverly exploits this fact. However, there is a certain degree of arbitrariness in this scheme and it seems to work for simple problems only.

Recently a new method, known as generalized sparse matrix reduction scheme (GSMR), is proposed, which seems to improve on the IML method.

The basic concept utilized in the GSMR technique may be qualitatively illustrated as follows. Following similar procedures of the MoM, a moment matrix is also generated in the GSMR method. However, in contrast to the conventional moment method where interaction is computed from each and every cell on other cells, only the interaction from the self-cell and few neighboring cells is computed in the GSMR technique. In fact, for single variable problems (wire scatterer and two-dimensional, infinite cylinders) only the self-term and two neighboring terms on either side of the self-cell are generated in this technique. This implies that the moment matrix for the GSMR technique is essentially sparse. Further, the effect of nonself terms is taken into account by defining a set of linearly independent functions over the entire structure. In the mathematical terms the procedure, for single variable problems, may be described as follows:

Let $[Z]$ represent the moment matrix for a given problem generated by using appropriate basis and weighting functions. Note that, for well-defined problems with proper choice of basis and testing functions, the moment matrix is well-conditioned and diagonally strong. The jth row of the moment matrix may be written as

$$\sum_{i=1}^{N} Z_{j,i} I_i = V_j \tag{49}$$

where all the matrix elements $Z_{j,i}$ are nonzero. In the new GSMR technique, the jth row is modified as

$$\sum_{i=j-1}^{j+1} \alpha_{j,i} Z_{j,i} I_i = \Gamma_j V_j \tag{50}$$

where $\alpha_{j,j-1}$, $\alpha_{j,j}$, $\alpha_{j,j+1}$, and Γ_j are the unknown coefficients and the rest of terms in the row are set to zero. Further, dividing by $Z_{j,j}$, Eq. (50) may be rewritten as

$$\sum_{i=j-1}^{j+1} \beta_{j,i} I_i = \gamma_j V_j \tag{51}$$

which may be written, using the matrix notation, as

$$[\boldsymbol{\beta}][I] = [V] \tag{52}$$

where $[\boldsymbol{\beta}]$ is a sparse matrix with, at most, three nonzero elements per row.

Upon a close examination of Eq. (52), it is obvious that one needs to reconstruct the $[\boldsymbol{\beta}]$-matrix. This task may be accomplished by first setting $\gamma_j = 1$ for $j = 1, \ldots, N$ in Eq. (51).

Next, define three linearly independent functions, $I^{(1)}$, $I^{(2)}$, and $I^{(3)}$, over the entire domain of the problem. These functions may be thought of as source distributions. For the examples we discuss below, these functions are assumed to be a constant, $\cos(kl)$ and $\sin(kl)$ where $k = 2\pi/\lambda$ is the wave number and l is the parameter measured along the length of the independent variable in the integral equation.

The next step in the GSMR technique is to compute the corresponding response functions, $V^{(1)}$, $V^{(2)}$, and $V^{(3)}$. This task

may be easily accomplished by using the assumed source distributions $I^{(1)}$, $I^{(2)}$, and $I^{(3)}$, and utilizing the Green's function for the problem.

Once we have $I^{(1)}$, $I^{(2)}$, $I^{(3)}$, $V^{(1)}$, $V^{(2)}$, and $V^{(3)}$, the $[\boldsymbol{\beta}]$-matrix may be constructed as follows:

- For any j, sample $I^{(1)}$, $I^{(2)}$, and $I^{(3)}$ at locations $j-1$, j, and $j+1$, and sample $V^{(1)}$, $V^{(2)}$, and $V^{(3)}$ at location j, and write the following system of equations:

$$\begin{aligned}
\beta_{j,j-1}I_{j-1}^{(1)} + \beta_{j,j}I_j^{(1)} + \beta_{j,j+1}I_{j+1}^{(1)} &= V_j^{(1)} \\
\beta_{j,j-1}I_{j-1}^{(2)} + \beta_{j,j}I_j^{(2)} + \beta_{j,j+1}I_{j+1}^{(2)} &= V_j^{(2)} \\
\beta_{j,j-1}I_{j-1}^{(3)} + \beta_{j,j}I_j^{(3)} + \beta_{j,j+1}I_{j+1}^{(3)} &= V_j^{(3)}
\end{aligned} \tag{53}$$

- Solve Eq. (53) to obtain $\beta_{j,j-1}$, $\beta_{j,j}$, and $\beta_{j,j+1}$ and store in the jth row of the $[\boldsymbol{\beta}]$-matrix.
- Repeat the previous two steps for all values of j.

Further, note that for $j = 1$ and $j = N$, we select $\beta_{1,N}$, $\beta_{1,1}$, and $\beta_{1,2}$, and, $\beta_{N,N-1}$, $\beta_{N,N}$, and $\beta_{N,1}$, respectively.

Once all the coefficients for each row are computed, we have successfully generated the new matrix representation for the integral equation. Finally, Eq. (52) may be solved efficiently using iterative methods such as the conjugate gradient method (32) or the GMRES method (33) since we are dealing with sparse matrices.

Example. Consider a 10λ straight wire, 0.001λ radius, illuminated by a normally incident plane wave. The matrix size for the MoM and GSMR method is 149×149 and 149×3, respectively. The results are shown in Fig. 12 and the comparison is excellent.

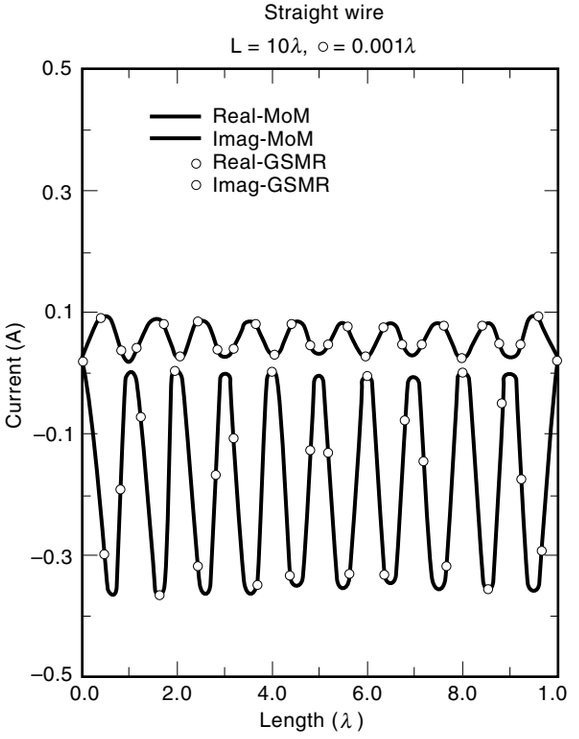


Figure 12. Current induced on the 10λ wire scatterer.

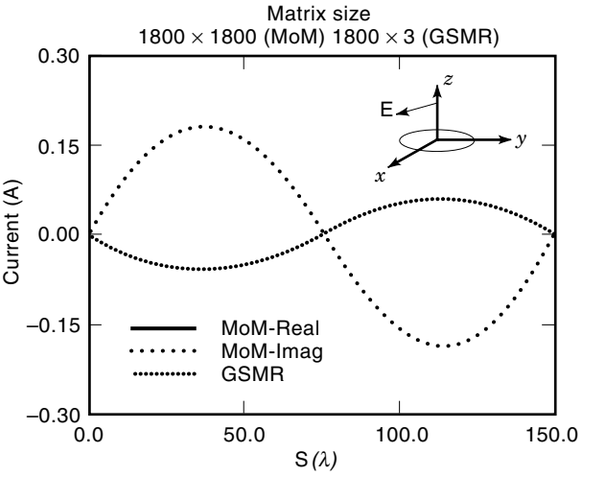


Figure 13. Current induced on the circular loop.

Example. Consider the case of a circular loop located in the $z = 0$ plane with center at the origin. The loop is illuminated by an x-polarized plane wave traveling along the z-axis. Figure 13 shows the results for $ka = 150$ where k and a are the wave number and the radius of the loop, respectively. The matrix size for the MoM and the GSMR technique 1800 is $\times$ 1800 and 1800×3, respectively. It is evident from the figure that the results compare very well with each other. This example clearly illustrates the applicability of the GSMR method for truly large bodies.

Example. Lastly, we present the case of an infinitely long, conducting strip illuminated by a transverse magnetic (TM) incident electromagnetic plane wave. The derivation of the governing integral equation for this problem may be found in (5). Figure 14 shows the current density induced on a 150λ bent strip obtained by applying MoM and GSMR techniques.

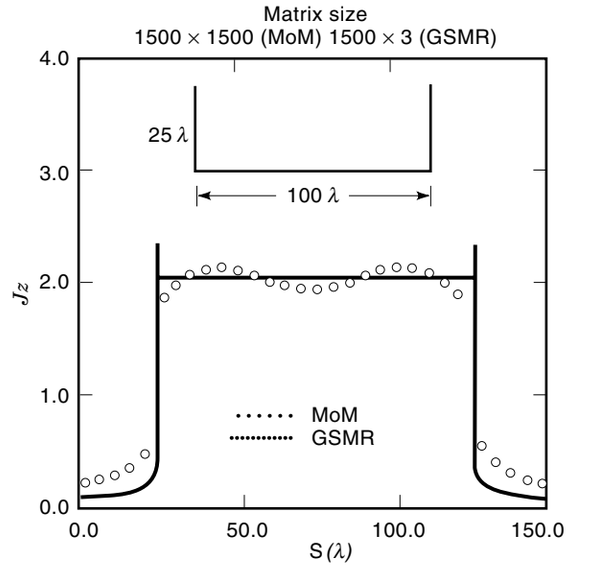


Figure 14. Current induced on the conducting bent strip by a TM incident plane wave. The cross-section of the strip is shown in the inset.

The comparison between both methods is reasonably accurate for both cases as evident from the figure.

Lastly, before closing the discussion on GSMR technique, the existence of $\boldsymbol{\beta}$ matrix may be explained in the following way. It may be noted that the moment matrix generated in the conventional MoM solution procedure is a representation of the unique relationship that exists between the source and the response. This relationship is specified in mathematical terms via the Green's function along with the boundary conditions. Further, this relationship holds for any source distribution and response function as long as the response function is derived utilizing the Green's function satisfying the appropriate boundary conditions. Since $\boldsymbol{\beta}$ matrix is developed using this unique relationship, Eq. (52) must represent the discretized form of the operator equation. Further, it should be noted that, although the operator equation is unique, the matrix representation is not necessarily unique. This is quite obvious since different basis and testing functions result in a different matrix representation. Also, one can perform elementary row and column operations on the given system of equations and arrive at another representation of the same operator equation.

However, as a word of caution, it may be noted that the GSMR technique is a recent concept and tested only on some simple problems. It is obvious that the procedure needs to be validated for more complex geometries. Presently, work is in progress to apply the GSMR technique to some of these cases.

BIBLIOGRAPHY

1. W. Pogorzelski, *Integral Equations and Their Applications,* Vol. I-III, New York: Pergamon, 1966.
2. B. L. Moiseiwitsch, *Integral Equations,* New York: Longman Inc., 1977.
3. A. C. Pipkin, *A Course on Integral Equations,* New York: Springer-Verlag, 1991.
4. I. Stakgold, *Green's Functions and Boundary Value Problems,* New York: Wiley, 1979.
5. R. F. Harrington, *Field Computation by Moment Methods,* New York: Macmillan, 1968.
6. T. K. Sarkar, A note on the choice of weighting functions in the method of moments, *IEEE Trans. Antennas Propag.,* **33**: 436–441, 1985.
7. T. K. Sarkar, A. R. Djordjevic, and E. Arvas, On the choice of expansion and weighting functions in the method of moments, *IEEE Trans. Antennas Propag.,* **33**: 988–996, 1985.
8. A. R. Djordjevic and T. K. Sarkar, A theorem on the moment methods, *IEEE Trans. Antennas Propag.,* **35**: 353–355, 1987.
9. J. M. Bornholdt and L. N. Medgyesi-Mitschang, Mixed domain Galerkin expansions in scattering problems, *IEEE Trans. Antennas Propag.,* **36**: 216–227, 1988.
10. R. W. P. King, *The Theory of Linear Antennas,* Cambridge, MA: Harvard Univ. Press, 1956.
11. D. S. Jones, *Methods in Electromagnetic Wave Propagation,* Oxford: Clarendon, 1979.
12. D. R. Wilton, S. M. Rao, A. W. Glisson, D. H. Schaubert, O. M. Al-Bundak, and C. M. Butler, Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains, *IEEE Trans. Antennas Propag.,* **32**: 276–281, 1984.
13. A. W. Glisson and D. R. Wilton, Simple and efficient numerical methods for problems of electromagnetic radiation and scattering from surfaces, *IEEE Trans. Antennas Propag.,* **28**: 593–603, 1980.
14. S. M. Rao, A. W. Glisson, D. R. Wilton, and B. S. Vidula, A simple numerical solution procedure for statics problems involving arbitrary shaped surfaces, *IEEE Trans. Antennas Propag.,* **27**: 604–608, 1979.
15. S. M. Rao, D. R. Wilton, and A. W. Glisson, Electromagnetic scattering by surfaces of arbitrary shape, *IEEE Trans. Antennas Propag.,* **30**: 409–418, 1982.
16. S. M. Rao and P. K. Raju, Application of the method of moments to acoustic scattering from multiple bodies of arbitrary shape, *J. Acous. Soc. Amer.,* **86**: 1143–1148, 1989.
17. P. K. Raju, S. M. Rao, and S. P. Sun, Application of the method of moments to acoustic scattering from multiple infinitely long fluid filled cylinders, *Comp. Struct.,* **39**: 129–134, 1991.
18. S. M. Rao and B. S. Sridhara, Application of the method of moments to acoustic scattering from arbitrary shaped rigid bodies coated with lossless, shearless materials of arbitrary thickness, *J. Acous. Soc. Amer.,* **90**: 1601–1607, 1991.
19. S. M. Rao and B. S. Sridhara, Acoustic scattering from arbitrarily shaped multiple bodies in half space: Method of moments solution, *J. Acous. Soc. Amer.,* **91**: 652–657, 1992.
20. C. L. Bennett, *A Technique for Computing Approximate Electromagnetic Impulse Response of Conducting Bodies,* Ph.D. thesis, Purdue University, Lafayette, IN, 1968.
21. S. M. Rao, T. K. Sarkar, and S. A. Dianat, The application of the conjugate gradient method to the solution of transient electromagnetic scattering from thin wires, *Radio Sci.,* **19**: 1319–1326, 1984.
22. S. M. Rao, T. K. Sarkar, and S. A. Dianat, A novel technique to the solution of transient electromagnetic scattering from thin wires, *IEEE Trans. Antennas Propag.,* **34**: 630–634, 1986.
23. S. M. Rao and D. R. Wilton, Transient scattering by conducting surfaces of arbitrary shape, *IEEE Trans. Antennas Propag.,* **39**: 56–61, 1991.
24. D. A. Vechinski and S. M. Rao, Transient scattering from dielectric cylinders—E-field, H-field, and combined field solutions, *Radio Sci.,* **27**: 611–622, 1992.
25. D. A. Vechinski and S. M. Rao, Transient scattering from two-dimensional dielectric cylinders of arbitrary shape, *IEEE Trans. Antennas Propag.,* **40**: 1054–1060, 1992.
26. D. A. Vechinski and S. M. Rao, Transient scattering by conducting cylinders—TE case, *IEEE Trans. Antennas Propag.,* **40**: 1103–1106, 1992.
27. D. A. Vechinski and S. M. Rao, A stable procedure to calculate the transient scattering by conducting surfaces of arbitrary shape, *IEEE Trans. Antennas Propag.,* **40**: 661–665, 1992.
28. D. A. Vechinski, S. M. Rao, and T. K. Sarkar, Transient scattering from three-dimensional arbitrarily shaped dielectric bodies, *J. Optical Soc. Amer.,* **11**: 1458–1470, 1994.
29. E. K. Miller, L. Medgyesi-Mitschang, and E. H. Newman, *Computational Electromagnetics—Frequency-Domain Method of Moments,* New York: IEEE Press, 1992.
30. B. Z. Steinberg and Y. Leviatan, On the use of wavelet expansions in the method of moments, *IEEE Trans. Antennas Propag.,* **41**: 610–619, 1993.
31. R. Coifman, V. Rokhlin, and S. Wandzura, The fast multipole method: A pedestrian prescription, *IEEE AP-S Mag.,* **35**: 7–12, 1993.
32. F. X. Canning, Improved impedance matrix localization method, *IEEE Trans. Antennas Propag.,* **41**: 659–667, 1993.
33. M. Hestenes and E. Stiefel, Method of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards,* **49**: 409–436, 1952.

34. Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comp.*, **7**: 856–869, 1986.

S. M. RAO
G. K. GOTHARD
Auburn University

INTEGRAL EQUATIONS. See INTEGRO-DIFFERENTIAL EQUATIONS; WAVELET METHODS FOR SOLVING INTEGRAL AND DIFFERENTIAL EQUATIONS.

INTEGRAL TRANSFORMS. See HANKEL TRANSFORMS; LAPLACE TRANSFORMS.

INTEGRATED ACOUSTOOPTIC DEVICES. See ACOUSTOOPTICAL DEVICES.

INTEGRATED CIRCUIT MANUFACTURING DIAGNOSIS. See DIAGNOSIS OF SEMICONDUCTOR PROCESSES.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Integro-Differential Equations
Standard Article
Manuel Garcia[1]
[1]Lightning Publications, Oakland, CA
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2429
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (197K)

Abstract

The sections in this article are

A Sample Analysis

Linear Equations

Volterra Animals

Applications

| | |

Copyright © 1999-2008 All Rights Reserved.

INTEGRO-DIFFERENTIAL EQUATIONS

This article will focus on methods of solution. The aim is to show how a student or engineer can manipulate an integro-differential problem into a form that is simple to calculate. Few of these equations yield analytical solutions. The direct numerical approach of using finite differences for derivatives and sums for integrals relies on the capability of the computer and on the stability of the numerical algorithm. The methods described in this article aim to improve the stability of the eventual calculation by removing derivatives, and to minimize repetitive calculations (nested loops). These techniques make it possible to solve realistic problems with modest personal computers.

An integro-differential equation describes the influence of an accumulation of points upon the value and dynamics of each individual member of the collection. These equations are a balance between a quantity, its derivatives, and its integrals. The most significant applications of integro-differential equations are in modeling the impact of heredity and the dynamics of systems out of equilibrium. Heredity problems in engineering include analyzing fluid and heat flow, mechanical stress, and the accumulation of residual charge for materials with memory. The study of nonequilibrium systems is based on kinetic theory, where the properties of a gas are calculated as the average of individual molecular collisions. Integro-differential equations are applied in biology and economics as well as in physics and engineering.

A differential equation describes the dynamics of a quantity. It is a balance between the values of the quantity and its

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

various rates of change *at a given moment.* One such balance is between the acceleration of a particle and the action of external forces, classical Newtonian mechanics. Basic examples are the one-dimensional mass-spring-damper equation and its electrical analog, the resistor–inductor–capacitor (*RLC*) circuit equation. These specific differential models each conserve a global quantity; for the mechanical example it is momentum, for the electrical example it is current. The implicit assumption in the differential description is that the future state of a system does not depend on its history. Erosion, fatigue, wear, failure, experience, heredity, evolution, karma—all these words express some observation about the impact of past dynamics on future dynamics. One example is the failure of mechanical components subjected to repetitive stress. Engineers routinely calculate the amount of twist of a metal bar subjected to a specific torque. If we assume this phenomenon to be purely differential, then the same amount of torque will always produce the same amount of twist. In reality, metal subjected to repeated strain experiences fatigue. Eventually the application of the same torque produces a different twist, perhaps a catastrophic event. The metal inherits a degradation of its elasticity, an integral of the history of deflections. This effect caused the breakup of two de Havilland Comet jet airliners during flight in 1954. The engineers of the day were unaware that the aluminum fuselage would experience metal fatigue as a result of the frequent cycles of cabin pressurization.

An integral equation describes the influence of all points in a field upon the value of any particular point. Integral equations express an equilibrium. The points can be spatial for a field of stress in a surface or rod, or they can be instants of time in an orbital trajectory, or they can be individual molecules in a gas in which the field is a statistical distribution of molecular velocities. When external conditions change suddenly so that the system is out of balance, energy or information must flow within the system to rearrange it into a new equilibrium. Describing this nonequilibrium process requires differential terms in addition to the original integral equation. Instantaneous equilibration is often assumed in engineering applications, for instance in thermodynamics, and integro-differential equations are avoided. However, the sharpening of technology into much smaller space and time scales has required more exacting physical models that account for nonequilibrium dynamics. This technological trend drives the continuing interest in solving integro-differential equations.

A method is described for transforming integro-differential equations with linear derivatives into purely integral forms, which are then solved by iteration. Knowledge of an approximate solution speeds the convergence of iteration. One method of developing such approximate solutions is described in the section that follows. By the very nature of approximation, such methods depend on the specifics of the particular integro-differential equation. It is best to view the development of the approximate solutions in the sample analysis as an example of the attitude and reasoning that may prove useful in other problems. After the discussion of linear equations a nonlinear system with hereditary effects is described. This nonlinear system describes the conflict between populations of predators and prey. This system is reduced to a single nonlinear integral equation for which an iterated solution is found. However, a much easier calculation of the solution is possible with an approximate equation developed from the nonlinear integral. This approximation recognizes the effect of the hereditary integral and casts the problem as a type of recursion formula $\{x(t) = f[x(t - \Delta t), t]\}$, eliminating the need for any iteration. A discussion of physical applications of integro-differential equations concludes the article.

A SAMPLE ANALYSIS

Consider the following linear, first-order, homogeneous integro-differential equation for unknown function $y(x)$:

$$\frac{dy(x)}{dx} = -\int_{x-\alpha}^{x+\beta} \frac{p(\xi)}{a(x)}\, y(\xi)\, d\xi \tag{1}$$

We will use this equation to demonstrate how a solution may be attempted. Both α and β are positive constants. This particular equation has a separable kernel

$$K(\xi, x) = p(\xi)/a(x) \tag{2}$$

and we suppose that $a(x)$ is not zero in the domain of interest. Can Eq. (1) be cast as a purely differential, or purely integral equation? If so, it may be possible to transform it to a standard form and solve it by established techniques. In this section we will look first at the effect of differentiating Eq. (1), then we will seek an approximate solution directly from the integro-differential form, then we will transform Eq. (1) into a purely integral form, and finally we will show specific examples.

Differentiating Eq. (1) results in

$$a(x)y''(x) + a'(x)y'(x) = p(x - \alpha)y(x - \alpha) - p(x + \beta)y(x + \beta) \tag{3}$$

where primes refer to differentiation with respect to x. In Eq. (3) derivatives of y at x depend on values of y at positions to either side of x. This form is a differential-difference equation. References 1 and 2 describe this type of equation. From Eq. (3), $y(x - \alpha)$ can be cast as depending on itself at higher x

$$y(x - \alpha) = \frac{a(x)y''(x) + a'(x)y'(x) + p(x + \beta)y(x + \beta)}{p(x - \alpha)} \tag{4}$$

This form of the equation is the basis of a numerical solution in cases where it is known that $y(x)$ decays exponentially with respect to positive x. Above a given coordinate, say x_2, y is assumed to be small and its derivatives are assumed to be zero. A solution is constructed for $x < x_2$ using Eq. (4). In a range $x < x_1$, where $x_1 < x_2$, the function $y(x)$ is exponentially larger than the starting value assumed, and $y(x)$ is considered an accurate solution. Care must be taken in the numerical treatment of the derivatives, and this is most directly accomplished by using closely spaced points and higher-order differences. If the kernel is not separable, then differentiation will not remove the integral. It will now contain the derivative of the kernel with respect to x, $K'(\xi, x)$.

Let us assume that $y(x)$ is a positive function that decays exponentially with respect to positive x. In this case both $a(x)$ and $p(x)$ are positive over the range of interest, $x_0 \leq x <$

∞. Let us seek a solution in the form

$$y(x) = \exp\left[-\int_{x_0}^{x} B(\eta)\,d\eta\right] \tag{5}$$

where x_0 is a reference coordinate where $y = 1$, an initial condition. Notice that $-y'/y = B$. Divide Eq. (1) by $y(x)$ and use Eq. (5),

$$a(x)B(x) = \int_{x-\alpha}^{x+\beta} p(\xi)\exp\left[-\int_{x}^{\xi} B(\eta)\,d\eta\right]d\xi \tag{6}$$

Notice that in Eq. (6) it is the ratio $y(\xi)/y(x)$ that appears in the integral with $p(\xi)$, and this ratio is given by the exponential involving $B(\eta)$. Given an estimate of function B, call it B_0, Eq. (6) can be used to find a possibly more accurate estimate B_1 by the method of successive approximations. This method, also known as the method of Picard, uses a prior iterant within the integral (B_0) to find a next iterant (B_1) from the equation. References 3 and 4 describe the validity and use of this method.

If the method of successive approximations converges to a solution, then the exact nature of the initial iterant B_0 is unimportant. However, the more accurately B_0 portrays the actual solution $B(x)$, the fewer iterants need to be calculated. We now seek an initial iterant from Eq. (6) by making whatever assumptions simplify this problem, while at the same time being mindful to avoid a trivial result by being too hasty. For the moment we will assume that $p(x)$ is weakly dependent on x within any band $(x - \alpha, x + \beta)$, and that $B(\xi)$ remains of the same order of magnitude for $(x - \alpha \le \xi \le x + \beta)$. The following approximations cascade from Eq. (6) by using these assumptions:

$$\begin{aligned} a(x)B_0(x) &\approx \int_{x-\alpha}^{x+\beta} p(\xi)e^{-B_0(x)(\xi-x)}\,d\xi \approx p(x)\int_{x-\alpha}^{x+\beta} e^{-B_0(x)(\xi-x)}\,d\xi \\ &\approx \frac{p(x)}{-B_0(x)}\int_{x-\alpha}^{x+\beta} e^{-B_0(x)(\xi-x)}[-B_0(x)]\,d\xi \qquad (7) \\ &\approx \frac{p(x)}{B_0(x)}[e^{\alpha B_0(x)} - e^{-\beta B_0(x)}] \end{aligned}$$

$$B_0(x) = \sqrt{\frac{p(x)}{a(x)}\,e^{\alpha B_0(x)}[1 - e^{-(\alpha+\beta)B_0(x)}]} \tag{8}$$

For a small $B_0(x)$ such that both $B_0(x)(\alpha + \beta) < 1$ and $B_0(x)\alpha < 1$, then

$$B_0(x) = \frac{p(x)}{a(x)}(\alpha + \beta) \tag{8a}$$

which is found by expanding the exponentials in Eq. (8). Notice that to be consistent, $p(x)/a(x)$ must be less than $(\alpha + \beta)^{-2}$. For $B_0(x)$ such that $B_0(x)(\alpha + \beta) > 1$ while $B_0(x)\alpha < 1$, which implies $\beta \gg \alpha$, then

$$B_0(x) = \sqrt{\frac{p(x)}{a(x)}} \tag{8b}$$

This case is consistent with $1/(\alpha + \beta) < \sqrt{p(x)/a(x)} < 1/\alpha$. Finally, for $B_0(x)$ such that both $B_0(x)(\alpha + \beta) > 1$ and $B_0(x)\alpha > 1$, then $B_0(x)$ is the root of a transcendental equation,

$$B_0(x) = \sqrt{\frac{p(x)}{a(x)}\,e^{\alpha B_0(x)}} \tag{8c}$$

If none of Eqs. (8a)–(8c) are applicable, then $B_0(x)$ must be found as a root of Eq. (8). The corresponding initial iterant $y_0(x)$ for Eq. (1) is given by using the appropriate result from Eqs. (8) or (8a)–(8c) in definition (5). The example shown below uses the simplest case, Eq. (8a),

$$y_0(x) = \exp\left[-(\alpha + \beta)\int_{x_0}^{x} \frac{p(\eta)}{a(\eta)}\,d\eta\right] \tag{9}$$

It is very important to capture the functional nature of $p(x)$ within the integral of Eq. (7). In the preceding, $p(x)$ was assumed to be very weakly dependent on x over the interval $(x - \alpha, x + \beta)$ in a manner similar to a constant or $\log(x)$. If instead, $p(x) = p_0(x)x$, where $p_0(x)$ is a weak function of x as used here, then the results in place of Eq. (8) are as follows:

$$B_0(x) = \sqrt[\frac{1}{3}]{\frac{p_0(x)}{a(x)}\,e^{\alpha B_0(x)}\{1 + B_0(x)(x-\alpha) - [1 + B_0(x)(x+\beta)]e^{-(\alpha+\beta)B_0(x)}\}} \tag{10}$$

The case of small $B_0(x)$ corresponding to Eq. (8a) is now

$$B_0(x) = \frac{p_0(x)}{a(x)}(\alpha + \beta)\left(x + \frac{\beta^2 - \alpha^2}{(\alpha + \beta)}\right) \tag{10a}$$

Note the additional linear factor in comparison to Eq. (8a). It is essential to retain that factor of $p(x)$ with significant variation within the integral of Eq. (7). We will only use the simplest B_0 and y_0, derived as Eqs. (8a) and (9), respectively, to illustrate a first iterant with Eq. (11):

$$a(x)B_1(x) = \int_{x-\alpha}^{x+\beta} p(\xi)\exp\left[-(\alpha + \beta)\int_{x}^{\xi} \frac{p(\eta)}{a(\eta)}\,d\eta\right]d\xi \tag{11}$$

and a $y_1(x)$ can be constructed from the $B_1(x)$ of Eq. (11). A $y_1(x)$ can also be written explicitly from the integral of Eq. (1) by using $y_0(x)$

$$y_1(x) = y(x_0) - \int_{x_0}^{x}\int_{\xi-\alpha}^{\xi+\beta} \frac{p(\eta)}{a(\xi)}y_0(\eta)\,d\eta\,d\xi \tag{12}$$

Recall that $y(x_0) = 1$ in this particular case. Whether the first iterant sought is B_1 from Eq. (11) or y_1 from Eq. (12), a double integration is required after the zeroth iterants B_0 and y_0 are calculated. It would be very discouraging to do all this work and then find that our iteration was diverging. An effort to reduce repetitive integration follows.

Equation (12) is a purely integral form of Eq. (1) when the subscripts on y are removed and $y(x_0)$ is arbitrary. By reversing the order of integration it is possible to reformulate this equation as a single integration over the unknown $y(\eta)$

with a new kernel

$$y(x) - y(x_0) = -\int_{x_0}^{x} \int_{\xi-\alpha}^{\xi+\beta} \frac{p(\eta)}{a(\xi)} y(\eta)\, d\eta\, d\xi = -\int_{x_0-\alpha}^{x+\beta} p(\eta) M(\eta, x) y(\eta)\, d\eta \tag{13}$$

The new kernel factor $M(\eta, x)$ is given below for this example. The method of reversing the order of integration and generating kernels of this type will be described in the section titled Linear Equations:

$$\begin{aligned} M(\eta, x) = \Bigg\{ & \int_{x_0}^{\eta+\alpha} \frac{d\xi}{a(\xi)}; [x_0 - \alpha] \le \eta \\ & < [\min(x_0 + \beta, \min(x_0 + \alpha + \beta, x) - \alpha)] \Bigg\} \\ & + \left\{ \int_{x_0}^{x} \frac{d\xi}{a(\xi)}; \min[x_0 + \beta, \min(x_0 + \alpha + \beta, x) - \alpha] \right. \\ & \left. \le \eta < [x_0 + \beta] \right\} \\ & + \left\{ \int_{\eta-\beta}^{\min(\eta+\alpha, x)} \frac{d\xi}{a(\xi)}; [x_0 + \beta] \le \eta \le [x + \beta] \right\} \end{aligned} \tag{14}$$

The function min(a, b, . . .), used in $M(\eta\, x)$, selects the minimum of its arguments. $M(\eta\, x)$ is the sum of three terms, each defined over a different range of η, and these ranges are functions of x. The new kernel $K(\eta, x) = p(\eta)M(\eta, x)$ can be calculated once from known functions $a(x)$ and $p(x)$, and by the explicit operations of Eq. (14). The derivation of Eqs. (13) and (14) proceeds directly from Eq. (1), without requiring any specialized assumptions, as were used in the development of B_0. Now the original integro-differential equation has been transformed into a purely integral form, a Volterra equation (variable upper limit) of the second kind [inhomogeneous if $y(x_0) \neq 0$]. The method of successive approximations applied to Eq. (13) proceeds more quickly because each iterant of $y(x)$ is now the result of a single integration.

Two specific numerical examples follow. In both cases $a(x) = a_0 x$, and $p(x) = p_0 x$, where a_0 and p_0 are constants. Solutions are sought in the range $[(x_0 = 1) \le x \le (x_1 = 3)]$, though calculations must consider the wider range $(1 - \alpha, 3 + \beta)$. In these cases $y(1) = 1$. $B_0(x)$ is found as the root of Eq. (10), and a $y_0(x)$ is calculated from Eq. (5). The kernel $K(\eta, x) = p(\eta)M(\eta, x)$ is calculated on the basis of Eq. (14). Two iterants, y_1 and y_2, are then found by the method of successive approximations from Eq. (13). Figure 1 shows y_0, y_1, and y_2 for $a_0 = 1$, $p_0 = 8$, $\alpha = 0$, and $\beta = 1.45$. Iterants y_0 and y_1 are quite smooth; with y_2 the point-to-point numerical noise becomes noticeable (point locations are shown for y_2). This noise diminishes as more closely spaced points are used. In this case $y(x)$ has a rapid exponential decay. A second case has $a_0 = 1$, $p_0 = 0.08$, $\alpha = 0.55$, and $\beta = 1.45$. Figure 2 shows the three iterants of y, which decay gently with x. In both cases y_0 and y_1 bracket y_2. Figures 3 and 4 show the kernel $K(\eta, x)$ for the first case (both appear similar). Two views are given to help visualize this surface over the full range of the calculation.

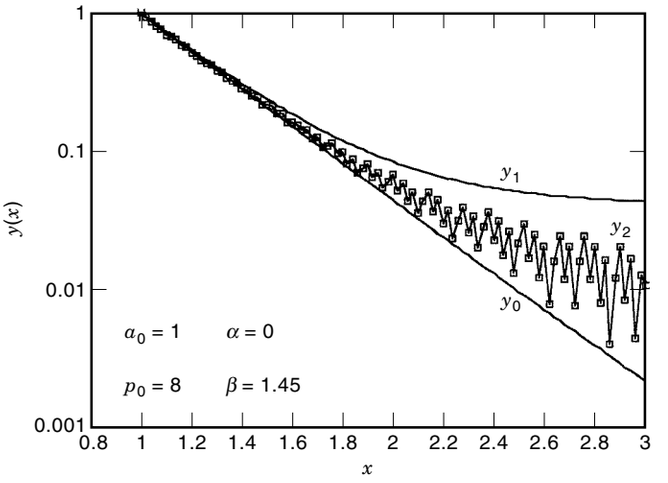


Figure 1. Three iterants for the $y(x)$ of Eq. (1) when $a(x) = x$, $p(x) = 8x$, $\alpha = 0$, and $\beta = 1.45$. The zeroth iterant $y_0(x)$ is found by an approximation to its logarithmic derivative $B_0 = -\{d\ln[y_0(x)]/dx\}$ that is given by Eq. (10). The iteration is applied to Eq. (13), which is a single integral form of Eq. (1) with a new kernel $p(\eta)M(\eta, x)$ that is described by Eq. (14). Convergence is rapid. This function decays by two orders of magnitude for $1 \le x \le 3$. The relative error is comparable to $y(x)$ at low amplitude. This error diminishes as more points are used (point locations shown for y_2).

LINEAR EQUATIONS

Casting a linear, first-order integro-differential equation into a simple integral form is very useful because then it can be solved by the method of successive approximations. This transformation involves switching the order of integration of a double integral, an operation mentioned without explanation in the section titled A Sample Analysis. This transforma-

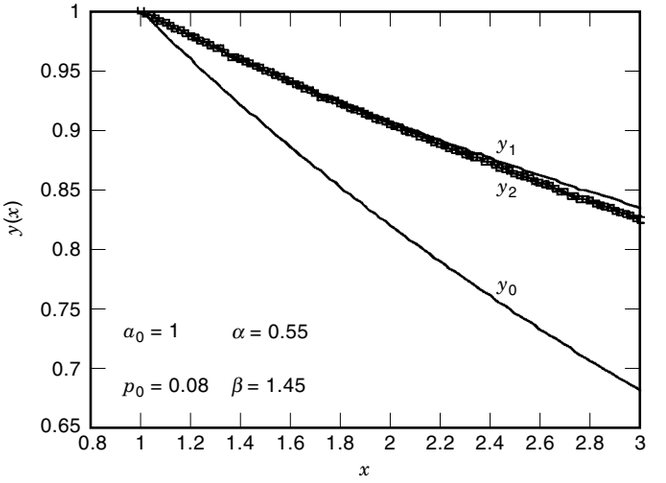


Figure 2. Three iterants for the $y(x)$ of Eq. (1) when $a(x) = x$, $p(x) = 0.08x$, $\alpha = 0.55$, and $\beta = 1.45$. Another case similar to that of Fig. 1. Here the function decays very gently, and the relative error is small.

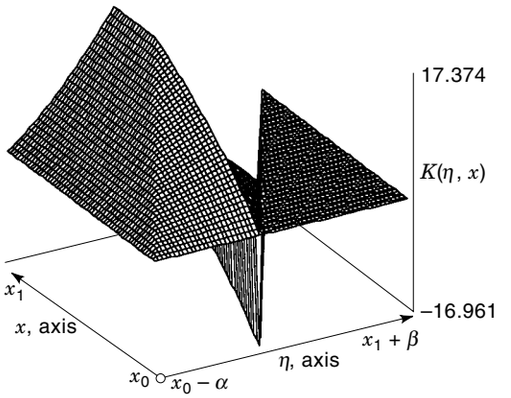


Figure 3. A surface plot of $K(\eta, x) = p(\eta)M(\eta, x)$, the kernel used in the example of Fig. 1. The kernel for the second example has the same shape but is of different magnitude. This view extends over the full area of the calculation in the (η, x) plane. The problem has been converted to an integral equation with single integration and a known kernel.

tion will be illustrated for the following equation:

$$\frac{dy(x)}{dx} + b(x)y(x) + c(x) = \int_{x-\alpha}^{x+\beta} K_1(\xi, x)y(\xi)\,d\xi + \int_{x_0}^{x} K_2(\xi, x)y(\xi)\,d\xi \tag{15}$$

We assume that over the domain of interest, $x_0 \le x \le x_1$, none of b, c, K_1 and K_2 become infinite. Also, α and β are positive constants. The labels $V_1(x)$ and $V_2(x)$ will be used to represent the integrals over K_1 and K_2, respectively. Now Eq. (15) is seen as a linear, first-order differential equation with an inhomogeneous term $V_1(x) + V_2(x) - c(x)$. This is formally integrated to

$$y(x) = e^{-\int_{x_0}^{x} b(\gamma)\,d\gamma} \left\{ y(x_0) + \int_{x_0}^{x} [V_1(\xi) + V_2(\xi) - c(\xi)]e^{\int_{x_0}^{\xi} b(\gamma)\,d\gamma}\,d\xi \right\} \tag{16}$$

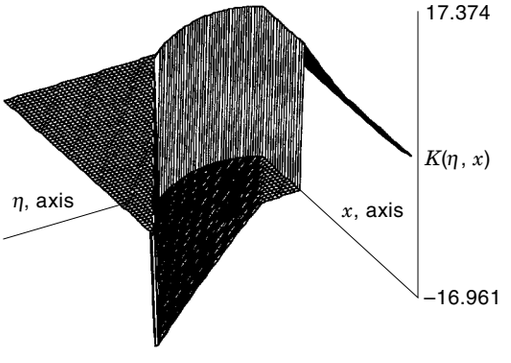


Figure 4. A surface plot of $K(\eta, x) = p(\eta)M(\eta, x)$ seen from a different orientation. This view shows features of the surface that are hidden in Fig. 3. This example shows that integral equations can have smooth solutions even with discontinuous kernels.

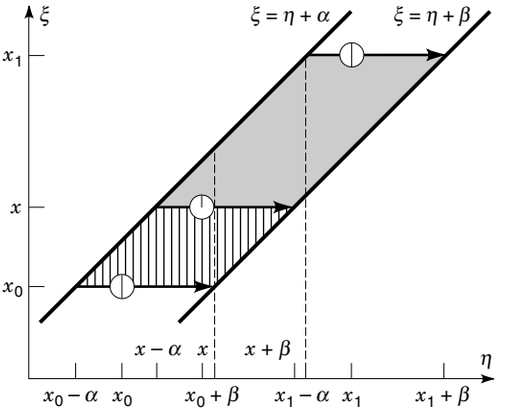


Figure 5. The area where double integration with the K_1 kernel is reversed. Horizontal arrows show the original direction of integration across the area. Reversed integration is shown by vertical hatching. As x moves up the ξ axis from x_0 to x_1, the horizontal arrow from $\eta = x - \alpha$ to $\eta = x + \beta$ moves vertically through the integration area. Reversed integration is done below this rising arrow. Here vertical integration proceeds in sections as η moves from limits $x_0 - \alpha$ to $x_1 + \beta$ (the new outer integral). ξ is integrated successively from: x_0 to the $\eta + \alpha$ boundary line, x_0 to x, the $\eta - \beta$ boundary line to the $\eta + \alpha$ boundary line, and the $\eta - \beta$ boundary line to x (the new inner integral). The limits are conditional statements because the transitions between vertical sections depend on the slant and width of the area.

$$y(x) = y(x_0)e^{-\int_{x_0}^{x} b(\gamma)\,d\gamma} - \int_{x_0}^{x} c(\xi)e^{-\int_{\xi}^{x} b(\gamma)\,d\gamma}\,d\xi + \int_{x_0}^{x}\int_{\xi-\alpha}^{\xi+\beta} e^{-\int_{\xi}^{x} b(\gamma)\,d\gamma} K_1(\eta, \xi)y(\eta)\,d\eta\,d\xi + \int_{x_0}^{x}\int_{x_0}^{\xi} e^{-\int_{\xi}^{x} b(\gamma)\,d\gamma} K_2(\eta, \xi)y(\eta)\,d\eta\,d\xi \tag{17}$$

The order of double integration will now be reversed. This is done to achieve single integral forms $\int M(\eta, x)y(\eta)d\eta$ with kernels $M(\eta, x)$ that are integrals of known functions. The K_1 and K_2 integrations of Eq. (17) occur over specific areas of the (η, ξ) plane determined by the limits. Figure 5 is a schematic of the area of integration for K_1. Figure 6 is a similar sche-

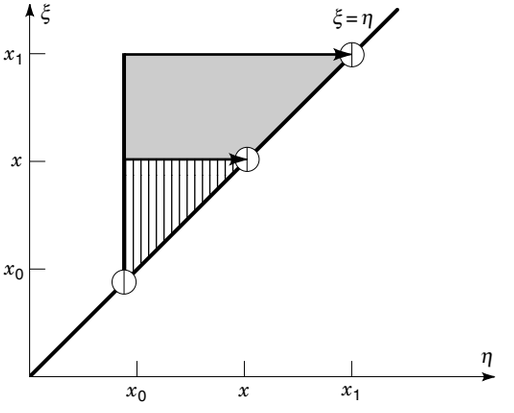


Figure 6. The area where double integration with the K_2 kernel is reversed. The original integration of $x_0 \le \xi \le x$ and $x_0 \le \eta \le \xi$ is reversed to $x_0 \le \eta \le x$ and $\eta \le \xi \le x$.

matic for K_2. In Figs. 5 and 6 these integrations would be visualized as progressing horizontally through the respective areas (see arrows). To reverse the order of integration is to progress vertically through the integral areas (see vertical hatching). The original double integrals could each become sums of several "reversed" terms. Each of the new, reversed double integrals would account for a portion of the original (η, ξ) area. The limits of the reversed integrals could be conditional statements that depend on the shape of the area boundary. The result here is shown as Eqs. (18) through (22):

$$y(x) = y(x_0)e^{-\int_{x_0}^{x} b(\gamma)\,d\gamma} - \int_{x_0}^{x} c(\xi)e^{-\int_{\xi}^{x} b(\gamma)\,d\gamma}\,d\xi + I_1(x) + I_2(x) \tag{18}$$

$$I_1(x) = \int_{x_0-\alpha}^{\min[x^*-\alpha,\,x_0+\beta]} \int_{x_0}^{\eta+\alpha} \ldots d\xi\,d\eta + \int_{\min[x^*-\alpha,\,x_0+\beta]}^{x_0+\beta} \int_{x_0}^{x} \ldots d\xi\,d\eta + \int_{x_0+\beta}^{x+\beta} \int_{\eta-\beta}^{\min[\eta+\alpha,\,x]} \ldots d\xi\,d\eta \tag{19}$$

where the integrands are

$$e^{-\int_{\xi}^{x^*} b(\gamma)\,d\gamma} K_1(\eta, \xi) y(\eta)$$

for the first term of I_1, and

$$e^{-\int_{\xi}^{x} b(\gamma)\,d\gamma} K_1(\eta, \xi) y(\eta) \tag{20}$$

for the last two terms of I_1. The function x^* is defined as

$$x^* = \min[x, (x_0 + \alpha + \beta)] \tag{21}$$

Finally, for I_2,

$$I_2(x) = \int_{x_0}^{x} y(\eta) \left\{ \int_{\eta}^{x} e^{-\int_{\xi}^{x} b(\gamma)\,d\gamma} K_2(\eta, \xi)\,d\xi \right\} d\eta \tag{22}$$

The original equation is now in a purely integral form with single integrations. New kernels, $M_1(\eta, x)$ (three terms) and $M_2(\eta, x)$, are defined as integrals of the products of an integrating factor and original kernels $K_1(\eta, \xi)$ and $K_2(\eta, \xi)$, respectively. The sample analysis describes solving a particular equation from this point.

Linear integro-differential equations with second-order derivatives can be transformed into a Volterra form in a manner similar to first-order equations. Consider the following second-order equation with the same integrals V_1 and V_2 as in Eq. (16):

$$\frac{d^2y(x)}{dx^2} + b(x)\frac{dy(x)}{dx} + c(x)y(x) + d(x) = V_1(x) + V_2(x) \tag{23}$$

Define the function $p(x) = dy(x)/dx$. Now Eq. (23) becomes a linear, first-order equation for $p(x)$ with inhomogeneous term $V_1(x) + V_2(x) - d(x) - c(x)y(x)$. This is integrated once for $p(x)$ by using an integrating factor $\exp[\int b(x)dx]$, and specifying an initial condition $p(x_0)$. The result for $p(x)$ is integrated from x_0 to x yielding $y(x) - y(x_0)$

$$\begin{aligned} y(x) = {} & y(x_0) + p(x_0)\int_{x_0}^{x} e^{-\int_{x_0}^{x} b(\gamma)\,d\gamma}\,d\xi \\ & - \int_{x_0}^{x}\int_{x_0}^{\xi} d(\eta)e^{-\int_{\eta}^{\xi} b(\gamma)\,d\gamma}\,d\eta\,d\xi \\ & + \int_{x_0}^{x}\int_{x_0}^{\xi} [V_1(\eta) + V_2(\eta) - c(\eta)y(\eta)]e^{-\int_{\eta}^{\xi} b(\gamma)\,d\gamma}\,d\eta\,d\xi \end{aligned} \tag{24}$$

The first three terms after the equal sign in Eq. (24) are all known, the fourth term contains $y(x)$ within double and triple integrals. Let $f(x, x_0)$ represent the sum of the three known terms in Eq. (24) and $H(x, \eta)$ represent the integral factor

$$H(x, \eta) = \int_{\eta}^{x} e^{-\int_{\eta}^{\xi} b(\gamma)\,d\gamma}\,d\xi \tag{25}$$

Using these definitions, and Eq. (21) for x^*, the form of Eq. (24) with only single integrals is

$$\begin{aligned} y(x) = {} & f(x, x_0) + \int_{x_0-\alpha}^{\min[(x^*-\alpha),\,(x_0+\beta)]} y(\zeta) \left\{ \int_{x_0}^{\zeta+\alpha} H(x^*, \eta)K_1(\zeta, \eta)\,d\eta \right\} d\zeta \\ & + \int_{\min[(x^*-\alpha),\,(x_0+\beta)]}^{x_0+\beta} y(\zeta) \left\{ \int_{x_0}^{x} H(x, \eta)K_1(\zeta, \eta)\,d\eta \right\} d\zeta \\ & + \int_{x_0+\beta}^{x+\beta} y(\zeta) \left\{ \int_{\zeta-\beta}^{\min[(\zeta+\alpha),\,x]} H(x, \eta)K_1(\zeta, \eta)\,d\eta \right\} d\zeta \\ & + \int_{x_0}^{x} y(\zeta) \left\{ \int_{\zeta}^{x} H(x, \eta)K_2(\zeta, \eta)\,d\eta \right\} d\zeta \\ & - \int_{x_0}^{x} y(\zeta)c(\zeta)H(x, \zeta)\,d\zeta \end{aligned} \tag{26}$$

The five integrals shown for Eq. (26) can be combined into a single integration from $(x_0 - \alpha)$ to $(x + \beta)$ by multiplying each kernel with a difference of Heaviside unit step functions to define limits. This was done to calculate examples from Eq. (14) in the sample analysis.

The linear equations discussed in the preceding have all described initial value problems. If boundary values are placed on $y(x)$ or its first derivative at two points (x_0, x_1), then the solution of a second-order equation is based on the characteristic functions, or eigenfunctions, of the differential part of the equation. A solution of the form $y(x) = C_0y_0(x) + C_1y_1(x) + C_2y_2(x) + \ldots$ is assumed, where the y_i are eigenfunctions corresponding to the eigenvalues λ_i. The coefficients C_i are found in exactly the same way as in boundary value problems involving nonhomogeneous differential equations. The three steps to the solution are: substitute the eigenfunction expansion into Eq. (23), multiply by a particular eigenfunction y_i to solve for its coefficient C_i, and integrate over the interval (x_0, x_1). In purely differential problems the result is a series of equations, one for each of the coefficients C_i. For integro-differential equations the result is a series of equations linking each coefficient to a weighted sum of coefficients, $C_i = \Sigma w_n C_n$. The weights w_n result from integration

and may be difficult to calculate. This matrix relationship among the coefficients reflects the nature of the original equation. The magnitude C_i of each mode $y_i(x)$ is linked to the magnitudes of the other modes in solution $y(x)$ by the integrals involving K_1 and K_2.

VOLTERRA ANIMALS

Volterra introduced the following system of coupled, nonlinear, first-order, integro-differential equations to describe the dynamics of survival for a population of predators $y(t)$ and a population of prey $x(t)$:

$$\begin{aligned}\left[\frac{1}{x(t)}\right]\frac{dx(t)}{dt} &= a(t) - b(t)y(t) - \int_c^t K_y(t,s)y(s)\,ds \\ \left[\frac{1}{y(t)}\right]\frac{dy(t)}{dt} &= -\alpha(t) + \beta(t)\,x(t) + \int_c^t K_x(t,s)\,x(s)\,ds\end{aligned} \tag{27}$$

These equations show rates of population growth that are dependent on three factors: herd size or predator density, encounters between species, and hereditary influences. Prey $x(t)$ are adversely affected by encounters with predators, $-b(t)x(t)y(t)$, and by evolutionary improvements in these predators, $-x(t)\int K_y(t,\ s)y(s)\ ds$. Predators are adversely affected by too high a population of their own kind, $-\alpha(t)y(t)$. Reference 4 discusses this system in detail. The hereditary integral is described for heredity coefficients (K_x and K_y) of the form $K(t - s)$ under various names: it is the "renewal equation" in Ref. 2, "convolution" in Ref. 3, the "superposition integral" in Ref. 5, and an integral with a "displacement kernel" in Ref. 6. We will describe an approximate method of solution for Eq. (27) that makes few assumptions about the coefficients a, b, α, and β, or the kernels K_x and K_y.

Figure 7 is a particular example of Eqs. (27) for $0 \le t \le 20$, $c = 0$ ("the creation"), $a = b = 2$, $\alpha = \beta = 1$, $K_x = K_y = -0.05$, and initial conditions $x(0) = 1$ and $y(0) = 2$. Figure 8 is a phase diagram for this case where the initial conditions and the direction of time's arrow are indicated. Without hereditary influences ($K_x = K_y = 0$), the nonlinear, purely differential system has a periodic trajectory that is a noncircular closed path on the xy phase plane (a "vortex cycle"). In general, neither $x(t)$ nor $y(t)$ can be expressed in terms of elementary functions. The effect of the hereditary integrals is to cause a "drift" in the solutions, seen as a rising trend for this example. Additional long-term effects for this case are a diminishing impact of predators (y) on prey (x) and a shortening of the time between cycles. Converging or diverging populations that either grow or diminish can clearly be simulated by changing the magnitudes and the signs of the coefficients and kernels. More interesting effects arise when these factors are time dependent.

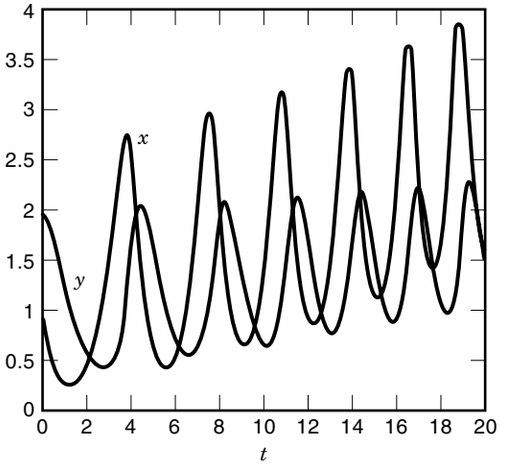


Figure 7. Population histories of predators $y(t)$ and prey $x(t)$ from the Volterra model of Eq. (27) with $a = b = 2$, $\alpha = \beta = 1$, $K_x = K_y = -0.05$, $c = 0$, $y(c) = 2$, and $x(c) = 1$. In this example heredity causes the populations to increase, diverge, and cycle more often.

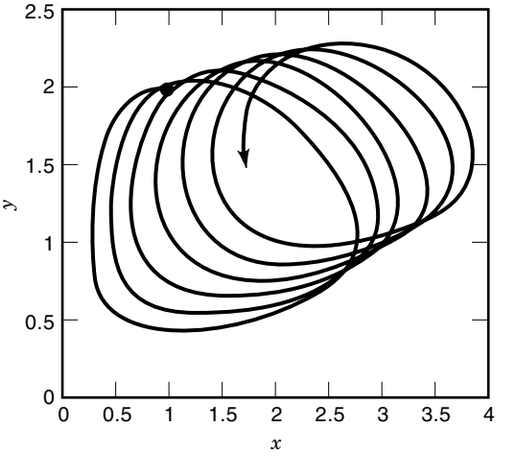


Figure 8. Phase diagram for the Volterra model example of Fig. 7. If heredity coefficients K_x and K_y are zero, then this curve is a closed noncircular path called a vortex cycle. The initial point and the direction of time's arrow are shown. Heredity causes a drift in the cyclic action.

Equations (27) are integrated once

$$\begin{aligned}\ln\left[\frac{x(t)}{x_c}\right] &= \int_c^t a(s)\,ds \\ &\quad - \int_c^t b(s)y(s)\,ds - \int_c^t \int_c^s K_y(s,u)y(u)\,du\,ds \\ \ln\left[\frac{y(t)}{y_c}\right] &= -\int_c^t \alpha(s)\,ds + \int_c^t \beta(s)x(s)\,ds \\ &\quad + \int_c^t \int_c^s K_x(s,u)x(u)\,du\,ds\end{aligned} \tag{28}$$

The order of double integration is reversed, and then the following functions are defined:

$$\begin{aligned}A(t) &= \int_c^t a(s)\,ds \\ \Lambda(t) &= \int_c^t \alpha(s)\,ds \\ M_x(t,u) &= \int_u^t K_x(s,u)\,ds \\ M_y(t,u) &= \int_u^t K_y(s,u)\,ds\end{aligned} \tag{29}$$

Now the equations are

$$\ln\left[\frac{x(t)}{x_c}\right] = A(t) - \int_c^t [b(u) + M_y(t,u)]y(u)\,du$$
$$\ln\left[\frac{y(t)}{y_c}\right] = -\Lambda(t) + \int_c^t [\beta(u) + M_x(t,u)]x(u)\,du \quad (30)$$

The equation for $y(t)$ is substituted into the equation for $x(t)$, yielding

$$\ln\left[\frac{x(t)}{x_c}\right] = A(t) - y_c \int_c^t [b(u) + M_y(t,u)]e^{-\Lambda(u)+\int_c^u [\beta(w)+M_x(u,w)]x(w)\,dw}\,du \quad (31)$$

This nonlinear equation for $x(t)$ would appear to be an excellent form on which to apply the method of successive approximations. Figure 9 is a display of twenty-three successive approximations to Eq. (31) for the specific example described in Figs. 7 and 8. Forty-one points are used in this calculation, and the range is restricted to $0 \le t \le 10$. The zeroth iterant is $x_c = 1$ for the entire range, and calculated values of $x(t)$ larger than $6x_c$ are reset to x_c. The solution is seen to chip its way into the unknown like a pickax repeatedly driven into concrete. This is because the derivative of the solution at its leading edge depends on the integral of its history, so each iterant only advances the solution a small amount in time. It would be more efficient to calculate a solution by advancing forward in time rather than iterating over the entire time domain. The calculation of $x(t)$ requires iteration because $x(t)$ appears on both sides of Eq. (31). If $x(t)$ could be shown to depend only on its history, and not also on its present value, then the solution would be a recursion formula and calculations would be speedier. A solution of this type is achieved by assuming that $x(t) \approx x(t - \Delta t)$ for Δt sufficiently small. The integral in Eq. (31) is split into two terms, the first with limits $c \le u \le t - \Delta t$, and the second with limits $t - \Delta t \le u \le t$. The second integral is now approximated by a two-point trapezoid rule [2 × integral/Δt = integrand(t) + integrand($t - \Delta t$)]. The trapezoid rule integrand at t has the form

$$L_y(t,t)e^{-\Lambda(t)+\int_c^t L_x(t,w)x(w)\,dw} \quad (32)$$

where $L_y(t, u) = b(u) + M_y(t, u)$ and $L_x(t, u) = \beta(u) + M_x(t, u)$. The two-point trapezoid approximation is used again for the integral in Eq. (32), which now has the form

$$\{L_y(t,t)e^{-\Lambda(t)+\int_c^{t-\Delta t} L_x(t,w)x(w)\,dw}\}e^{\int_{t-\Delta t}^t L_x(t,w)x(w)\,dw}$$
$$\approx \{\ldots\}e^{(\Delta t/2)[L_x(t,t)x(t)+L_x(t,t-\Delta t)x(t-\Delta t)]}$$
$$\approx \{\ldots\}e^{(\Delta t/2)[L_x(t,t)+L_x(t,t-\Delta t)]x(t-\Delta t)} \quad (33)$$

The resulting approximation in place of Eq. (31) is

$$\ln\left[\frac{x(t)}{x_c}\right] = A(t) - y_c \int_c^{t-\Delta t} L_y(t,u)e^{-\Lambda(u)+\int_c^u L_x(u,w)x(w)\,dw}\,du$$
$$- \frac{y_c \Delta t}{2} L_y(t, t-\Delta t)e^{-\Lambda(t-\Delta t)+\int_c^{t-\Delta t} L_x(t-\Delta t,w)x(w)\,dw}$$
$$- \left\{\frac{y_c \Delta t}{2} L_y(t,t)e^{-\Lambda(t)+\int_c^{t-\Delta t} L_x(t,w)x(w)\,dw}\right\}$$
$$e^{(\Delta t/2)[L_x(t,t)+L_x(t,t-\Delta t)]x(t-\Delta t)} \quad (34)$$

Notice that for $\Delta t = 0$ we recover the original equation. Here the calculation is an explicit operation moving forward in time. This result makes it much easier to calculate the example shown in Figs. 7 and 8 than by iteration (161 points span the range $0 \le t \le 20$).

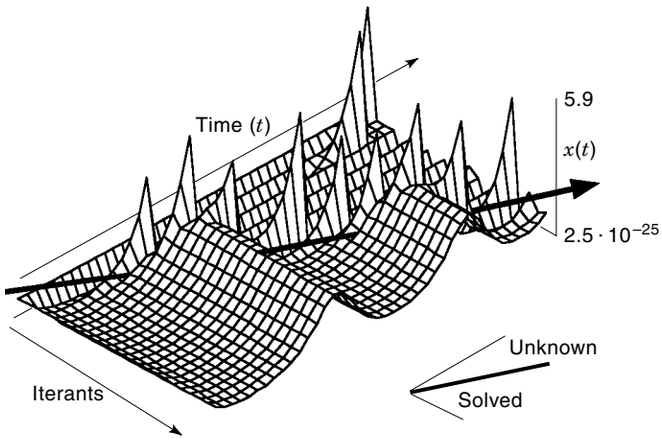


Figure 9. A sequence of successive approximations to Eq. (31) for the case shown in Figs. 7 and 8. Equation (31) is a nonlinear integral form of the Volterra predator and prey model. Each iterant builds up the solution sequentially, even though iteration occurs over the full time interval ($0 \le t \le 10$ in these calculations of 23 iterants). This is because the derivative of the solution at the present moment depends on the integral of its history. The iteration had an upper limit of six times the initial condition; any point calculated above this limit was reset to the initial condition. A better method of calculation is based only on prior events.

APPLICATIONS

Equation (1) in the sample analysis section is a form of the Boltzmann equation for the drift of a cloud of electrons along a constant electric field through a uniform molecular gas. Physical quantities are as follows: x is electron kinetic energy in units of eV, $y(x)$ is the distribution function of electron kinetic energy in units of $eV^{-3/2}$, $a(x) = a_0 x = (1/3)(E/N)^2(x/Q)$, E is the electric field in V/cm, N is the particle density of the gas in cm^{-3}, Q is the electron-molecule elastic collision cross section in cm^2, $p(x) = p_0 x = Sx$, S is the electron–molecule inelastic collision cross section in cm^2, $\alpha = 0$, $B_0(x)$ is an approximation for the logarithmic derivative of $y(x)$, β is large so $\beta B_0(x) > 1$, and $xB_0(x) > 1$ (this model of electron kinetics is for energies x above the range of thermal motion, $x \gg 0.03$ eV). Both Q and S are assumed to be only mildly dependent on x. $B_0(x)$ is given by Eq. (10) and then $y_0(x)$ is

$$y_0(x) = \exp\left\{-\int_0^x \frac{\sqrt{3Q(\xi)S(\xi)}}{E/N}\,d\xi\right\} \quad (35)$$

Typical parameters in experiments might be $Q = 10^{-15}$, $S = 3 \times 10^{-16}$, $E = 1000$, and $N = 10^{18}$. Good approximations for distribution functions of electron energy in nitrogen mixtures have been calculated from this result by using cross section

data. Literature on the Boltzmann equation is vast. The pervasive approximation is that the system is never far from thermal equilibrium $f_0(x)$ ("Maxwellian" distribution), so that the nonequilibrium solution $f(x)$ is a perturbation given by an expansion $f(x) = f_0(x)(1 + \phi_1(x) + \phi_2(x) + \ldots)$, where succeeding terms are of smaller magnitude. The full development of this Chapman–Enskog method is quite involved (see Ref. 7). The electron energy distribution may be far from thermal equilibrium with the gas molecules in an electric discharge because of the high electric fields. A Chapman–Enskog expansion for electrons might require the calculation of many terms. The alternative is to expand the electron distribution function in a series of spherical harmonics defined by an axis aligned with the electric field. This creates a sequence of linked equations. Once the zeroth-order equation is solved for the leading term of the expansion, then the first-order term can be solved, and so on. The zeroth term describes the average energy of an isotropic cloud of electrons, and the first term describes the drift of this cloud along the field, a current. The result given by Eq. (35) is an approximation to the isotropic part of the electron distribution. References 8, 9, and 10 describe kinetic theory and the mathematics of the Boltzmann equation. References 11, 12, and 13 describe the theory for electrons in a gas.

The mechanical constitutive equation of a material relates the stress tensor to the deformation tensor for a solid, and to the rate of strain tensor for a fluid. Many engineering materials are characterized by linear isotropic constitutive relations: the generalized Hooke's law for solids; and the Newtonian fluid. Technology is rapidly increasing the application of nonlinear "engineered" solids and plastic "rheological" fluids. These materials can have stress dependent on the deformation and rate of strain in a nonlinear way, on higher velocity derivatives, on anisotropies of their internal structure, and on the history of their deformation and motion. In the most general case the constitutive relation is an integro-differential equation that relates stress to the entire history of the material. One example follows:

$$m\frac{d^2u(t)}{dt^2} + au(t) + \int_0^t K(t-s)\frac{du(s)}{ds}\,ds = q(t) \qquad (36)$$

This is a mass-spring-damper equation with heredity in the damping term. Here u is distance, t is time, m is mass, a is the spring constant, K is a renewal kernel, and q is a forcing term. References 14 and 15 describe this equation. Reference 2 shows how to solve linear, constant-coefficient renewal equations with Laplace transforms. The solution of Eq. (36) by the methods of this article is

$$\begin{aligned} mu(t) = {} & u(0)\left[m + \int_0^t \int_0^s K(x)\,dx\,ds\right] \\ & + mp(0)t + \int_0^t (t-s)q(s)\,ds \\ & - \int_0^t u(s)\left\{a(t-s) + \int_0^{t-s} K(x)\,dx\right\} ds \end{aligned} \qquad (37)$$

where $u(0)$ and $p(0)$ are the initial conditions of the displacement u and its first derivative $p = du/dt$. Notice that the kernel is a function of one variable. For $K = q = 0$ the problem collapses to a harmonic oscillator, and it is easy to show that $\sin(\sqrt{a/m}\,t)$ is a solution of the reduced form of Eq. (37). Volterra had shown that kernels of the type $K(t - s)$ produce periodic solutions (see Refs. 4, 16, and 17). In general we can expect heredity to alter the frequency of oscillations, to introduce a damping, and to shift the mean position, all during the course of time.

An integro-differential equation for heat transfer occurs when the constitutive relation between heat flux and temperature gradient in the material is a hereditary integral. In a similar way, an integro-differential equation describes the evolution of an electric field in the vicinity of a nonconducting material dielectric with a memory of its charging history (a "Maxwell–Hopkinson dielectric"). These and other applications are described in Ref. 18, a mathematician's treatise on integro-differential equations.

The technological application of mechanics with heredity and of nonequilibrium kinetics is likely to drive future efforts to improve the solution of integro-differential problems. These problems arise in the development of nonequilibrium processes, such as plasma-chemical reactors for modifying material surfaces, and in the development of synthetic materials with engineered physical properties. Another thrust to solving these equations is the desire to improve our understanding of natural phenomena and materials. It is not hard to imagine that natural flows like lava or glaciers, and natural cycles like climate and weather, can have a hereditary factor. It would be interesting to have a method for easily estimating a distant cycle-time average of a quantity influenced by heredity, be it metal fatigue or species extinction.

BIBLIOGRAPHY

1. B. Sherman, The difference-differential equation of electron energy distribution in a gas, *J. Math. Anal. Appl.,* **1**: 342–354, 1960.
2. R. Bellman and K. L. Cooke, *Differential-Difference Equations,* New York: Academic Press, 1963.
3. F. B. Hildebrand, *Methods of Applied Mathematics,* 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1965.
4. H. T. Davis, *Introduction to Nonlinear Differential and Integral Equations,* Washington, DC: United States Atomic Energy Commission, USGPO, 1960; New York: Dover, 1962.
5. F. B. Hildebrand, *Advanced Calculus for Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1962.
6. J. Matthews and R. L. Walker, *Mathematical Methods of Physics,* 2nd ed., Menlo Park, CA: W. A. Benjamin, 1970.
7. S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases,* 3rd ed., London: Cambridge University Press, 1970.
8. T. I. Gombosi, *Gaskinetic Theory,* New York: Cambridge University Press, 1994.
9. C. Cercignani, *Mathematical Methods in Kinetic Theory,* 2nd ed., New York: Plenum Press, 1990.
10. C. Cercignani, *Theory and Application of the Boltzmann Equation,* New York: Elsevier, 1975.
11. T. Holstein, Energy distribution of electrons in high frequency gas discharges, *Phys. Rev.,* **70** (5, 6): 1946.
12. W. L. Nighan, Electron energy distributions and collision rates in electrically excited N_2, CO, CO_2, *Phys. Rev. A,* **2** (5): 1989–2000, 1970.
13. B. E. Cherrington, *Gaseous Electronics and Gas Lasers,* New York: Pergamon Press, 1979.
14. E. Volterra, On elastic continua with hereditary characteristics. *J. Appl. Mech.,* **18**: 273–279, 1951.

15. V. Volterra, Sur la théorie mathématique des phénomenès héréditaires. *J. Math. Pures Appl.,* **7**: 249–298, 1928.
16. V. Volterra, *Leçons sur les équations intégrales et les équations integro-differentielles,* Paris: Gauthier-Villars, 1913.
17. V. Volterra, *Leçons sur la théorie mathématique de la lutte pour la vie,* Paris: Gauthier-Villars, 1931.
18. F. Bloom, *Ill-Posed Problems for Integrodifferential Equations in Mechanics and Electromagnetic Theory,* Philadelphia: SIAM, 1981.

MANUEL GARCIA
Lightning Publications

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Laplace Transforms
Standard Article
James A. Ritcey[1]
[1]University of Washington, Seattle, WA
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2430
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (184K)

Abstract

The sections in this article are

The Laplace Transform and Linear time Invariant Systems

First Applications

Basic Properties of Unilateral Laplace Transforms

Advanced Transform Pairs

Applications of the Laplace Transform in Probability

| | |

Copyright © 1999-2008 All Rights Reserved.

LAPLACE TRANSFORMS

In this article, we describe the fundamentals of the Laplace transform. This is one of the many integral transforms and is used primarily to simplify and solve differential equations, integral equations, and interconnected linear systems, among others. It is especially useful when convolution is involved. This occurs in both linear system theory and probability.

The Laplace Transform was introduced by Marquis Pierre-Simon de Laplace (1749–1827), the great French mathematician and astronomer. Many of the modern applications were developed by Oliver Heaviside (1850–1925), a British engineer. Heaviside was prolific and highly original, but he had many critics.

The Laplace transform is equal in importance to the Fourier transform, and both are mainstays of all undergraduate curricula in mathematics, physics, and engineering. They are also important in probability theory. Many of the examples in this article are drawn from electrical engineering.

The Laplace and Fourier transforms are the only integral transforms discussed here, but they are hardly the only integral transforms. Many of these are closely related to the Laplace transform and include the Mellin transform.

THE LAPLACE TRANSFORM AND LINEAR TIME INVARIANT SYSTEMS

Many systems in electrical engineering are linear and time invariant (LTI). Let L be a continuous time (CT) LTI system with input $x(t)$ and output $y(t) = L[x(t)]$. Let $\delta(t)$ be the Dirac delta, or CT impulse. Then $h(t) = L[\delta(t)]$, the response of the linear system to a unit impulse, is called the *impulse response.* Time invariance leads to the *convolution* representation of the system,

$$y(t) = (h * x)(t) = (x * h)(t) \tag{1}$$

$$= \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)\,d\tau \tag{2}$$

The limits of integration depend on the causality properties of both the input signal x and the impulse response h. If the system is causal, $h(t) = 0$, $\forall t < 0$. If the input also begins at time $t = 0$, then the integral becomes

$$y(t) = \int_0^t h(\tau)x(t-\tau)\,d\tau \tag{3}$$

An eigenvalue of the linear system L is a complex number $H = H(s)$, depending on a complex parameter $s = \sigma + j\omega$, that is associated with an eigenfunction $\phi(t; s)$, which satisfies

$$L[\phi(t;\, s)] = H(s)\phi(t;\, s) \tag{4}$$

The solution is

$$\phi(t;\, s) = \exp(st) \tag{5}$$

$$H(s) = \int_{-\infty}^{+\infty} e^{-st}h(t)\,dt \tag{6}$$

$H(s)$ is called the *transfer function* of the system and is the Laplace transform of $h(t)$.

The limits of integration extend only over the support of h or the set of t for which h is nonzero. If the limits are $0 \le t < \infty$, we work with the unilateral or one-sided Laplace transform. When the limits are $-\infty < t < +\infty$, this Laplace transform is called bilateral or two-sided. For any time function $x(t)$, we write $x \bullet\!\!\rightarrow X$ or $X(s) = \mathcal{L}\{x(t)\}$ (equivalently $x(t) = \mathcal{L}^{-1}\{X(s)\}$) to denote the corresponding Laplace transform pair

$$X(s) = \int x(t)e^{-st}\,dt$$

When the limits are unspecified, they are determined by the support of $x(t)$.

The integral defining the Laplace transform may exist only for some complex s, and this set is called the Region of Convergence (ROC) of the Laplace Transform for the time function $x(t)$. For the *unilateral* Laplace transform, the ROC is nonempty, and the Laplace transform exists (converges) for some s if and only if x is of exponential order

$$x(t) = O(e^{kt}) \text{ as } t \to \infty \tag{7}$$

where it is important to note that k can be positive.

In general, we say that f is "big-Oh of g,"

$$f(t) = O(g(t)) \text{ as } t \to \infty \text{ if } \lim_{t\to\infty} \frac{f(t)}{g(t)} < \infty$$

Then, f and g are said to be of the same order. For example, $\sin(t) = O(1)$.

Example 1. Exponentially Decaying Impulse Response. Here we consider the exponential impulse response

$$h(t) = Ae^{-at}u(t) \tag{8}$$

where $u(t)$ is the *unit step* function

$$u(t) := \int_0^t \delta(\tau)\,d\tau \tag{9}$$

$$= 1,\ t \ge 0 \tag{10}$$

$$= 0,\ t < 0 \tag{11}$$

and a is real. We find that

$$H(s) = \mathcal{L}\{h(t)\} \tag{12}$$

$$= \int_0^\infty Ae^{-at}e^{-st}\,dt \tag{13}$$

$$= \frac{A}{s+a}, \quad \text{ROC} = \{\mathcal{R}(s) > -a\} \tag{14}$$

where $\mathcal{R}$ denotes "real part of." This H is a *rational function,* a ratio of polynomials in s. Rational Laplace transforms arise frequently in the analysis of engineering systems and corre-

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

spond to lumped element systems. The roots of the numerator polynomial are called the *zeros,* whereas the roots of the denominator are called the *poles.* This system has one pole at $s = -a$, and no finite zeros, although $|H| \to 0$ as $s \to \infty$.

The linear system associated with this exponential impulse response is called a *first-order* or *one-pole* system. The differential equation is

$$u'(t) + ay(t) = x(t) \tag{15}$$

where $y'(t) := dy(t)/dt$. It is interesting to note that the transfer function H can be determined directly from the differential equation through the eigenvalue property, without the intermediate step of finding the impulse response $h(t)$. Substituting $x(t) = \exp(-st)$ and $y(t) = H(s)x(t)$ into Eq. (15) we directly solve for $H(s) = A(s + a)^{-1}$.

We now allow the decay rate a to be complex, $a = b + j\beta$. The pole and ROC remain unchanged. The time function

$$x(t) = e^{-bt}\cos(\beta t)u(t)$$

can be simply obtained from earlier results. Treat s as *real-valued* for the moment, and use the linearity of the Laplace transform integral to see that

$$X(s) = \mathscr{L}\{\Re^{-(b+j\beta)t}\} \tag{16}$$

$$= \Re(s + b + j\beta)^{-1} \tag{17}$$

$$= \frac{s+b}{(s+b)^2 + \beta^2} \tag{18}$$

Finally, let s extend over the complete ROC, the half plane $\{\mathscr{R}(s) > -b\}$.

When $\mathscr{R}(a) < 0$, the ROC includes the $s = 2\pi jf$ axis. In this case, we say that the system L has a *frequency response* or Fourier transform. We will explore some basic properties via $\mathscr{H}(f) = H(s = 2\pi jf)$,

$$\mathscr{H}(f) = \int_{-\infty}^{+\infty} e^{-2\pi jft}h(t)\,dt \tag{19}$$

The input to the system is a pure complex tone, and oscillates without decay ($\mathrm{Re}(s) = 0$). Revisiting the eigenvalue property, we confirm the fact that a monochromatic (pure tone) put into a linear system results in an output at the same frequency, with a shift in amplitude and phase.

Decompose $\mathscr{H}(f) = A(f)\exp[j\Phi(f)]$, into the magnitude response $A(f)$ and the phase response $\Phi(f)$ so that

$$A(f) = |\mathscr{H}(f)|, \quad \text{an even function} \tag{20}$$

$$\Phi(f) = \arg\mathscr{H}(f), \quad \text{an odd function} \tag{21}$$

For the one-pole system, we see that the Fourier Transform $\mathscr{H}$ is given by

$$\mathscr{H}(f) = \frac{A}{a + 2\pi jf} \tag{22}$$

$$\text{with } A(f) = \frac{A}{\sqrt{a^2 + (2\pi f)^2}} \tag{23}$$

$$\text{and } \Phi(f) = \tan^{-1}(2\pi f/a) \tag{24}$$

The Fourier and Laplace Transform is discussed in many undergraduate texts in engineering. A good starting point is is either of Papoulis' texts (1,2).

FIRST APPLICATIONS

Example 2. Linear Differential Equation with Constant Coefficients. The Laplace transform is especially useful in solving initial value problems. Consider

$$y''(t) + 3y(t) + 2y(t) = x(t) = e^{-3t}u(t) \text{ for } t \geq 0 \tag{25}$$

We seek a solution subject to the initial conditions

$$y'(0) = y'_0 = -3 \tag{26}$$

$$y(0) = y_0 = 1 \tag{27}$$

Transforming both sides of Eq. (25) yields

$$[s^2Y(s) - sy_0 - y'_0] + 3[sY(s) - y_0] + 2Y(s) = \frac{1}{s+3} \tag{28}$$

Substituting $y_0 = +1$, $y'_0 = -3$, we find

$$[s^2 + 3s + 2]Y(s) = s + X(s) = s + \frac{1}{s+3} \tag{29}$$

This can be interpreted as the superposition of response to the input $x(t)$ and the initial conditions. Define the transfer function

$$H(s) = \frac{1}{s^2 + 3s + 2} = \frac{1}{(s+1)(s+2)} \tag{30}$$

so that

$$Y(s) = H(s)X(s) + H(s)[sy_0 + y'_0 + 3y_0] \tag{31}$$

Substituting and solving gives

$$Y(s) = \frac{s^2 + 3s + 1}{(s+1)(s+20)(s+3)} \tag{32}$$

A partial fraction expansion quickly yields

$$Y(s) = -\frac{1}{2}\frac{1}{s+1} + \frac{1}{s+2} + \frac{1}{2}\frac{1}{s+3} \tag{33}$$

The ROC is $\{\mathscr{R}(s) > -1\}$, so the inverse transform can be written

$$y(t) = [e^{-2t} + \frac{1}{2}(e^{-3t} - e^{-t})]u(t) \tag{34}$$

The slowest mode decays as $O(e^{-t})$, and corresponds to the largest pole at $s = -1$.

Example 3. Linear Feedback Control Systems Analysis. The Laplace transform is widely used to analyze the stability and

response of linear feedback control systems. Consider the block diagram in Fig. 1.

Operating in steady state, we find that

$$Y(s) = K(s)E(s) \tag{35}$$

$$E(s) = X(s) - G(s)Y(s) \tag{36}$$

or

$$Y(s) = H(s)X(s) \tag{37}$$

where

$$H(s) = \frac{K(s)}{1 + K(s)G(s)} \tag{38}$$

K is called the *open-loop* transfer function, whereas H is the *closed-loop* transfer function. To illustrate the advantages of working in the s domain, we compare it to the time domain formulation. Consider the particular example of

$$K(s) = K/s \tag{39}$$

$$G(s) = s + \frac{1}{s+a} \tag{40}$$

so that

$$H(s) = \frac{K(s+a)}{K_1 s^2 + K_1 a s + K} \tag{41}$$

where $K_1 = K + 1$. The time domain system equations are

$$y(t) = K \int_{-\infty}^{t} e(\tau)\, d\tau \tag{42}$$

where

$$e(t) = x(t) - y(t) - \int_{-\infty}^{t} e^{-a\tau} y(t-\tau)\, d\tau \tag{43}$$

This integrodifferential equation is difficult to analyze or solve for particular inputs $x(t)$. However, when $X(s)$ is available, the solution is straightforward, given that we can invert $Y(s)$ in Eq. (37), with $H(s)$ given by Eq. (41).

Example 4. Solving Partial Differential Equations by the Laplace Transform. Laplace transforms can also be used to solve linear partial differential equations (PDEs). Examples include the diffusion equation and the wave equation. In this example we solve the diffusion equation, a linear PDE that is first order in time and second order in space. The problem prescribes boundary conditions in space and an initial condition in time.

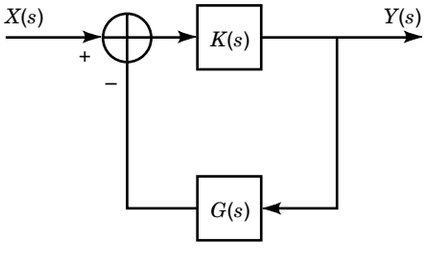


Figure 1. Control system block diagram.

Let $u = u(x, t)$ be twice continuously differentiable in x and t on $\{t \geq 0\}$ and $\{0 \leq x \leq L\}$. Denote partial derivatives by subscripts so that

$$u_{xx} = \frac{\partial^2}{\partial x^2} u(x, t) \quad u_t = \frac{\partial}{\partial t} u(x, t)$$

The diffusion equation with diffusion constant κ is written

$$u_{xx} = \frac{1}{\kappa} u_t, \quad t \geq 0,\ 0 \leq x \leq L \tag{44}$$

We assume that the boundary conditions are

$$u(0, t) = T_0 \tag{45}$$

$$u(x, 0) = 0 \tag{46}$$

Our approach is to first transform with respect to time t, and obtain an ordinary differential equation (ODE) for $U(x, s)$. Next, we solve for $U(x, s)$, subject to the boundary conditions. Finally, we invert to obtain $u = u(x, t)$.

Define

$$U(x, s) = \int_0^\infty u(x, t) e^{-st}\, dt \tag{47}$$

subject to

$$U(0, s) = \int_0^\infty u(0, t) e^{-st}\, dt \tag{48}$$

$$= T_0/s \tag{49}$$

Using basic properties of the Laplace transform, we reduce the PDE to an ODE in x. Transforming gives

$$U_{xx}(x, s) - \frac{1}{\kappa}[sU(x, s) - u(x, 0)] = 0 \tag{50}$$

Substituting the boundary condition gives the ODE, which we view as parametric in s,

$$I_{xx}(x, s) = \frac{s}{\kappa} U(x, s) \tag{51}$$

subject to

$$U(0, s) = \frac{T_0}{s} \tag{52}$$

Solving this for a fixed s yields

$$U(x, s) = \frac{T_0}{s} \exp\left(-x\sqrt{\frac{s}{\kappa}}\right) \tag{53}$$

The remaining task is to carry out the inversion and find

$$u(x, t) = \mathscr{L}^{-1}\left\{\frac{T_0}{s} \exp\left(-x\sqrt{\frac{s}{\kappa}}\right)\right\} \tag{54}$$

We will solve this inversion problem using a roundabout approach. Differentiate with respect to x, and write out the inversion integral to obtain

$$u_x(x,\,t) = \int \frac{ds}{2\pi j}\left(-\frac{T_0}{\sqrt{s\kappa}}\right) e^{-x\sqrt{\frac{s}{\kappa}}}\, e^{st}$$

Substitute $\sqrt{s} = j\omega$ and simplify to obtain

$$u_x(x,\,t) = \frac{-2T_0}{\sqrt{\kappa}} \int \frac{d\omega}{2\pi}\, e^{-j\omega\frac{x}{\sqrt{\kappa}} - t\omega^2}$$

Completing the square in ω,

$$-t\omega^2 - j\omega(x/\sqrt{\kappa}) = -t\left(\omega - \frac{jx}{2t\sqrt{\kappa}}\right)^2 - \frac{x^2}{4\kappa t}$$

we substitute and simplify to get

$$u_x(x,\,t) = \frac{-2T_0}{\sqrt{\kappa}}\, e^{-\frac{x^2}{4\kappa t}} \int \frac{d\omega}{2\pi}\, \exp\left\{-t\left(\omega - \frac{jx}{2t\sqrt{\kappa}}\right)^2\right\}$$

This last integral is evaluated by normalization because it is essentially the probability mass under the Gaussian curve, and we find that

$$u_x(x,\,t) = \frac{-T_0}{\sqrt{\pi\kappa t}}\, e^{-x^2/4\kappa t} \tag{55}$$

Integrating with respect to x yields

$$u(x,\,t) = T_0\left[1 - \operatorname{erf}\left(\frac{x}{2}\sqrt{\frac{1}{\kappa t}}\right)\right] \tag{56}$$

where $\operatorname{erf}(x) = (2/\sqrt{\pi}) \int_0^x e^{-t^2}\, dt$.

BASIC PROPERTIES OF UNILATERAL LAPLACE TRANSFORMS

In this section, we list a number of the basic properties of the unilateral or "one-sided" Laplace transform, adopted from Henrici (3). We let $f(t)$ denote a time function, and $F(s)$, the corresponding transform, $f \bullet\!\!\longrightarrow F$. We assume that $f(t)$ satisfies some reasonable properties.

1. $f(t)$ is identically zero for all negative time

$$f(t) = 0,\ \forall t < 0 \tag{57}$$

2. $f(t)$ is continuous except for a countable number of step discontinuities on the nonaccumulating set of points $0 < t_1 < t_2 < \ldots$
3. $f(t)$ is absolutely integrable ($f \in L_1[0, \infty]$)

$$\int_0^\infty |f(t)|\, dt < \infty \tag{58}$$

Here the Laplace transform of $f(t)$, denoted $f(t) \bullet\!\!\longrightarrow F(s)$ or $F(s) = \mathcal{L}\{f(t)\}$, consists of *both* the complex function

$$F(s) = \int_0^\infty f(t) e^{-st}\, dt \tag{59}$$

and the associated ROC, $\mathcal{R}(s) > s_0$. Many of the properties continue to hold for suitably defined time functions on the whole real line $-\infty < t < +\infty$.

Property 1. Inversion and Uniqueness. A Laplace transform $F(s)$ and its ROC can be inverted to a unique time function $f(t)$. Note that not all functions of a complex variable s are Laplace transforms. Techniques to carry out the inversion are discussed later. Usually inversion is the most difficult part of the process, either theoretically or computationally.

Property 2. Linearity. If $f \bullet\!\!\longrightarrow F$ and $g \bullet\!\!\longrightarrow G$, then for constants a, b,

$$af + bg \bullet\!\!\longrightarrow aF + bG \tag{60}$$

This extends directly for finite sums.

Property 3. Analyticity. Within its ROC, $F(s)$ is an analytic function. This implies that derivatives of all orders exist and can be computed by

$$\frac{d^k}{ds^k} F(s) = \int_0^\infty (-t)^k e^{-st}\, dt \tag{61}$$

Property 4. Decay in $F(s)$.

$$\lim_{s\to\infty} F(s) = 0 \tag{62}$$

where the limit is taken along any ray lying within the ROC of F.

Property 5. Shifting.

$$f(t - t_0) \bullet\!\!\longrightarrow e^{-st_0} F(s) \tag{63}$$

Property 6. Scaling.

$$f(at) \bullet\!\!\longrightarrow \frac{1}{a} F\left(\frac{s}{a}\right) \tag{64}$$

Property 7. Differentiation. When $f'(t) \in L_1$ with an initial value $f(0+)$,

$$f'(t) \bullet\!\!\longrightarrow sF(s) - f(0+) \tag{65}$$

When $f(t)$ is sufficiently differentiable and $f^{(n)}(t) := (d/dt)^n f(t)$ with $f^{(0)}(t) := f(t)$, we find that

$$f^{(n)}(t) \bullet\!\!\longrightarrow s^n F(s) - \sum_{k=0}^{n-1} s^{n-1-k} f^{(k)}(0+) \tag{66}$$

For example,

$$f''(t) \bullet\!\!\longrightarrow s^2 F(s) - sf(0+) - f'(0+) \tag{67}$$

This is the most useful property in solving initial value problems that arise from linear circuits or mechanical problems.

Property 8. Integration.

$$\int_0^t f(\tau)\, d\tau \bullet\!\!\longrightarrow \frac{1}{s} F(s) \tag{68}$$

Property 9. Convolution. If $f(t) \bullet\!\!\longrightarrow F(s)$ and $h(t) \bullet\!\!\longrightarrow H(s)$, then

$$(h * f)(t) = \int_0^t h(\tau) f(t-\tau)\, d\tau \bullet\!\!\longrightarrow F(s)H(s) \tag{69}$$

Property 10. Multiplication. If $f(t) \bullet\!\!\longrightarrow F(s)$ and $g(t) \bullet\!\!\longrightarrow G(s)$, then

$$f(t)g(t) \bullet\!\!\longrightarrow \int_C \frac{d\lambda}{2\pi j} G(\lambda)H(s-\lambda) \tag{70}$$

An important special case is when $s = 0$, and yields a Parseval theorem for the Laplace transform

$$\int_0^\infty f(t)g(t)\, dt = \int_C \frac{d\lambda}{2\pi j} F(\lambda)G(-\lambda) \tag{71}$$

In both cases, a suitable contour of integration in the complex λ plane is required. This contour C is vertical and lies within the ROC of both F and G, $\mathrm{ROC}_F \cap \mathrm{ROC}_G$. If this intersection is empty, the resulting integral does not exist.

Property 11. Periodic Functions. Let $f_0(t)$ be defined over the fundamental period $0 < t < T$, and let

$$f(t) = \mathrm{rep}_T[f_0(t)] := \sum_{n=-\infty}^{\infty} f_0(t + nT) \tag{72}$$

so that $f(t) = f(t + T)$ is T periodic. Then if $f_0(t) \bullet\!\!\longrightarrow F_0(s)$,

$$f(t) \bullet\!\!\longrightarrow F(s) = \frac{1}{1 - e^{-sT}} \int_0^T e^{-st} f(t)\, dt \tag{73}$$

Property 12. Multiplication by t^k.

$$t^k f(t) \bullet\!\!\longrightarrow (-1)^k F^{(k)}(s) \tag{74}$$

This property is very useful in quickly evaluating Laplace transforms. For example, from Example 1 we know that

$$e^{-t} \bullet\!\!\longrightarrow (s+1)^{-1} \tag{75}$$

Using this property, we immediately see that

$$\frac{t^{k-1}e^{-t}}{k!} \bullet\!\!\longrightarrow (s+1)^{-k} \tag{76}$$

ADVANCED TRANSFORM PAIRS

In this section we present further properties, and applications, of a more advanced nature.

Log Functions

An Application of Differentiation. Assume that a time function, $f = f(t;\, \nu)$, depends on a parameter ν. We further assume that f is analytic in ν on some open set $\mathcal{D}$, so that derivatives of any order k, $(d/d\nu)^k f(t,\, \nu)$, exist. Then if

$$F(s;\, \nu) = \mathcal{L}\{f(t;\, \nu)\} \tag{77}$$

we find that

$$(d/d\nu)^k F(s;\, \nu) = \mathcal{L}\{(d/d\nu)^k f(t;\, \nu)\} \tag{78}$$

We apply this result to the determination of $\mathcal{L}\{\log t\}$.

Recall that the Laplace transform of t^ν can be found from $\Gamma(x)$, the Gamma function,

$$\Gamma(\nu) = \int_0^\infty t^{\nu-1} e^{-t}\, dt \tag{79}$$

The correspondence is

$$\mathcal{L}\{t^\nu\} = \int_0^\infty t^{\nu-1} e^{-st}\, dt \tag{80}$$

$$= s^{-\nu-1}\Gamma(\nu+1) \tag{81}$$

Note that both sides are analytic functions of ν on the ν plane, cut on the negative real ν axis. Differentiate with respect to ν, and evaluate the result at $\nu = 0$ to find

$$\mathcal{L}\{\log t\} = -\frac{1}{s}(\log s + \gamma) \tag{82}$$

where $\gamma := -\Gamma'(1) = 0.57721566$ is Euler's constant. The method can be repeated, for example, to find $\mathcal{L}\{\log^2 t\}$.

An Application of Integration. Again consider the time function, $f = f(t;\, \nu)$, depending on a parameter ν. We further assume that f is integrable with respect to ν, so that $\int f(t;\, \nu)\, d\nu$ exists. Then, if $F(s;\, \nu) = \mathcal{L}\{f(t;\, \nu)\}$,

$$\mathcal{L}\{\textstyle\int f(t;\, \nu)\, d\nu\} = \int F(s;\, \nu)\, d\nu \tag{83}$$

As an example, recall that

$$\mathcal{L}\{\cos(bt)\} = \frac{s}{s^2 + b^2} \tag{84}$$

Integrating both sides with respect to b yields the transform pair

$$\mathcal{L}\left\{\frac{\sin(bt)}{t}\right\} = \tan^{-1}(b/s) \tag{85}$$

Series Expansions

Hardy's theorem is frequently invoked to find the transform or inverse transform of a time function given as a convergent series. There are two closely related variants: for Laplace transforms and for inversions.

Let

$$f(t) = t^\nu \sum_{n=0}^{\infty} a_n t^n \tag{86}$$

and consider $F(s)$ obtained by term-by-term integration of f,

$$\mathcal{L}\{f(t)\} = \sum_{n=0}^{\infty} a_n \int_0^{\infty} t^{\nu+n} e^{-st}\, dt \tag{87}$$

$$F(s) = \sum_{n=0}^{\infty} a_n \Gamma(\nu + n + 1) s^{-\nu-n-1} \tag{88}$$

Hardy's Theorem. If $F(s)$ is convergent for some $s = s_0 > 0$, then $f(t)$ converges for all $t > 0$ and $F(s) = \mathcal{L}\{f(t)\}$.

The second version is a converse.

Corollary. Let

$$F(s) = s^{-\nu} \sum_{n=0}^{\infty} c_n s^{-n} \tag{89}$$

which converges for some $|s| \geq \rho > 0$ and $|\arg s| < \pi$. Then $F(s)$ is the analytic continuation of $\mathcal{L}\{f(t)\}$, where

$$f(t) = \sum_{n=0}^{\infty} \frac{c_n}{\Gamma(\nu + n)} t^{\nu+n-1} \tag{90}$$

This provides a useful inversion theorem.

We apply this to find the Laplace transform for the Bessel function J_0. Consider

$$F(s) = s^{-1} e^{-a/s} \tag{91}$$

$$= s^{-1} \sum_{n=0}^{\infty} \frac{(-a)^n}{n!} s^{-n} \tag{92}$$

Applying the corollary and inverting term by term gives

$$\mathcal{L}^{-1}\{F(s)\} = \sum_{n=0}^{\infty} \frac{(-a)^n}{n!} s^{-n} \tag{93}$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n (\sqrt{at})^{2n}}{n!n!} \tag{94}$$

$$= J_0(2\sqrt{at}) \tag{95}$$

The Saddlepoint Approximation and Numerical Contour Integration

Many times the inversion integral cannot be carried out analytically. This often occurs when the integrand contains branch cuts and essential singularities (2), or when the number of poles is so large as to preclude numerical summation of the residue series. In these cases, techniques from asymptotic analysis (3–5) suggest some useful numerical methods. We refer to these as *saddlepoint* methods.

Consider the inversion integral

$$f(t) = \int_C \frac{ds}{2\pi j} F(s) e^{st}$$

where C is a suitable contour. We will consider the generic form

$$f(t) = \int_C \frac{ds}{2\pi j} e^{\phi(s)}$$

where

$$\phi(s) = st + \log F(s)$$

The method relocates the path of integration C so that it passes through a saddlepoint s_0, where

$$\phi'(s_0) = \frac{d}{ds} \phi(s)|_{s_0} = 0$$

In many cases, especially for large t, the main contribution to $f(t)$ arises from points on C close to s_0. Expand $\phi(s)$ about the point s_0,

$$\phi(s) = \phi(s_0) + \phi''(s_0) \frac{(s - s_0)^2}{2!} + \cdots$$

because $\phi'(s_0) = 0$. Substituting, with $s = s_0 + jy$, we find

$$f(t) = \frac{e^{\phi_0}}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \frac{ds}{\sqrt{s\pi}} e^{-\frac{\phi_0'' y^2}{2}} \tag{96}$$

where $\phi_0 = \phi(s_0)$, $\phi_0'' = \phi''(s_0)$. The integral is evaluated by normalization to give the saddlepoint approximation

$$f(t) = \frac{e^{\phi_0}}{\sqrt{2\pi\phi_0''}} \tag{97}$$

The methods can be extended by numerically integrating along the "steepest descent" contour from the saddlepoint. For applications to probability, and in particular in communications and statistical signal processing, see Refs. 6 to 8.

APPLICATIONS OF THE LAPLACE TRANSFORM IN PROBABILITY

The Laplace Transform is a natural tool for handling many distributional problems in applied probability and mathematical statistics, especially those involving linear combinations of statistically independent positive random variables. For these, a natural integral transform is the one-sided Laplace transform. We begin this section with some basic concepts about random variables and their distributions and lead into the use of transform techniques. Appropriate references include Papoulis (9) and Feller (10). Because the inversion problem is usually the most difficult step in the analysis, some approximate methods from asymptotic analysis are discussed. Detection theory at the level of Helstrom (6) is a source of many of our applications.

Random Variables and Their Distributions

Let $\mathbf{x}$ denote a positive random variable (rv) with probability density function (pdf) $f(x)$, distribution function (df) $F(x)$, and exceedance $\overline{F}(x) := 1 - F(x)$. The pdf is a positive integrable function defined on $(0, \infty)$, with unit normalization,

$$\int_0^{\infty} f(x)\, dx = 1$$

The probability that the rv $\mathbf{x}$ falls into some interval $[a, b]$ where $0 \leq a \leq \infty$ can be determined from either f or F,

$$\Pr\{a < \mathbf{x} \leq b\} = F(b) - F(a) \tag{98}$$

$$= \int_a^b f(x)\,dx \tag{99}$$

Do not confuse the df $F(x)$ with a Laplace transform. The pdf is also used in the concept of the *expectation* of a measurable function of a rv. For a suitable g, define

$$E\{g(x)\} = \int_0^\infty g(x) f(x)\,dx$$

The *moment generating function* (mgf) is defined as the Laplace transform of the density $f(x)$. Alternatively, it can be interpreted as an expectation. We will use $h = h(s)$ or $h_{\mathbf{x}}(s)$ to denote

$$h(s) = E\{\exp(-\mathbf{x}s)\} \tag{100}$$

$$= \int_0^\infty e^{-xs} f(x)\,dx \tag{101}$$

Note the distinction between the rv $\mathbf{x}$ and the parameter x. It is this probabilistic interpretation of the mgf which makes it so useful in theory and application. The mgf exists within the ROC of the Laplace transform. By normalization of the pdf, $h(s = 0) = 1$. In fact, other properties hold. Bernstein's theorem states that a function $h(s)$ is a mgf if and only if it is a *completely monotonic* (cm) function

$$(-1)^k \frac{d^k}{ds^k} h(s) \geq 0 \;\forall s \geq 0$$

In many of our applications in detection theory, *tail probabilities* are of interest. The right-hand tail is simply

$$\overline{F}(x) = \int_x^\infty f(x)\,dx$$

where x is so large that $\overline{F}(x) \lll 0.5$. For example, to find P_e, the probability of error of a digital communications system, we are often asked to evaluate tail probabilities of the order of 10^{-5} to 10^{-10}.

The distribution is a complete statistical description of the random variable $\mathbf{x}$. Often simpler descriptors suffice. The most common are the *moments* of the rv. Define μ_k, the kth moment of the rv $\mathbf{x}$, by

$$\mu_k = E\{\mathbf{x}^k\} = \int_0^\infty x^k f(x)\,dx \tag{102}$$

Moment generating function refers to the fact that the moments are determined by differentiating $h(s)$. Expanding e^{-xs} and integrating term by term gives

$$h(s) = \int_0^\infty e^{-xs} f(x)\,dx \tag{103}$$

$$= \sum_{k=0}^\infty \int \frac{(-x)^k}{k!} f(x)\,dx \tag{104}$$

$$= \sum_{k=0}^\infty (-1)^k \frac{\mu_k}{k!} \tag{105}$$

Equating this moment expansion with the Taylor series for $h(s)$ gives

$$\mu_k = E\{x^k\} = (-1)^k h^{(k)}(0)$$

where the kth derivative is denoted $h^{(k)}(s) := (d/ds)^k h(s)$. Thus, the moments can be easily determined from the mgf.

One of the most important uses of the mgf is in analyzing the distribution of sums of independent random variables. Let $\mathbf{t}_1, \ldots, \mathbf{t}_n$ be a collection of n independent and identically distributed (iid) component random variables with common pdf $p(x)$ and mgf $g(s) = E\{\exp(-\mathbf{t}s)\}$. Then if $\mathbf{x} = \mathbf{t}_1 + \cdots + \mathbf{t}_n$,

$$h(s) = E\{\exp(-\mathbf{x}s)\} \tag{106}$$

$$= E\left\{\exp\left(-s \sum_{j=1}^n \mathbf{t}_j\right)\right\} \tag{107}$$

$$= \prod_{j=1}^n E\{\exp(-s\mathbf{t}_j)\} \tag{108}$$

$$= g^n(s) \tag{109}$$

The mgf of the sum of n iid random variables is nth power of the mgf of the individual components. This extends easily to the nonidentically distributed case. We illustrate many of these general properties with an example.

Sums of Independent Exponentially Distributed Random Variables. Again let $\mathbf{t}_1, \ldots, \mathbf{t}_n$ be a collection of n iid component random variables with common pdf $p(x)$ and mgf $g(s) = E\{e^{-}\mathbf{t}^s\}$. Of interest is the distribution of the sum, $\mathbf{x} = \mathbf{t}_1 + \cdots + \mathbf{t}_n$, under an exponential assumption on the distribution of the component random variables. When the $\mathbf{t}$ are exponentially distributed with mean $E\{\mathbf{t}\} = \mu$,

$$f(t) = \frac{1}{\mu} e^{-t/\mu}, \quad t \geq 0 \tag{110}$$

$$F(t) = 1 - e^{-t/\mu}, \quad t \geq 0 \tag{111}$$

$$g(s) = \frac{1}{1 + \mu s} \tag{112}$$

The moments are easily determined by differentiation,

$$\mu_k = E\{\mathbf{t}^k\} = k!\mu^k$$

The mgf of $\mathbf{x}$, the sum of the iid exponential components, is

$$h(s) = (1 + \mu s)^{-n}$$

To determine the density or distribution, we must invert this Laplace transform. The pdf is given by the contour integral representation of the inverse transform

$$f(x) = \int_{\mathrm{C}} \frac{ds}{2\pi j} h(s) e^{xs}$$

where C is a vertical contour in the complex s plane lying in the region of convergence. In our example, the ROC is the half place $\mathscr{R}(s) > -1/\mu$. The density can be obtained using

the method of residues. Closing the contour in the left half plane, and using Cauchy's integral formula,

$$f(x) = \oint \frac{ds}{2\pi j} \frac{e^{xs}}{(1+\mu s)^n} \tag{113}$$

$$= \frac{1}{\mu} \left(\frac{x}{\mu}\right)^{n-1} \frac{e^{-x/\mu}}{(n-1)!}, \quad x > 0 \tag{114}$$

In many problems, the distribution F or exceedance $\overline{F}$ is of more interest than the density f. General contour integral representations for the distribution and exceedance is

$$\overline{F}(x) = -\int_{C-} \frac{ds}{2\pi j} s^{-1} h(s) e^{xs} \tag{115}$$

$$F(x) = \int_{C+} \frac{ds}{2\pi j} s^{-1} h(s) e^{xs} \tag{116}$$

The contours C+, C− are both vertical and lie in the ROC of mgf $h(s)$ as shown in Fig. 2. The contour C− crosses the *negative* real s axis, whereas the contour C+ crosses the *positive* real s axis.

To obtain the cdf of $\mathbf{x}$, we will compute $\overline{F}(x)$ and obtain F by subtraction, $F(x) = 1 - \overline{F}(x)$. From the contour integral, with $h(s) = (1 + \mu s)^{-n}$, closing the contour are the pole at $s = -1/\mu$, and invoking Cauchy's integral formula, we find that

$$\overline{F}(x) = \int_{C-} \frac{ds}{2\pi j} \frac{(-s)^{-1} e^{xs}}{(1+\mu s)^n} \tag{117}$$

$$= \frac{\mu^{-n}}{(n-1)!} \left(\frac{d}{ds}\right)^{n-1} [(-s)^{-1} e^{xs}]_{1/\mu} \tag{118}$$

To complete the derivation, we must carry out the differentiation using Leibnitz's rule,

$$\left(\frac{d}{ds}\right)^m h(s)g(s) = \sum_{k=0}^{m} \binom{m}{k} h^{(k)}(s) g^{(m-k)}$$

to obtain a residue series. Simplifying gives

$$\overline{F}(x) = \sum_{k=0}^{n-1} \frac{1}{\mu} \left(\frac{x}{\mu}\right)^k \frac{e^{-x/\mu}}{k!}$$

The derivation of this formula is not the end of the story. When $\overline{F}(x)$ is small, that is, for large n and $x \gg n\mu = E\{\mathbf{x}\}$, the sum contains many terms and is difficult to evaluate as a result of the disparity in magnitude between the large $(x/\mu)^k$ and the small $\exp(-x/\mu)$. This disparity causes overflow, which must be handled carefully. The point is that a residue series often leads to numerical instability when the number of poles, or their multiplicity, is large.

Upper bounds on the exceedance $\overline{F}$, which are tight and easy to compute, are also important. A general technique to obtain these is based on Markov's inequality.

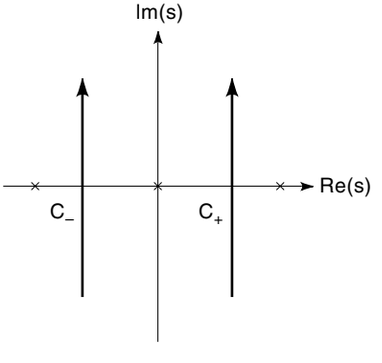


Figure 2. Bromwich contour definitions.

Markov's Inequality. Here $\mathbf{x}$ has exceedance $\overline{F}(x)$ and mgf $h(s)$. If $v(x; s)$ is a positive nondecreasing function for $x \in [0, \infty]$ and with parameter s,

$$\overline{F}(x_0) \le \min_s E\{v(\mathbf{x}; s)/v(x_0; s)\}$$

The most important special cases include the Chernoff bound, where $v(x; s) = \exp(-sx)$, $s \le 0$, and the moment bound, where $v(x; s) = (x)^s$, $s \ge 0$. The Chernoff bound is

$$\overline{F}(x_0) \le \min_{s \le 0} \{e^{sx_0} h(s)\}$$

For the example, we find that

$$\overline{F}(x_0) \le \min_{s \le 0} \{e^{sx_0} (1+\mu s)^{-n}\}$$

The best choice of $s = s_0 \le 0$ is at $s_0 = (n/x_0) - (1/\mu)$, which is negative for $x > n\mu = E\{\mathbf{x}\}$. Substituting s_0 yields the Chernoff bound,

$$\overline{F}(x) \le \frac{(x/\mu)^n e^{-x/\mu}}{n^n e^{-n}}$$

The denominator is asymptotic to $n!/\sqrt{2\pi n}$. An important feature of the Chernoff bound is the exponential rate of decay which captures the dominant features of the exact exceedance. The Laplace transform clearly plays a prominent role in the bound. These bounds can be developed into approximations using saddlepoint methods.

BIBLIOGRAPHY

1. A. Papoulis, *Circuits and Systems: A Modern Approach,* New York: McGraw-Hill, 1980.
2. A. Papoulis, *The Fourier Integral and Its Applications,* New York: McGraw-Hill, 1962.
3. P. Henrici, *Applied and Computational Complex Analysis,* vols. 1–3, New York: Wiley, 1986.
4. C. Bender and S. Orszag, *Advanced Mathematical Methods for Scientists and Engineers,* New York: McGraw-Hill, 1978.
5. N. Bleistein and A. Handelsman, *Asymptotic Expansions of Integrals,* New York: Dover, 1986.
6. C. W. Helstrom, *Elements of Signal Detection and Estimation,* Upper Saddle River, NJ: Prentice-Hall, 1995.
7. C. W. Helstrom and J. A. Ritcey, Evaluating radar detection probabilities by steepest descent integration, *IEEE Trans. Aerosp. Electron. Syst.,* **AES-20**: 624–633, 1984.
8. C. Rivera and J. A. Ritcey, Error probabilities for QAM systems on partially coherent channels with intersymbol interference and crosstalk, *IEEE Trans. Commun.,* **46**: 775–783, 1998.

9. A. Papoulis, *Probability, Random Variables, and Stochastic Processes,* New York: McGraw-Hill, 1991.
10. W. Feller, *An Introduction to Probability Theory and Its Applications,* vols. 1 and 2, New York: Wiley, 1957.

Reading List

Erdelyi et al., *Higher Transcendental Functions,* New York: McGraw-Hill, 1953–55.

I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series, and Products,* rev. ed., New York: Academic Press, 1980.

D. Zwillinger, *Handbook of Integration,* Boston: Jones and Bartlett, 1992.

D. V. Widder, *The Laplace Transform,* Princeton, NJ: Princeton Univ. Press, 1946.

R. Bellman and K. L. Cooke, *Differential-Difference Equations,* New York: Academic Press, 1963.

H. Amindavar and J. A. Ritcey, Estimating density functions using Padé approximants, *IEEE Trans. Aerosp. Electron. Syst.,* **AES-30**: 416–424, 1994.

D. E. Chaveau, A. C. van Rooij, and R. H. Ruymgaart, Regularized inversion of noisy Laplace transforms, *Adv. Appl. Math.,* **15**: 186–201, 1994.

JAMES A. RITCEY
University of Washington

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Least-Squares Approximations
Standard Article

Zoran I. Mitrovski[1] and Petar M. Djuric[2]
[1]Morgan Stanley Dean Witter
[2]State University of New York

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2431
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (422K)

Abstract

The sections in this article are

Preliminaries

More Basic Concepts

Basic Linear-Algebra Concepts

Least-Squares Curve Fitting

The General Linear Least-Squares Problem

Solving the Linear Least-Squares Problem

Weighted Least Squares

Polynomial Fitting and Spline Interpolation

Nonlinear Least Squares

Sequential Least Squares

Predictive Least Squares

The Bootstrap Method

| | |

Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

LEAST-SQUARES APPROXIMATIONS

The method of least squares dates back about two hundred years. The main stimulus for its invention and development was provided by studies in astronomy. The first to publish on the method was the French mathematician Adrien-Marie Legendre, who studied the orbits of comets. It seems, however, that the first to invent the least-squares approximation method was the German mathematician Karl Friedrich Gauss, who used it in 1795 (Legendre discovered the method independently). Ever since, the method has been one of the basic tools for data analysis by scientists and engineers, and in its class it is the most popular.

In practice, data measured for studying physical phenomena are often erroneous due to the lack of absolutely accurate measurement devices. For example, when Legendre and Gauss studied the motion of planets, it was important to estimate their orbits accurately from imperfect observations. In general, mathematical models are built for observed phenomena, which are represented by functions with unknown parameters. These models have to approximate observed data as closely as possible, so the objective is to fit the model to the data in the best possible way. A criterion for goodness of fit for this purpose is often the total sum of squared residuals. The smaller the sum, the better the model. Clearly, then, for a given model it is important to estimate its parameters that yield the smallest sum of squared residuals; therefore we want to apply least-squares estimation methods.

In this article we discuss selected topics in the field of least-squares approximation. We start with preliminaries related to quantifying the approximations and definitions of the concept of norms. Then we introduce the notions of linear independence and inner product, and describe some basic linear-algebra concepts. The least-squares method is first explained with a couple of simple examples of curve fitting. It is followed by presentation of the general least-squares problem and approaches for solving it. Many important practical cases of linear least-squares involve polynomial fitting and spline interpolation; thus, we present some basic information for their use. Next, the focus of our attention shifts to nonlinear least-squares methods, and we address methods with reduced complexity as well as iterative techniques. The next topic, sequential least-squares estimation, involves implementation of the least-squares method recursively in time. The last two sections are on predictive least squares and the bootstrap method. The former is important for choosing the correct model from a set of candidates, and the latter, for assessing the accuracy of the least-squares estimates.

Preliminaries

In general, three main components are needed to specify every approximation problem: (1) the function $y(t)$, which is to be approximated over a given closed interval $[a,b]$, (2) the class Ψ of approximating functions $\psi(t)$, and (3) the norm $\|\cdot\|$ that gives some measure of magnitude of functions. The goal of best approximation is to find a function $\hat{\psi}$ such that

$$\|y(t) - \hat{\psi}(t)\| \leq \|y(t) - \psi(t)\| \qquad \text{for all} \quad \psi(t) \in \Psi \tag{1}$$

If the above is satisfied, the function $\hat{\psi}$ is the *best approximation* to $y(t)$ from the class Ψ with respect to the norm $\|\cdot\|$.

The class Ψ is called a *real linear space* if for any two functions $\psi_1(t),\psi_2(t) \in \Psi$, it also contains $\theta_1\psi_1(t) + \theta_2\psi_2(t)$ for any real θ_1 and θ_2. A linear space Ψ_p of finite dimension p can be defined given a set of (constituent) *basis functions* $h_i(t) \in \Psi$, $i = 1,2,\ldots, p$. Any function $\psi(t) \in \Psi_p$ can be represented as a linear combination of the basis functions $h_i(t)$:

$$\psi(t) = \sum_{i=1}^{p} \theta_i h_i(t) \tag{2}$$

for any real θ_i.

A typical example for a linear space is the set of polynomials of finite degree. They are very convenient for approximating functions over bounded-support domains. Any continuous function defined over a bounded support (or closed interval, in the single-variable case) can be approximated to any level of accuracy with a polynomial of sufficiently large degree (Weierstrass's theorem).

As mentioned previously, the norm is the third important component needed for specifying an approximation problem. It is the criterion that determines the goodness of each approximating function. A function that is a good approximant in one norm may turn out to be a bad approximant under a different norm. The following are some possible norms for a function $\varepsilon(t)$, defined over the finite closed interval $[a,b]$ (in the subsequent sections the function $\varepsilon(t)$ denotes the approximation error, and $w(t)$ is some nonnegative "weight function" defined on $[a,b]$):

$$L_\infty: \qquad \|\varepsilon\|_\infty = \max_{a \le t \le b} |\varepsilon(t)| \tag{3}$$

$$L_1: \qquad \|\varepsilon\|_1 = \int_a^b |\varepsilon(t)|\, dt \tag{4}$$

$$\text{weighted } L_1: \qquad \|\varepsilon\|_{1,w} = \int_a^b w(t)|\varepsilon(t)|\, dt \tag{5}$$

$$L_2 \text{ (least squares)}: \qquad \|\varepsilon\|_2 = \sqrt{\int_a^b |\varepsilon(t)|^2\, dt} \tag{6}$$

$$\text{weighted } L_2 \text{ (weighted least squares)}: \qquad \|\varepsilon\|_{2,w} = \sqrt{\int_a^b w(t)|\varepsilon(t)|^2\, dt} \tag{7}$$

The discrete versions of the above equations correspond to the situations involving a set of N distinct points $t_1, t_2, \ldots, t_N$ and a set of N nonnegative weight factors $w_1, w_2, \ldots, w_N$:

$$L_\infty: \qquad \|\varepsilon\|_\infty = \max_{1\le n\le N} |\varepsilon[t_n]| \tag{8}$$

$$L_1: \qquad \|\varepsilon\|_1 = \sum_{n=1}^{N} |\varepsilon[t_n]| \tag{9}$$

$$\text{weighted } L_1: \qquad \|\varepsilon\|_{1,w} = \sum_{n=1}^{N} w_n|\varepsilon[t_n]| \tag{10}$$

$$L_2 \text{ (least squares)}: \qquad \|\varepsilon\|_2 = \sqrt{\sum_{n=1}^{N} |\varepsilon[t_n]|^2} \tag{11}$$

$$\text{weighted } L_2 \text{ (weighted least squares)}: \qquad \|\varepsilon\|_{2,w} = \sqrt{\sum_{n=1}^{N} w_n|\varepsilon[t_n]|^2}. \tag{12}$$

In our presentation, we study continuous- and discrete-variable functions. We discriminate them in the notation by using $\xi(t)$ for the continuous case, and $\xi[t_n]$, or simply $\xi[n]$, for the discrete case, where $t \in [a,b] \subset \Re$ and $n = \{1,2,\ldots, N\}$. In the continuous case, the given (approximated) function $y(t)$ and the approximating functions $\psi(t)$ of the class Ψ must be defined on the interval $[a,b]$, so that the chosen norm $\|y - \psi\|$ is also defined on the same interval. Similarly, in the discrete case, $y[t_n]$, $\psi[t_n]$, and $\|y - \psi\|$ must be defined at the N distinct support points. If the best approximant $\hat{\psi}$ in the discrete case satisfies $\|y - \hat{\psi}\| = 0$, then $\hat{\psi}[t_n] = y[t_n]$ for $n = 1,2,\ldots, N$, in which case it is said that $\hat{\psi}$ *interpolates* y at the points t_n. This sort of approximation problem is called an *interpolation problem*.

More Basic Concepts

Linear Independence. The set of functions $h_i(t)$ is said to be *linearly independent* on a given support S_t if

$$\sum_{i=1}^{p} \theta_i h_i(t) \equiv 0 \quad \text{for all } t \in S_t \quad \text{implies that} \quad \theta_i = 0, \quad i = 1, 2, \cdots, p$$

As an example, the set of functions $h_i(t) = t^{i-1}$, $i = 1,2,\ldots, p$, is linearly independent on the support $S_t = [a,b]$ where a and b are real and $a < b$. If, however, the support is $S_t = \{t_1, t_2, \ldots, t_N\}$, then the set is linearly

independent if and ony if $N \geq p$, because otherwise the sum of squared approximation errors could be made equal to zero at the N support points without necessarily implying that the coefficients θ_i are zero.

Inner Product. An *inner product* of two (real) functions $h_1(t)$ and $h_2(t)$ whose $\boldsymbol{L}_2$ norms are finite is defined as

$$\langle h_1, h_2 \rangle = \int_a^b h_1(t) h_2(t)\, dt \tag{13}$$

In the discrete case associated with the set of points $\{t_1, t_2, \ldots, t_N\}$ the inner product will be defined as

$$\langle h_1, h_2 \rangle = \sum_{n=1}^{N} h_1[t_n] h_2[t_n] \tag{14}$$

Note that the least-squares ($\boldsymbol{L}_2$) norm (squared) of a given function is simply its inner product with itself, i.e., $\|\varepsilon\|^2{}_2 = \langle \varepsilon, \varepsilon \rangle$.

Two functions $h_1(t)$ and $h_2(t)$ are said to be *orthogonal* if $\langle h_1, h_2 \rangle = 0$. An *orthogonal system* is defined as a set of functions $\{h_i(t)\}$, $i = 1, 2, \ldots, p$, that satisfy

$$\langle h_i, h_j \rangle = 0 \text{ if } i \neq j \text{ and } h_k(t) \neq 0 \text{ on the support } S_t, \qquad i, j, k = 1, 2, \cdots, p$$

It can be shown that every orthogonal system is linearly independent on the support S_t. An orthogonal system is called *orthonormal* if $\langle h_i, h_i \rangle = 1$, $i = 1, 2, \ldots, p$.

Basic Linear-Algebra Concepts

Every function can be represented by a vector of some sort. In the discrete case, this can be a vector of samples of the function taken at distinct values of t. A polynomial function can be represented by a vector of its polynomial coefficients. A periodic function can be represented by its Fourier series coefficients. The field of linear algebra covers the concepts associated with vectors and *vector spaces*. In this section we will cover a few basic notions. Almost regularly, the vector–matrix (discrete-valued) concepts have their equivalents in concepts associated with functions. These equivalencies can usually be established by simply substituting the word "vector" for "function." For instance, a set of vectors $\boldsymbol{h}_i$, $i = 1, 2, \ldots, p$, is said to be linearly independent if

$$\sum_{i=1}^{p} \theta_i \boldsymbol{h}_i = \boldsymbol{0}$$

holds only with $\theta_1 = \theta_2 = \cdots = \theta_p = 0$.

The set of all N-dimensional vectors is an *N-dimensional vector space*. Equivalently to the concept of linear (function) spaces described previously, if $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are members of this vector space, then so are $\boldsymbol{h}_1 + \boldsymbol{h}_2$ and $\theta\, \boldsymbol{h}_1$. (Note that there is no need for using "linear" in the case of vector spaces.) We call these two conditions *closure under vector addition and scalar–vector multiplication*. If a subset P of a vector space $\boldsymbol{Q}$ is closed under vector addition and scalar–vector multiplication, then P is called a *subspace*. The maximal number of linearly independent vectors in P is called the *dimension* of the subset P. A maximal-size set of linearly independent vectors in a subspace P is a *basis* for P. Given a p-dimensional subspace P and a set

of $k < p$ linearly independent member vectors, there are always $p - k$ additional vectors in P such that the concatenated set of p vectors represents a basis for P. If a set of vectors $\boldsymbol{h}_i$, $i = 1,2,\ldots, p$, constitutes a basis for the subspace P, then any vector $\boldsymbol{u} \in P$ can be represented as $\boldsymbol{u} = \sum_{i=1}^{p} \theta_i \boldsymbol{h}_i$. The concepts of vector norms, vector inner products, and orthogonal (and orthonormal) vectors, can be defined analogously to the same concepts described previously in the case of functions.

The subspace formed by the set of all linear combinations of the vectors $\boldsymbol{h}_i$, $i = 1, 2, \ldots, k$, is called the *span* of this vector set. The dimension of this subspace is less than or equal to k. Given an $N \times p$ matrix $\boldsymbol{H}$, the subspace spanned by its column vectors is called the *range* or the *column space*, and the subspace spanned by its row vectors is called the *row space*. For any matrix, the dimension of the row space is equal to the dimension of the column space, and this number is called the *rank* of the matrix. An $N \times p$ matrix $\boldsymbol{H}$ is of *full rank* if rank $(\boldsymbol{H}) = \min(N, p)$, and it is *rank-deficient* if rank $(\boldsymbol{H}) < \min(N, p)$. A square $N \times N$ matrix is *singular* if rank $(\boldsymbol{H}) < N$, and *nonsingular* if rank $(\boldsymbol{H}) = N$. A square matrix $\boldsymbol{Q}$ is *orthogonal* if $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. Matrices with this property have extensive use in many approaches to solving least-squares problems. Lawson and Hanson (1) give a succinct and clear description of more linear-algebra concepts pertaining to the least-squares problem and its solution.

Least-Squares Curve Fitting

We continue our discussion of the least-squares problem by introducing some of its simplest and clearest manifestations as applied to the curve-fitting problem.

Straight-Line Fitting. Experiments in science and engineering produce data points that are subsequently used to derive relationships between variables in the observed models. In particular, a set of distinct data points $(t_1, y[1])$, $(t_2, y[2]),\ldots, (t_N, y[N])$ needs to be *fitted* with a function $f(t)$ that relates these two variables. Depending on our confidence in the exactness of the measured data points, we may approach this problem in two main ways. If the data points are known to be highly accurate (i.e., the measuring devices add little or no errors that are not accounted for in the model, or the level of external noise in the system is insignificant), then interpolation is the best way to go.

Interpolation will attempt to fit a curve that goes straight through all of the data points. If, however, the data points are known to be insufficiently accurate, interpolation will almost regularly give unsatisfactory results. Intuitively, attempting to fit a curve through data points that are likely to be randomly positioned around their "accurate" positions is bound to produce overly complex approximating functions that poorly describe the real underlying phenomenon. In these cases, the true value of $f(t_n)$ satisfies $f(t_n) = y[n] + \varepsilon[n]$, where $\varepsilon[n]$ denotes the measurement *error* $\varepsilon[n] = f(t_n) - y[n]$, which is also called the *residual*. Each of the norms listed previously under the discrete case can be associated with this residual to serve as a quality measure for the fit. Least-squares methods exploit the $\boldsymbol{L}_2$ norm.

First, we cover the simplest case of linear approximation. How do we find a best approximation of form $f(t) = \theta_1 t + \theta_2$ that goes near a set of data points $(t_1, y[1])$, $(t_2, y[2]),\ldots, (t_N, y[N])$ scattered in the (t, y) two-dimensional space? Our goal is to position this line "as close as possible" to all data points contained in the set. For convenience, our measuring stick for goodness will be the square of the L_2 norm of the residual,

$$\|\varepsilon\|^2 = \sum_{n=1}^{N} |\varepsilon[n]|^2 = \sum_{n=1}^{N} \big[f(t_n) - y[n]\big]^2 = \sum_{n=1}^{N} (\theta_1 t_n + \theta_2 - y[n])^2 = J(\theta_1, \theta_2). \quad (15)$$

The best linear approximation (i.e., the best straight line) is the one whose parameter pair (θ_1, θ_2) minimizes the error function $J(\theta_1, \theta_2)$; it is denoted by $(\hat{\theta}_1, \hat{\theta}_2)$. Hence, the approximation problem is transformed

into a minimization problem in the parameter space spanned by the parameters θ_1 and θ_2. At the point that minimizes the value of $J(\theta_1,\theta_2)$, the partial derivatives $\partial J/\partial\theta_1$ and $\partial J/\partial\theta_2$ are both zero:

$$\frac{\partial J(\theta_1,\theta_2)}{\partial\theta_1}=\sum_{n=1}^{N}2(\theta_1 t_n+\theta_2-y[t_n])t_n=2\sum_{n=1}^{N}\left(\theta_1 t_n^2+\theta_2 t_n-y[t_n]t_n\right)=0 \quad (16)$$

$$\frac{\partial J(\theta_1,\theta_2)}{\partial\theta_1}=\sum_{n=1}^{N}2(\theta_1 t_n+\theta_2-y[t_n])=2\sum_{n=1}^{N}\left(\theta_1 t_n+\theta_2-y[t_n]\right)=0 \quad (17)$$

The above equations can be arranged into a system of two equations with two unknowns, which—in the context of least-squares approximations—are referred to as the *normal equations*:

$$\left(\sum_{n=1}^{N}t_n^2\right)\theta_1+\left(\sum_{n=1}^{N}t_n\right)\theta_2=\sum_{n=1}^{N}t_n y[t_n] \quad (18)$$

$$\left(\sum_{n=1}^{N}t_n\right)\theta_1+N\theta_2=\sum_{n=1}^{N}y[t_n] \quad (19)$$

The solution to this system is given by

$$\hat{\theta}_1=\frac{N\sum t_n y[t_n]-\sum t_n\sum y[t_n]}{N\sum t_n^2-\left(\sum t_n\right)^2} \quad (20)$$

$$\hat{\theta}_2=\frac{\sum t_n^2\sum y[t_n]-\sum t_n\sum t_n y[t_n]}{N\sum t_n^2-\left(\sum t_n\right)^2} \quad (21)$$

where for simplicity $\Sigma(\cdot)$ denotes $\sum_{n=1}^{N}$.

Nonlinear Fitting Functions. The same least-squares method used to fit a straight line through a set of data points can be extended to nonlinear cases as well. For instance, consider the function $f(t)=\theta t^c$, where c is some known constant. Given a set of N data points and following the same least-squares method, we need to find a value of the parameter θ that minimizes the function

$$J(\theta)=\sum_{n=1}^{N}\left(\theta t_n^c-y[t_n]\right)^2 \quad (22)$$

Here we need to solve $\partial J/\partial\theta = 2\sum_{n=1}^{N}(\theta t^{2c}{}_n - t^c{}_n y[t_n]) = 0$, which yields the solution

$$\hat{\theta} = \frac{\sum_{n=1}^{N} t_n^c y[t_n]}{\sum_{n=1}^{N} t_n^{2c}} \tag{23}$$

One familiar example ($c = 2$) covered by this power fit is finding the acceleration of gravity from a set of time and distance measurements.

The General Linear Least-Squares Problem

As a generalization of the above curve-fitting discussions, the linear least-squares problem can be presented as follows. We need to model a function $y(t)$ over a given interval with an optimal (best in the least-squares sense) linear combination of p known basis functions $h_i(t)$, $i = 1, 2, \ldots, p$:

$$y(t) = \sum_{i=1}^{p} \theta_i h_i(t) + \varepsilon(t) \tag{24}$$

where $\varepsilon(t)$ is the modeling (fitting) error and θ_i are the unknown modeling coefficients. In the straight-line fitting discussion, we used two basis functions (a linear and a constant function), while in the nonlinear curve-fitting discussion we used a single (nonlinear) basis function. Regardless of whether the basis functions themselves are linear or nonlinear, the linearity of the least-squares procedure stems from the fact that the basis functions are elements of a linear space, i.e., the model function in Eq. (24) is linear in the unknown coefficients θ_i.

In the discrete case where the functions are only known at N distinct points, the linear least-squares problem is as follows:

$$\begin{aligned} y[1] &= \sum_{i=1}^{p} \theta_i h_i[1] + \varepsilon[1] \\ y[2] &= \sum_{i=1}^{p} \theta_i h_i[2] + \varepsilon[2] \\ &\vdots \\ y[N] &= \sum_{i=1}^{p} \theta_i h_i[N] + \varepsilon[N] \end{aligned} \tag{25}$$

that is, $y[n] = \Sigma^{p}{}_{i=1}\, \theta_i h_i[n] + \varepsilon[n]$ for $n = 1, 2, \ldots, N$. In matrix form this can be written as

$$\begin{bmatrix} y[1] \\ y[2] \\ \vdots \\ y[N] \end{bmatrix} = \begin{bmatrix} h_1[1] & h_2[1] & \cdots & h_p[1] \\ h_1[2] & h_2[2] & \cdots & h_p[2] \\ \vdots & \vdots & \ddots & \\ h_1[N] & h_2[N] & \cdots & h_p[N] \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix} + \begin{bmatrix} \varepsilon[1] \\ \varepsilon[2] \\ \vdots \\ \varepsilon[N] \end{bmatrix} \tag{26}$$

or

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{\varepsilon} \tag{27}$$

where $\boldsymbol{H}$ is an $N \times p$ matrix and the compositions of the vectors and matrices involved are obvious from the above. This data-modeling problem corresponds to selecting the basis coefficients so that the data model best represents the measured data in a least-squares sense.

We seek the vector θ that minimizes the square of the L_2 norm of the criterion (modeling error)

$$\hat{\theta} = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\theta}\|^2 \tag{28}$$

In nonmatrix form, the function to be minimized is

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} \left(y[n] - \sum_{i=1}^{p} \theta_i h_i[n] \right)^2 \tag{29}$$

After taking the partial derivatives $\partial J(\theta)/\partial\theta_i$, $i = 1, 2, \ldots, p$, and setting them equal to zero, we get the following system of equations:

$$\sum_{n=1}^{N} \left(y[n] - \sum_{i=1}^{p} \theta_i h_i[n] \right) h_l[n] = 0 \qquad \text{for} \quad l = 1, 2, \ldots, p \tag{30}$$

Notice the introduction of a new index l, which is not to be confused with the index i. After interchanging the order of summations, we get p (linear) normal equations with p unknown coefficients θ_i:

$$\sum_{i=1}^{p} \left(\sum_{n=1}^{N} h_l[n] h_i[n] \right) \theta_i = \sum_{n=1}^{N} h_l[n] y[n] \qquad \text{for} \quad l = 1, 2, \cdots, p \tag{31}$$

Equivalently, the matrix form of the normal equations can be presented as $(\boldsymbol{H}^{\mathrm{T}}\,\boldsymbol{H})\theta = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{y}$, where $\boldsymbol{H}^{\mathrm{T}}$ denotes the transpose of the matrix $\boldsymbol{H}$. The solution for the unknown coefficient vector θ is

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{y} \tag{32}$$

which is referred to as a least-squares solution. Without going into derivations, it should be noted here that in the case of complex data modeling (where we allow for complex-valued elements of θ, $\boldsymbol{H}$, and/or $\boldsymbol{y}$), the above formula becomes

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{H}^{*}\boldsymbol{H})^{-1}\boldsymbol{H}^{*}\boldsymbol{y} \tag{33}$$

where $\boldsymbol{H}^{*}$ denotes the *complex conjugate transpose* of $\boldsymbol{H}$.

Note that the solutions yielded by Eqs. (32) and (33) are by no means unique. (They are unique if and only if $\boldsymbol{H}$ is of full rank.) It is possible that there exists a whole set of least-squares solutions that are associated with the same (and unique) minimal (least-squares) value.

Solving the Linear Least-Squares Problem

Orthogonal Decomposition. The solutions presented in Eqs. (32) and (33) are oftentimes computationally costly (because of high-order matrix inversions) and/or very sensitive to small perturbations in the observed model. There are several ways to reach a least-squares solution more efficiently. Some of them are direct, and some of them are iterative. Most of them take advantage of the important property of orthogonal matrices that they preserve the L_2 norm under multiplication. In other words, given an $N \times N$ orthogonal matrix $\boldsymbol{Q}$ and the N-vector ε,

$$\|\boldsymbol{Q}\,\varepsilon\| = \|\varepsilon\|, \qquad \text{or equivalently,} \quad \|\boldsymbol{Q}\,(\boldsymbol{y} - \boldsymbol{H}\,\boldsymbol{\theta})\| = \|(\boldsymbol{y} - \boldsymbol{H}\,\boldsymbol{\theta})\| \tag{34}$$

Using this property, we can present the least-squares problem in a modified form. Following Lawson and Hanson (1), we first assume that the $N \times p$ matrix $\boldsymbol{H}$ ($N \geq p$) is of rank k and that it can be decomposed as

$$\boldsymbol{H} = \boldsymbol{A}\boldsymbol{R}\boldsymbol{B}^{\mathrm{T}} \tag{35}$$

where $\boldsymbol{A}$ is an orthogonal $N \times N$ matrix, $\boldsymbol{B}$ is an orthogonal $p \times p$ matrix, and $\boldsymbol{R}$ is of the form

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{36}$$

where $\boldsymbol{R}_{11}$ is a $k \times k$ matrix of rank k. Next, we introduce the new N-vector

$$\boldsymbol{u} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix} \begin{matrix} \text{size } k \\ \text{size } N-k \end{matrix} \tag{37}$$

and the new p-vector

$$\lambda = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{\theta} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \begin{matrix} \text{size } k \\ \text{size } p-k \end{matrix} \tag{38}$$

We define $\hat{\lambda}_1$ to be the unique solution of

$$\boldsymbol{R}_{11}\lambda_1 = \boldsymbol{u}_1 \tag{39}$$

Then:

(1) All solutions to the least-squares problem of minimizing $\|\boldsymbol{y} - \boldsymbol{H}\theta\|$ can be presented as

$$\hat{\theta} = \boldsymbol{B}\begin{bmatrix} \hat{\lambda}_1 \\ \lambda_2 \end{bmatrix}, \qquad \text{where } \lambda_2 \text{ is any arbitrary } (p-k)\text{-vector} \tag{40}$$

(2) Any $\hat{\theta}$ so defined is associated with the same residual (error) vector

$$\varepsilon = \boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{\theta}} = \boldsymbol{A}\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{u}_2\end{bmatrix} \tag{41}$$

(3) The norm of ε satisfies

$$\|\varepsilon\| = \|\boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{\theta}}\| = \|\boldsymbol{u}_2\| \tag{42}$$

(4) The unique *solution of minimum length* is

$$\hat{\boldsymbol{\theta}} = \boldsymbol{B}\begin{bmatrix}\hat{\lambda}_1\\ \boldsymbol{0}\end{bmatrix} \tag{43}$$

The proofs of the above four statements can be found in Lawson and Hanson (1). The decomposition $\boldsymbol{H} = \boldsymbol{ARB}^{\mathrm{T}}$ is called an *orthogonal decomposition* of $\boldsymbol{H}$, and it is by no means unique. Some of the most widely known and applied orthogonal decompositions are the *QR decomposition* and the *singular value decomposition* (*SVD*). The *discrete Fourier transform* can also be applied. In general, regardless of the orthogonal decomposition employed, every least-squares problem has a *unique solution of minimum length*, a *unique set of all solutions*, and a *unique minimum residual value*. In the special case when rank $(\boldsymbol{H}) \equiv k = p$, the solution to the least-squares problem is itself unique.

Weighted Least Squares

Oftentimes, in practice, it is desirable to attach separate weights to each of the terms in the norm summation for the modeling error. In the continuous case, a given continuous weighting function may be associated with the norm formula. Cases where this is needed are the ones where each data point is known to be associated with a different level of certainty, accuracy, or reliability. Intuitively, we would like the members of the error-norm sum stemming from more reliable data to make larger contributions to the total. This is where weighted least squares comes into play. In the more general complex data-modeling case, the error function to be minimized is

$$J(\boldsymbol{\theta}) = (\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\theta})^{*}\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{\theta}) \tag{44}$$

where the $N \times N$ weighting matrix $\boldsymbol{W}$ is positive definite and Hermitian (i.e., $\boldsymbol{W}* = \boldsymbol{W}$). If $\boldsymbol{W}$ is equal to the identity matrix, the model reduces to the classical (unweighted) least-squares case. Most often, $\boldsymbol{W}$ is diagonal with the weighting coefficients populating the main diagonal. Skipping the rather straightforward derivation (which is readily found in numerical analysis and optimization-related textbooks) the solution to the weighted least-squares problem is

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{H}^{*}\boldsymbol{W}\boldsymbol{H})^{-1}\boldsymbol{H}^{*}\boldsymbol{W}\boldsymbol{y} \tag{45}$$

Polynomial Fitting and Spline Interpolation

After analyzing the general least-squares problem and discussing some solution approaches, we return to some more complex manifestations of the curve-fitting problem such as polynomial fitting and spline interpolation.

One convenient way of modeling empirical data is by fitting a polynomial of degree p through N data points $(t_n, y[t_n])$, n = 1, 2, ..., N. We have already presented a special case when we addressed the fitting of a quadratic function through a set of data points. The linear least-squares approach described above can be implemented towards the solution of the generalized polynomial fitting problem by using the basis functions $h_i(t) = t^i$, $i = 0, 1, \ldots, p$. The approximating function is

$$\psi(t) = \sum_{i=0}^{p} \theta_i h_i(t) = \theta_0 + \theta_1 t + \theta_2 t^2 + \cdots + \theta_p t^p \qquad (46)$$

Note that there are $p + 1$ unknown coefficients. Of course, all the previous discussion for the general least-squares problem and its solution applies to this special case.

If the number of data points, N, is less than or equal to the number of unknown polynomial coefficients, $p + 1$, there will always exist a polynomial curve (of degree p) that can bring the modeling error down to zero (i.e., it will manage to go straight through the data points). This special subcase of polynomial fitting is called *polynomial interpolation*.

There are difficulties associated with polynomial fitting, and much attention needs to be paid to the nature of the modeled phenomenon and the empirical data. Unless we are sure that the data points actually lie on the polynomial curve (polynomial interpolation problem) and unless *a priori* knowledge is available for the polynomial degree of the phenomenon, polynomial fitting can often lead to unsatisfactory solutions. Note that a polynomial of degree p can have $p - 1$ local extrema. This means that the least-squares polynomial will have more oscillations as p increases. The modeling errors (measured at the data points) may still be zero or very small (or, at least, be minimal in a least-squares sense), but that fact itself will not guarantee "nice" behavior of the polynomial curve in the space between the data points.

Example: As an illustration of the above discussion, Fig. 1 shows a polynomial fitting ($p + 1 < N$) example, while Fig. 2 shows a polynomial interpolation ($p + 1 \geq N$) example. All polynomial curves shown are least-squares solutions for a specific case where the number of data points is $N = 5$. The data points are $\{(t_n, y[t_n])\} = \{(1,1), (2,3), (3,2), (4,5), (5,7)\}$. Figure 1 shows the $1 \leq p \leq 3$ (fitting) cases, and Fig. 2 the $4 \leq p \leq 6$ (interpolation) cases, where p is the degree of the polynomial. Note how the magnitude of the oscillations increases with p. A remedy for this so-called *polynomial wiggle* phenomenon can be found in *spline interpolation*.

The spline interpolation approach tries to piece together lower-degree polynomial curves $\psi_k(t)$, each of which interpolates the data over predetermined abscissa segments. The combined curve $\psi(t)$, defined over the whole relevant abscissa range, is called a *spline*. The connection points between the segments are called *knots*. The simplest and most trivial spline interpolation case is when the polynomials are linear (i.e., of degree 1). This amounts to simply connecting adjacent data points with straight lines. Most widely used, especially in the computer graphics industry, are the *cubic splines*. They are smooth interpolating curves without excessive oscillations. Intuitively, they are a good choice because they are the lowest-degree polynomials with nonzero first and second derivatives. A low polynomial degree minimizes unwanted oscillations (wiggles) while the (existing, hence controllable) first and second derivatives enforce the desirable behavior around the knots.

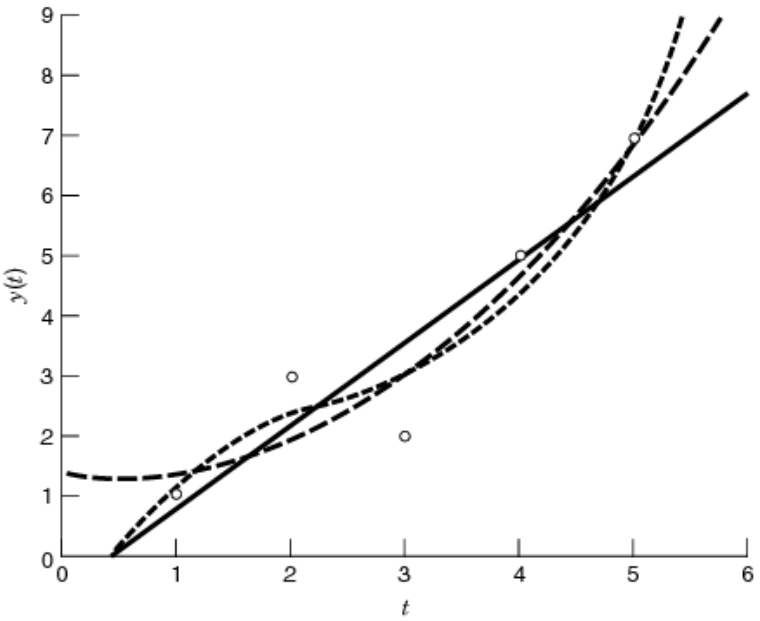


Fig. 1. Polynomial fitting example ($p + 1 < N$). The $N = 5$ data points are $\{(t_k, y_k)\} = \{(1,1), (2,3), (3,2), (4,5), (5,7)\}$. Full line ($p = 1$) represents $y(t) = 1.4000t - 0.6000$. Dashed line ($p = 2$) represents $y(t) = 0.2857t^2 - 0.3143t + 1.4000$. Dotted line ($p = 3$) represents $y(t) = 0.1667t^3 - 1.2143t^2 + 3.6190t + 1.4000$.

Consider $N + 1$ knots $(t_0, y[t_0])$, $(t_1, y[t_1])$,..., $(t_N, y[t_N])$ such that $t_0 < t_1 \cdots < t_N$. Also consider N cubic polynomials

$$\psi_n(t) = c_{n,0} + c_{n,1}(t - t_n) + c_{n,2}(t - t_n)^2 + c_{n,3}(t - t_n)^3 \qquad (47)$$

for $t \in [t_n, t_{n+1}]$ and $n = 0, 1, \ldots, N - 1$. A cubic spline $\psi(t)$ will be formed if the following four properties are satisfied:

$$\psi(t_n) = y[t_n] \qquad \text{for} \quad n = 0, 1, \ldots, N-1 \qquad (48)$$

$$\psi_n(t_{n+1}) = \psi_{n+1}(t_{n+1}) \qquad \text{for} \quad n = 0, 1, \ldots, N-2 \qquad (49)$$

$$\left.\frac{d\psi_n(t)}{dt}\right|_{t=t_{n+1}} = \left.\frac{d\psi_{n+1}(t)}{dt}\right|_{t=t_{n+1}} \qquad \text{for} \quad n = 0, 1, \ldots, N-2 \qquad (50)$$

$$\left.\frac{d^2\psi_n(t)}{dt^2}\right|_{t=t_{n+1}} = \left.\frac{d^2\psi_{n+1}(t)}{dt^2}\right|_{t=t_{n+1}} \qquad \text{for} \quad n = 0, 1, \ldots, N-2 \qquad (51)$$

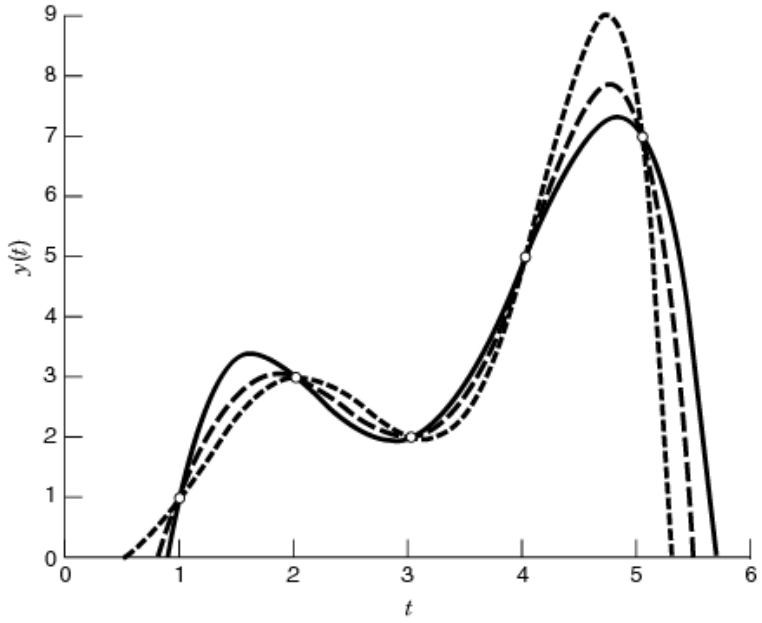


Fig. 2. Polynomial interpolation example ($p + 1 \geq N$). The data points are the same as in Fig. 1. Full line ($p = 4$) represents $y(t) = -0.5000t^4 + 6.1667t^3 - 26.0000t^2 + 44.3333t - 23.0000$. Dashed line ($p = 5$) represents $y(t) = -0.1618t^5 + 1.9270t^4 - 7.5864t^3 + 10.4051t^2 - 3.5839$. Dotted line ($p = 6$) represents $y(t) = -0.0625t^6 + 0.7458t^5 - 2.9375t^4 + 3.9375t^3 - 0.6833t$.

The property (48) ensures that the spline passes through the knots (data points). The property (49) ensures that the spline is a continuous function. The property (50) guarantees that the spline is smooth around the knots. The property (51) further limits the curvature of the spline around the knots.

The goal of the spline interpolation procedure is to find the set of cubic polynomials that satisfy the above properties. Each of the N cubic polynomials $\psi_n(t)$ has four coefficients, which results in a total of $4N$ unknowns. The property (48) provides $N + 1$ equations. Each of the properties (49), (50), and (51) provides $N - 1$ conditions, bringing the total to $N + 1 + 3(N - 1) = 4N - 2$ equations. Two more conditions can be added to control the spline's derivatives at the endpoints $(t_0, y[t_0])$ and $(t_N, y[t_N])$, thus bringing the number of equations to $4N$, which equals the number of unknown coefficients. Obviously, different endpoint conditions will produce correspondingly different end-segment polynomials. For an in-depth coverage of splines see Dierckx (2) and the bibliography therein.

Nonlinear Least Squares

In many problems the data $\boldsymbol{y}$ are modeled as a nonlinear function of unknown parameters θ:

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{\theta}) + \boldsymbol{\varepsilon} \tag{52}$$

where $\boldsymbol{y}$ is an $N \times 1$ vector of observed samples, $\boldsymbol{g}(\cdot)$ is a nonlinear function in the parameters θ, and ε is an $N \times 1$ vector of errors. The parameter vector belongs to the parameter space $\Theta \subset \Re^p$; that is θ is a p-dimensional

vector. The expression in Eq. (52) can also be written as

$$y[n] = g(\boldsymbol{\theta}, n) + \varepsilon[n] \qquad \text{for} \quad n = 1, 2, \ldots, N \tag{53}$$

and the least-squares estimate of θ, $\hat{\theta}$, is obtained by minimizing the error sum of squares

$$J(\boldsymbol{\theta}) = (\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{\theta}))^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{\theta})) \tag{54}$$

or

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} (y[n] - g(\boldsymbol{\theta}, n))^2 \tag{55}$$

over $\theta \in \Theta$, i.e.,

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \tag{56}$$

As opposed to the linear least-squares problem, the estimation of θ when $\boldsymbol{g}(\cdot)$ is a nonlinear function may be very difficult. The general methods for finding $\hat{\theta}$ are based on iterative numerical techniques, of which the best known are the Gauss–Newton method and the Newton–Raphson algorithm. Before we proceed with their description, it is important to comment on approaches that reduce the complexity of the problem and allow for easier implementation of the least-squares estimation.

Methods with Reduced Complexity. In some situations it is possible to transform the original nonlinear problem to a linear one by using a one-to-one transformation. In that case, the original parameters θ are transformed to α via

$$\boldsymbol{\alpha} = \boldsymbol{q}(\boldsymbol{\theta}) \tag{57}$$

where $\boldsymbol{q}(\cdot)$ is the transformation, so that

$$\boldsymbol{g}(\boldsymbol{\theta}(\boldsymbol{\alpha})) = \boldsymbol{g}(q^{-1}(\boldsymbol{\alpha})) = \boldsymbol{H}\,\boldsymbol{\alpha} \tag{58}$$

Then one can apply first the linear least-squares approach to estimate $\hat{\alpha}$ and follow it with the transformation

$$\hat{\boldsymbol{\theta}} = \boldsymbol{q}^{-1}(\hat{\boldsymbol{\alpha}}) \tag{59}$$

to obtain the desired estimates. Unfortunately, it is usually difficult to find a function $\boldsymbol{q}(\cdot)$ that converts the nonlinear problem to a linear one. To demonstrate the method, we provide an example that is of interest in many signal-processing applications, as presented in Kay (3).

Example: Let the data $y[n]$ be modeled as a sinusoid with known frequency f and unknown amplitude A and phase φ:

$$y[n] = A \sin(2\pi f n + \varphi) + \varepsilon[n] \qquad \text{for} \quad n = 1, 2, \ldots, N \tag{60}$$

and let the objective be to find the least-squares estimates of A and φ. Obviously, one of the parameters, φ, is nonlinear, which means that the straightforward approach would be to apply a nonlinear least-squares method. The alternative is to transform A and φ to a new set of parameters B_1 and B_2, which appear in the model of the data as linear parameters. The transformation is given by

$$B_1 = A\cos\varphi \tag{61}$$

$$B_2 = A\sin\varphi \tag{62}$$

and the model becomes

$$y[n] = B_1\,\sin(2\pi fn) + B_2\,\cos(2\pi fn) + \varepsilon[n] \qquad \text{for} \quad n = 1, 2, \ldots, N \tag{63}$$

The parameters B_1 and B_2 can now easily be estimated, and once they are obtained, the original parameters A and φ are found from

$$\hat{A} = \hat{B}_1^2 + \hat{B}_2^2 \tag{64}$$

$$\hat{\varphi} = \arctan\left(\frac{\hat{B}_2}{\hat{B}_1}\right) \tag{65}$$

Another approach that may reduce the complexity of the problem is based on the concept of separability, as described by Seber and Wild (4). Namely, in some problems the parameters θ can be partitioned, $\theta = [\beta, \gamma]$, so that the minimization of the criterion $J\,(\beta, \gamma)$ with respect to β is easy. The idea of the approach is to estimate β first and then proceed with estimating γ by minimizing $J\,(\,\hat{\gamma}, \gamma)$. For instance, when the dimension of θ is p and the dimensions of β and γ are p_β and p_γ, respectively, it is clear that the dimension of the parameter space over which a nonlinear least-squares procedure has to be applied is reduced from p to p_γ.

Example: The data represent a sinusoid as in Eq. (63). Let the unknowns be the amplitudes of the quadrature components B_1 and B_2 as well as the frequency f. The goal is to estimate the unknown parameters from the data $\boldsymbol{y}$. It is not difficult to see that Eq. (63) can be rewritten in vector–matrix form as

$$\boldsymbol{y} = \boldsymbol{H}\,(f)\beta + \boldsymbol{\varepsilon} \tag{66}$$

where $\boldsymbol{y}$ and ε are $\boldsymbol{N} \times 1$ vectors, $\boldsymbol{H}\,(f)$ is an $N \times 2$ matrix whose n-th row $\boldsymbol{h}^{\mathrm{T}}{}_n = [\sin(2\pi fn)\ \cos(2\pi fn)]$, and $\beta^{\mathrm{T}} = [B_1\ B_2]$. The unknown parameters $\theta^{\mathrm{T}} = [B_1 \quad B_2 \quad f]$ are thus split to β and $\gamma = [f]$. For given f, the parameters β can easily be estimated from

$$\hat{\beta} = \left(\boldsymbol{H}^{\mathrm{T}}(f)\boldsymbol{H}\,(f)\right)^{-1}\boldsymbol{H}^{\mathrm{T}}(f)\boldsymbol{y}. \tag{67}$$

If this result is plugged into the minimization of J, the estimate of f is obtained from

$$\hat{f} = \arg\min_{f} (\boldsymbol{y} - \boldsymbol{H}(f)\hat{\beta})^{\mathrm{T}} (\boldsymbol{y} - \boldsymbol{H}(f)\hat{\beta}) \tag{68}$$

which, after simple algebra, can be rewritten as

$$\hat{f} = \arg\max_{f} \boldsymbol{y}^{\mathrm{T}} \boldsymbol{H}(f) \left(\boldsymbol{H}^{\mathrm{T}}(f)\boldsymbol{H}(f)\right)^{-1} \boldsymbol{H}^{\mathrm{T}}(f)\boldsymbol{y} \tag{69}$$

Thus, the estimation of the unknowns is implemented as follows: first the least-squares estimate of $\hat{f}$ is obtained from Eq. (69), and then the estimate of β according to

$$\hat{\beta} = \left(\boldsymbol{H}^{\mathrm{T}}(\hat{f})\boldsymbol{H}(\hat{f})\right)^{-1} \boldsymbol{H}^{\mathrm{T}}(\hat{f})\boldsymbol{y} \tag{70}$$

A rigorous treatment of problems where separability is possible can be found in Golub and Pereira (5).

Numerical methods. When the above methods fail, one usually resorts to numerical iterative techniques. There are two general iterative approaches; one is known as the Gauss–Newton, and the other, as the Newton–Raphson method. Suppose that the process of estimating θ is iterative and that at iteration k, $\theta^{(k)}$ is the estimate of θ. If $\theta^{(k)}$ is close enough to θ, one can expand $\boldsymbol{g}(\theta)$ around $\theta^{(k)}$ using linear Taylor expansion. The result is

$$\boldsymbol{g}(\boldsymbol{\theta}) \approx \boldsymbol{g}\left(\boldsymbol{\theta}^{(k)}\right) + \boldsymbol{G}^{(k)}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(k)}\right) \tag{71}$$

where $\boldsymbol{G}^{(k)}$ is an $N \times p$ matrix whose elements are

$$\left[\boldsymbol{G}^{(k)}\right]_{i,j} = \frac{\partial g_i}{\partial \theta_j} \tag{72}$$

and where θ is substituted by $\theta^{(k)}$ to evaluate the partials, and g_i and θ_j denote the ith and jth elements of $\boldsymbol{g}$ and θ, respectively. Recall now that the least-squares estimate is the value of θ that minimizes $J = \varepsilon^{\mathrm{T}}\varepsilon$. If in

$$\varepsilon = \boldsymbol{y} - \boldsymbol{g}(\boldsymbol{\theta}) \tag{73}$$

we substitute the approximated $\boldsymbol{g}(\theta)$ in Eq. (71), and with it we form the approximation of the criterion J, its minimization with respect to θ becomes easy, because then θ appears as a set of linear parameters. This estimate of θ is denoted as $\theta^{(k+1)}$ and is given by

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \left(\boldsymbol{G}^{(k)\mathrm{T}}\boldsymbol{G}^{(k)}\right)^{-1} \boldsymbol{G}^{(k)\mathrm{T}}\left(\boldsymbol{y} - \boldsymbol{g}\left(\boldsymbol{\theta}^{(k)}\right)\right) \tag{74}$$

Once $\theta^{(k+1)}$ is evaluated, it is used to compute $\boldsymbol{G}^{(k+1)}$, and the equation (74) is applied to determine $\theta^{(k+1)}$. The iterations continue until a predefined criterion of convergence is satisfied. This algorithm is known as the Gauss–Newton algorithm, and as $k \to \infty$, under fairly general assumptions on $\boldsymbol{g}$, it converges to the least-squares estimate $\hat{\theta}$ [Seber and Wild (4)]. In summary, the Gauss–Newton method is obtained by using a

first-order Taylor expansion of ε in Eq. (73) around the most recent value $\theta^{(k)}$, substituting the approximated ε in $J(\theta) = \varepsilon^{\mathrm{T}}\varepsilon$, and estimating $\theta^{(k+1)}$ as the value which minimizes the approximated $J(\theta)$.

The Newton–Raphson method is also obtained by using an approximation of $J\ (\theta)$. This time $J\ (\theta)$ is expanded directly using a quadratic Taylor expansion around the most recent estimate $\theta^{(k)}$. If the gradient vector of $J\ (\theta)$ is denoted by

$$\boldsymbol{q}(\boldsymbol{\theta}) = \frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{75}$$

and the Hessian matrix of $J(\theta)$ is written as

$$\boldsymbol{H}\,(\boldsymbol{\theta}) = \frac{\partial^2 J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}\ \partial \boldsymbol{\theta}^{\mathrm{T}}} \tag{76}$$

the quadrature approximation of $J(\theta)$ around $\theta^{(k)}$ can be expressed by

$$J(\boldsymbol{\theta}) \approx J\big(\boldsymbol{\theta}^{(k)}\big) + \boldsymbol{q}^{\mathrm{T}}\big(\boldsymbol{\theta}^{(k)}\big)\big(\boldsymbol{\theta} - \boldsymbol{\theta}^{(k)}\big) + \frac{1}{2}\big(\boldsymbol{\theta} - \boldsymbol{\theta}^{(k)}\big)^{\mathrm{T}}\boldsymbol{H}\,\big(\boldsymbol{\theta}^{(k)}\big)\big(\boldsymbol{\theta} - \boldsymbol{\theta}^{(k)}\big) \tag{77}$$

The minimization of the approximated $J\ (\theta)$ is again a linear problem and thus is easily obtained. If the solution is $\theta^{(k+1)}$, we can write

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \mathrm{H}^{-1}\big(\boldsymbol{\theta}^{(k)}\big)\boldsymbol{q}\big(\boldsymbol{\theta}^{(k)}\big) \tag{78}$$

In summary, the Newton–Raphson method starts with an initial guess $\theta^{(0)}$ and proceeds by applying Eq. (78), where Eqs. (75) and (76) define the gradient $\boldsymbol{q}(\theta)$ and the Hessian $\boldsymbol{H}(\theta)$.

The Gauss–Newton and Newton–Raphson method must be applied with great care because they are iterative approaches and convergence to the least-squares estimate is a critical issue. Many adaptations and protective strategies have been developed to improve their reliability. For details, consult for example Seber and Wild (4). Convergence is assessed usually by criteria of the following forms:

$$0 \leq \frac{J\big(\boldsymbol{\theta}^{(k)}\big) - J\big(\boldsymbol{\theta}^{(k+1)}\big)}{J\big(\boldsymbol{\theta}^{(k)}\big)} < \tau \tag{79}$$

or

$$\max_{1 \leq j \leq p} \frac{\left|\theta_j^{(k)} - \theta_j^{(k+1)}\right|}{\left|\theta_j^{(k)}\right|} \leq \eta \tag{80}$$

where τ and η are some small predefined positive numbers. It should be kept in mind that these criteria do not guarantee convergence; they should, rather, be considered *termination* criteria.

Sequential Least Squares

In many applications the observed data are received sequentially in time. Quite often in such cases it is preferable to find and update the least-squares estimates of the unknowns as the data keep arriving. Procedures developed for processing the data in this fashion are known as sequential (or recursive) least-squares methods, as opposed to batch methods that use all the data at once. The concept is best described when the unknowns, which have to be estimated, are linear parameters.

Let the $N \times 1$ vector $\boldsymbol{y}$ be as in Eqs. (26) and (27), i.e.,

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{\varepsilon} \tag{81}$$

where $\boldsymbol{H}$ is an $N \times p$ matrix with known elements, and θ is a $p \times 1$ vector of unknown parameters. Suppose that the last sample that has been received is $y[n]$, that is, $y[k]$ has been observed for $k = 1,2, \ldots, n$, where

$$y[k] = \boldsymbol{h}^{\mathrm{T}}[k]\boldsymbol{\theta} + \varepsilon[k] \tag{82}$$

Here, the $p \times 1$ vector $\boldsymbol{h}^{\mathrm{T}}[k]$ is the kth row of the matrix $\boldsymbol{H}$ in Eq. (81). If we denote the observed vector up to sample n by $\boldsymbol{y}[n]$, then

$$\boldsymbol{y}[n] = \boldsymbol{H}[n]\boldsymbol{\theta} + \boldsymbol{\varepsilon}[n] \tag{83}$$

where $\boldsymbol{y}^{\mathrm{T}}[n] = [y[1]\ y[2]\ \cdots\ y[n]]$, $\boldsymbol{\varepsilon}^{\mathrm{T}}[n] = [\varepsilon[1]\ \varepsilon[2]\ \cdots\ \varepsilon[n]]$, and $\boldsymbol{H}[n]$ is an $n \times p$ matix identical to the first n rows of $\boldsymbol{H}$ (it is assumed that $n \geq p$, and that $\boldsymbol{H}[n]$ is a full-rank matrix). Then the least-squares estimate of θ based on the first n samples, denoted by $\hat{\theta}[n]$, is

$$\hat{\boldsymbol{\theta}}[n] = (\boldsymbol{H}^{\mathrm{T}}[n]\boldsymbol{H}[n])^{-1}\boldsymbol{H}^{\mathrm{T}}[n]\boldsymbol{y}[n] \tag{84}$$

Now, a new sample is received, $y[n + 1]$, and the objective is to modify $\hat{\theta}[n]$ so that the new information in $y[n + 1]$ is included in the estimate of θ. Of course, a straightforward way of accomplishing it would be to estimate $\hat{\theta}[n + 1]$ by an analogous expression to Eq. (84). However, there is a better way of getting $\hat{\theta}[n + 1]$, and it saves a great deal of computation. We rewrite Eq. (83) with the new sample $y[n + 1]$ as follows:

$$\begin{bmatrix} \boldsymbol{y}[n] \\ y[n+1] \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}[n] \\ \boldsymbol{h}^{\mathrm{T}}[n+1] \end{bmatrix} \boldsymbol{\theta} + \boldsymbol{\varepsilon}[n+1] \tag{85}$$

The usual minimization yields

$$\hat{\boldsymbol{\theta}}[n+1] = (\boldsymbol{H}^{\mathrm{T}}[n+1]\,\boldsymbol{H}[n+1])^{-1}\boldsymbol{H}^{\mathrm{T}}[n+1]\,\boldsymbol{y}[n+1] \tag{86}$$

where

$$(\boldsymbol{H}^{\mathrm{T}}[n+1]\,\boldsymbol{H}[n+1])^{-1} = (\boldsymbol{H}^{\mathrm{T}}[n]\,\boldsymbol{H}[n] + \boldsymbol{h}[n+1]\boldsymbol{h}^{\mathrm{T}}[n+1])^{-1} \tag{87}$$

The expression on the right-hand side of Eq. (87) can be rewritten by using the matrix inversion lemma

$$(\boldsymbol{A}+\boldsymbol{BCD})^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{C}^{-1}+\boldsymbol{DA}^{-1}\boldsymbol{B})^{-1}\boldsymbol{DA}^{-1} \tag{88}$$

and this allows us, after a few lines of derivation, to write

$$\hat{\boldsymbol{\theta}}[n+1]=\hat{\boldsymbol{\theta}}[n]+\boldsymbol{\kappa}[n+1](y[n+1]-\boldsymbol{h}^{\mathrm{T}}[n+1]\hat{\boldsymbol{\theta}}[n]) \tag{89}$$

where the $p \times 1$ vector $\kappa[n+1]$ is given by

$$\boldsymbol{\kappa}[n+1]=\boldsymbol{P}[n]\boldsymbol{h}[n+1](1+\boldsymbol{h}^{\mathrm{T}}[n+1]\boldsymbol{P}[n]\boldsymbol{h}[n+1])^{-1} \tag{90}$$

with $\boldsymbol{P}[n]$ being defined according to

$$\boldsymbol{P}[n]=(\boldsymbol{H}^{\mathrm{T}}[n]\boldsymbol{H}[n])^{-1} \tag{91}$$

From Eq. (89), it is important to note that the estimate of $\hat{\psi}[n+1]$ is given as a function of the previous estimate $\hat{\theta}[n]$. The term $\boldsymbol{h}^{\mathrm{T}}[n+1]\ \hat{\theta}[n]$ can be interpreted as the predicted value of $y[n+1]$ based on the past samples and the adopted linear model, and

$$e[n+1]=y[n+1]-\boldsymbol{h}^{\mathrm{T}}[n+1]\hat{\boldsymbol{\theta}}[n] \tag{92}$$

as the prediction error of the model. The vector $\kappa[n+1]$ is known as a gain vector, and thus, Eq. (89) has the predictor–corrector form. The updated estimate $\hat{\theta}[n+1]$ in Eq. (89) can also be written as

$$\hat{\boldsymbol{\theta}}[n+1]=(\boldsymbol{I}-\boldsymbol{\kappa}[n+1]\boldsymbol{h}^{\mathrm{T}}[n+1])\hat{\boldsymbol{\theta}}[n]+\boldsymbol{\kappa}[n+1]y[n+1] \tag{93}$$

which suggests a different interpretation of the updated estimate—it is a weighted sum of the previous estimate and the information provided by $y[n+1]$, where the gain $\kappa[n+1]$ is determined to allocate the weights optimally.

It seems that the computation of $\kappa[n+1]$ is rather demanding because of the need to compute $\boldsymbol{P}[n]$, which is obtained by inverting the $p \times p$ matrix $\boldsymbol{H}^{\mathrm{T}}[n]\boldsymbol{H}[n]$. In fact, this is not needed, because it can be shown that $\boldsymbol{P}[n]$ may be obtained from $\boldsymbol{P}[n-1]$ by

$$\boldsymbol{P}[n]=\boldsymbol{P}[n-1]-\boldsymbol{\kappa}[n]\boldsymbol{h}^{\mathrm{T}}[n]\boldsymbol{P}[n-1] \tag{94}$$

where, evidently, there is no inversion involved in the computation. In summary, given the most recent estimate $\hat{\theta}[n]$, gain $\kappa[n]$, and matrix $\boldsymbol{P}[n]$, upon receiving a new sample, $y[n+1]$, the sequential least-squares method updates them by applying Eqs. (90), (89), and (94).

It is important to stress that all sequential algorithms require initializations of θ, κ, and $\boldsymbol{P}$. This can be done in two ways: one is to use *a priori* knowledge and quantify it by assigning initial values to them; the other is to obtain the initial values by applying a batch method to a small portion of the data.

There is abundant literature on sequential least squares in many journals and books. What is described here is only the standard recursive least-squares algorithm. It has limitations in numerical robustness and (its recursive nature notwithstanding) excessive computational complexity. Some alternatives to the standard

recursive least-squares algorithm include the square-root recursive least-squares method, which is based on QR decomposition of matrices, and the fast recursive least-squares method, which is based on special implementations of linear least-squares prediction in both the forward and backward directions. For more information, refer to Haykin (6) and the references therein.

Predictive Least Squares

In many situations, observed data may have more than one possible model for their description. A typical example is when we want to fit the data with a polynomial and we do not know its degree. As was mentioned in the section on polynomial fitting and spline interpolation, a too high degree can easily overfit the data, whereas a too low degree may also be unsatisfactory. In most practical situations, the degree is unknown and some statistics must be used to determine it. In the case when we want to use the least-squares approach for estimation of unknown parameters, which precludes probabilistic assumptions about the observed data, there are not many reliable criteria available for selecting the right model for the data. An additional difficulty is that for different problems the roles of the models may be quite distinct, and the differences are then strongly reflected in the criteria for choosing the best model. The implication is clear—there can be no universal criterion for model selection.

One approach for selecting a model, which has been shown to work very well and is generally applicable, is known as the predictive least-squares method [Rissanen (7)]. It is very simple to use, and is based on the principle that good models predict the future from the past better than poor models. It is well known that a polynomial of high degree fits data better than a polynomial of low degree; in fact, if the degree is high enough, the fitting can be perfect. It should be noted that in usual situations, the criterion for measuring the goodness of fit is given by the total sum of squared residuals. It is very important to observe that the same data are used for estimation of the unknown polynomial coefficients and the computation of the residual. In many cases, this is not good philosophy: it goes against the principle of parsimony in science and engineering, which states that use of unnecessary parameters in modeling should be avoided. Here, we present the predictive least-squares method in the context of choosing the best degree of a polynomial.

Suppose that the observed data $\boldsymbol{y}$ can be modeled by a polynomial whose degree m comes from the set $\{0, 1, 2, \ldots, p\}$. The idea is to compare all the polynomials by using the same yardstick, which is the accumulated prediction error of each polynomial. The estimation of the polynomial coefficients is carried out in the usual way by applying one of the sequential least-squares algorithms. However, the validation of the polynomials is implemented by data that have not been used for parameter estimation. For example, if the next sample is $y[n + 1]$, once it is received, it is compared with the predicted value of $y[n + 1]$ given by

$$\hat{y}[n+1] = \boldsymbol{h}^{\mathrm{T}}[n+1]\hat{\boldsymbol{\theta}}[n] \tag{95}$$

and the prediction error $e[n + 1]$ computed as in Eq. (92). The squared value of the error is added to the accumulated sum of the previous prediction errors. Next, the parameter estimates are updated according to Eq. (86), and $\hat{\theta}[n + 1]$ is used for prediction of $y[n + 2]$. The degree of the polynomial is then selected as the one that minimizes the accumulated prediction error, that is,

$$\hat{m} = \arg\min_{m} \sum_{n} (y[n] - \hat{y}_m[n])^2 \tag{96}$$

where $\hat{y}[n]$ is the sample of $y[n]$ predicted by the polynomial with degree m.

In summary, the best polynomial is the one that minimizes the accumulated prediction errors as given by Eq. (96). The coefficients of every polynomial are estimated sequentially as presented in the previous section. As the unknown parameters of each polynomial are updated, the corresponding squared prediction errors are accumulated. Once all the data are processed and all the polynomials are examined, the polynomial whose total sum of squared prediction errors is minimal is the winner.

The Bootstrap Method

Least-squares estimation is used in problems where probabilistic assumptions about the data are not made. It seems then that it would be difficult to make claims about the accuracy of the obtained estimates, unless ubiquitous normal assumptions are made or large-sample theory invoked. Indeed, take the simple case of linear least-squares estimation where unknown parameters θ are estimated according to Eq. (32). Can we say anything about the accuracy of the estimates θ without making suppositions about the error vectors ε? The answer is yes, and a powerful statistical procedure for providing such assessments is known as the bootstrap method. Although the bootstrap method can be applied to various tasks, including confidence-interval estimation and hypothesis testing, the underlying principle in all the applications is the same.

The bootstrap method imitates a situation that a practitioner would like to have in order to assess the quality of estimates. In the case of the linear least-squares estimation, it would be nice to have many sets of data $\boldsymbol{y}_i$, $i = 1, 2, \ldots, M$, for which we could write

$$\boldsymbol{y}_i = \boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{\varepsilon}_i \tag{97}$$

where the ε_i's have the same statistical distribution. In that case, one would normally estimate the unknown parameters from each data set to obtain $\hat{\theta}_i$, from which statistics can be constructed that provide information about the accuracy of the estimates. Clearly, in most practical situations, multiple data sets are simply not available. In the late seventies, Efron proposed the bootstrap method to generate such data sets by repeatedly drawing random samples from the original data sets. To illustrate the procedure we proceed by way of example.

Suppose that the data set $\boldsymbol{y}$ is modeled as in Eq. (32) and the least-squares estimate of $\hat{\theta}$ is obtained in the usual way. Then we compute the residual data $\hat{\varepsilon}$ by

$$\hat{\boldsymbol{\varepsilon}} = \boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{\theta}} \tag{98}$$

The bootstrap is now applied by sampling randomly (with substitution) from the samples $\hat{\varepsilon}[n]$, $n = 1, 2, \ldots, N$, and constructing new data sets of the form

$$\boldsymbol{y}_i^* = \boldsymbol{H}\hat{\boldsymbol{\theta}} + \boldsymbol{\varepsilon}_i{}^* \tag{99}$$

where i stands for the ith data set, and

$$\boldsymbol{\varepsilon}_i^{*\mathrm{T}} = \begin{bmatrix} \varepsilon_i^*[1] & \varepsilon_i^*[2] & \cdots & \varepsilon_i^*[N] \end{bmatrix} \tag{100}$$

When sampling from $\hat{\varepsilon}$, some of the samples may not appear in $\varepsilon^*{}_i$, at all, and some may show up more than once. With the so constructed data sets, we proceed as if they were observed. Each $\boldsymbol{y}^*{}_i$ is processed to find

the least-squares estimate $\hat{\theta}_i$, and from all the estimates $\hat{\theta}_i$, $i = 1,2,\ldots,M$, various statistics for assessing the accuracy of $\hat{\theta}$ can easily be constructed.

This procedure can also be applied to any nonlinear least-squares method with practically no modifications. For further details about the bootstrap method, see Efron and Tibshirani (8).

In conclusion, the bootstrap is a simple method for statistical inference, especially in cases where a few statistical assumptions are made about the observed data, as in problems where least-squares estimation is employed. The drawback of the method is that it is computationally rather intensive.

BIBLIOGRAPHY

1. C. L. Lawson R. J. Hanson *Solving Least Squares Problems*, Englewood Cliffs, NJ: Prentice-Hall, 1974.
2. P. Dierckx *Curve and Surface Fitting with Splines*, Oxford, England: Oxford University Press, 1993.
3. S. M. Kay *Fundamentals of Statistical Signal Processing: Estimation Theory*, Englewood Cliffs, NJ: PTR Prentice-Hall, 1993.
4. G. A. F. Seber C. J. Wild *Nonlinear Regression*, New York: Wiley, 1989.
5. G. H. Golub V. Pereira The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate, *SIAM J. Numer. Anal.*, **10**: 413–432, 1973.
6. S. Haykin *Adaptive Filter Theory*, Upper Saddle River, NJ: Prentice-Hall, 1996.
7. J. Rissanen Order estimation by accumulated prediction errors, *J. Appl. Probab.*, **23A**: 55–61, 1986.
8. B. Efron R. J. Tibshirani *An Introduction to the Bootstrap*, New York: Chapman & Hall, 1993.

READING LIST

G. Evans *Practical Numerical Analysis*, West Sussex, England: Wiley, 1995.
R. P. Feinerman D. J. Newman *Polynomial Approximation*, Baltimore: Williams & Wilkins, 1974.
W. Gautschi *Numerical Analysis: An Introduction*, Boston: Birkhäuser, 1997.
J. H. Mathews *Numerical Methods for Computer Science, Engineering and Mathematics*, Englewood Cliffs, NJ: Prentice-Hall, 1987.

ZORAN I. MITROVSKI
Morgan Stanley Dean Witter
PETAR M. DJURIC
State University of New York

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Linear Algebra
Standard Article
N. Puri[1]
[1]Rutgers University, Piscataway, NJ
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2432
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (271K)

Abstract

The sections in this article are

Vector Spaces

Euclidian Space and Matrix Representation

Matrix Algebra

Rank of a Matrix

Eigenvalues and Eigenvectors of Matrices

Diagonalization of Matrices

The Jordan Canonical Form

Cayley–Hamilton Theorem

Computation of a Polynomial Function of the Matrix math

Companion Matrices

Cholesky Decomposition (Also Known as LU Decomposition)

Jacobi and Gauss–Seidel Methods

Least-Squares Best-Fit Problem (Also Known as the Pseudoinverse Problem)

Hermitian (or Symmetric Real) Matrices and Definite Functions

Some Facts and Identities

| | |

Copyright © 1999-2008 All Rights Reserved.

LINEAR ALGEBRA

This article deals with linear vector spaces, transformations, quadratic forms, and structural relationships between algebraic systems. Matrix theory concepts necessary to compute functions of matrices involved in system theory are developed.

VECTOR SPACES

Definition

Many different topics, such as matrices, orthogonal polynomials, Fourier series, and integrodifferential equations, can be

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

united as a study of linear vector spaces, because they all satisfy the following definition: A *linear vector space* **v** (over a scalar field R or C) is a set of objects $\boldsymbol{x}, \boldsymbol{y}, \ldots$ called *vectors,* together with the two operations of *addition* and *scalar multiplication* with the following properties: If vectors $\mathbf{0}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \ldots \in \mathbf{v}$ and α and β are complex numbers, then

1. $\boldsymbol{x} + \boldsymbol{y} \in \mathbf{v}$
2. $\alpha\boldsymbol{x} \in \boldsymbol{v}$
3. $\alpha(\boldsymbol{x} + \boldsymbol{y}) = \alpha\boldsymbol{x} + \alpha\boldsymbol{y}$
4. $(\alpha\beta)\boldsymbol{x} = \alpha(\beta\boldsymbol{x})$
5. $\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{y} + \boldsymbol{x}$
6. $1\mathbf{x} = \mathbf{x}$
7. $\mathbf{0} + \boldsymbol{x} = \boldsymbol{x}$
8. $\boldsymbol{x} + (-\boldsymbol{x}) = \boldsymbol{0}$

A *metric vector space* is associated with a real valued nonnegative function such that:

1. $g(\mathbf{x},\mathbf{y}) = g(\mathbf{y},\mathbf{x})$
2. $g(\mathbf{x},\mathbf{y}) = 0$, if $\mathbf{x} = \mathbf{y}$
3. $g(\mathbf{x},\mathbf{y}) \leq g(\mathbf{x},\mathbf{z}) + g(\mathbf{z},\mathbf{y})$

A *metric vector space* is called complete if every "Cauchy sequence" $\mathbf{x}_n$ in the *metric space* converges to some $\mathbf{x} \in \mathbf{v}$. A metric space is normed if for all vectors **x** and **y** in **v** and a scalar α a norm $\| \ \|$ is defined with the following properties:

1. $\|\mathbf{x}\| > 0$, $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = 0$
2. $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$
3. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

A normed metric space which is complete is called *Banach* space.

Some important metric notions such as length, direction, and energy can be expressed if the vector space is endowed with the additional *inner product* $(\boldsymbol{x}, \boldsymbol{y})$ of $\boldsymbol{x}$ and $\boldsymbol{y}$ which satisfies

1. $(\boldsymbol{x}, \boldsymbol{y}) = \overline{(\boldsymbol{y}, \boldsymbol{x})}$
2. $(\boldsymbol{x}, \boldsymbol{x}) \geq \mathbf{0}$
3. $(\alpha\boldsymbol{x}, \boldsymbol{y}) = \overline{\alpha}(\boldsymbol{x}, \boldsymbol{y})$
4. $(\mathbf{x}, \mathbf{x}) = \mathbf{0} \rightarrow \mathbf{x} = \mathbf{0}$
5. $(\mathbf{x}, \mathbf{y} + \mathbf{z}) = (\mathbf{x}, \mathbf{y}) + (\mathbf{x}, \mathbf{z})$

where the overbar stands for complex conjugation.

A *Banach space* with the inner product defined as above is known as *Hilbert space.*

Different Types of Linear Spaces

***n*-Dimensional Euclidean Space.** Two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are a n-tuples complex numbers

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Their inner product is

$$(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} \overline{x}_i y_i$$

The *length* of $\boldsymbol{x}$ is

$$(\boldsymbol{x}, \boldsymbol{x})^{1/2} = \sum_{i=1}^{n} (\overline{x}_i x_i)^{1/2} = \sum_{i=1}^{n} |x_i| = \|\boldsymbol{x}\| \geq 0$$

The *Cauchy–Schwartz inequality* states

$$|(\boldsymbol{x}, \boldsymbol{y})| \leq \|\boldsymbol{x}\| \, \|\boldsymbol{y}\|$$

Vectors $\boldsymbol{x}_i$ form an *orthonormal set* if

$$(\boldsymbol{x}_i, \boldsymbol{x}_j) = \delta_{ij} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \qquad (i, j = 1, 2, \ldots)$$

The $L^2(a, b)$ Space Known as Hilbert Space. The sum of two functions (vectors) $\boldsymbol{f}(t)$ and $\boldsymbol{g}(t)$ is $\boldsymbol{f}(t) + \boldsymbol{g}(t)$, and their inner product is

$$(\boldsymbol{f}, \boldsymbol{g}) = \int_a^b \overline{\boldsymbol{f}}(t)\boldsymbol{g}(t)\, dt$$

Polynomial Space. Let

$$\boldsymbol{f}(t) = \sum_{i=1}^{n} a_i t^i$$

Addition is defined as usual, and for a weighting function $w(t) > 0$ and interval (a, b), the inner product is

$$(\boldsymbol{f}, \boldsymbol{g}) = \int_a^b \overline{\boldsymbol{f}}(t) w(t) \boldsymbol{g}(t)\, dt$$

Generalized Fourier Space. Define an orthonormal set of vectors $\boldsymbol{e}_k$ ($k = 0, \pm 1, \pm 2, \ldots$). If $\boldsymbol{x}$ is any arbitrary vector in **v**, then its *Fourier expansion* is

$$\boldsymbol{x} = \sum_{k=-\infty}^{\infty} c_k \boldsymbol{e}_k$$

with *Fourier coefficients*

$$(\boldsymbol{e}_k, \boldsymbol{x}) = c_k \qquad (k = 0, \pm 1, \pm 2, \ldots)$$

In particular, the *Fourier vectors* are given by

$$\boldsymbol{e}_k = \boldsymbol{e}_k(t) = e^{jk\omega t}$$

on the interval $[a, b] \equiv [-T, T]$, $\omega = \pi/T$.

Parseval's identity is

$$(\boldsymbol{x}, \boldsymbol{x}) = \sum_{k=-\infty}^{\infty} |(\mathbf{e}_k, \mathbf{x})|^2 = \sum_{k=-\infty}^{\infty} |c_k|^2$$

If we choose n-dimensional orthogonal space, the *least-squares error approximation* of $\boldsymbol{x}$ is defined as

$$\boldsymbol{x}^* = \sum_{k=-n}^{n} c_k \boldsymbol{e}_k, \qquad (\boldsymbol{e}_k, \boldsymbol{e}_j) = \delta_{kj} \quad (k, j = 0, \pm 1, \pm 2, \ldots)$$

Bessel's inequality is

$$(\boldsymbol{x}, \boldsymbol{x}) = \sum_{k=-\infty}^{\infty} |c_k|^2 \geq \sum_{k=-n}^{n} |c_k|^2$$

An orthogonal set is *complete* if ther is $\boldsymbol{x}$ for which

$$\|\boldsymbol{x} - \boldsymbol{x}^*\| < \epsilon, \qquad \epsilon \geq 0$$

The *Riemann-Lebesgue lemma* states that

$$|(\boldsymbol{e}_k, \boldsymbol{x})| = |c_k| \to 0$$

as $k \to \infty$.

Gram–Schmidt Orthogonalization

Given a vector set $\boldsymbol{e}_i$ ($i = 1, 2, \ldots, n$) and constants α_i ($i = 1, 2, \ldots, n$) not all zero, such that

$$\sum_{i=1}^{n} \alpha_i \boldsymbol{e}_i = \boldsymbol{0}$$

the vectors are said to be *linearly dependent.* Otherwise the set is composed of *linearly independent* vectors.

Given a linearly independent set of vectors $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n$, suppose we are required to determine a new orthonormal set $\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_n$.

Take $\boldsymbol{f}_1 = \boldsymbol{e}_1/\|\boldsymbol{e}_1\|$. Assume $\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_k$ ($k < n$) have been computed; then

$$\boldsymbol{f}'_{k+1} = \boldsymbol{e}_{k+1} - \sum_{j=1}^{k} (\boldsymbol{f}_j, \boldsymbol{e}_{k+1}) \boldsymbol{f}_j, \quad \boldsymbol{f}_{k+1} = \frac{\boldsymbol{f}'_{k+1}}{\|\boldsymbol{f}'_{k+1}\|} \qquad k = 2, 3, \ldots, n$$

A vector space $\mathbf{v}$ is *n-dimensional* if it contains only n linearly independent vectors. Every set of $n + 1$ vectors is linearly dependent. The set of linearly independent vectors *spans* a space $\mathbf{v}$ if every vector $\boldsymbol{x} \in \mathbf{v}$ can be expressed as

$$\boldsymbol{x} = \sum_{i=1}^{n} \alpha_i \boldsymbol{e}_i$$

The $\boldsymbol{e}_i$, $i = 1, \ldots, n$, are called a *basis* of $\mathbf{v}$.

Linear Operators

A *linear operator* $\mathbf{T}$ on a vector space $\mathbf{v}$ defines a rule that computes $\mathbf{T}x$ for $\boldsymbol{x} \in \mathbf{v}$ such that

$$\mathbf{T}(\alpha_1 \boldsymbol{x} + \alpha_2 \boldsymbol{y}) = \alpha_1 \mathbf{T}\boldsymbol{x} + \alpha_2 \mathbf{T}\boldsymbol{y}$$

If any vector in $\mathbf{v}$ can be expressed as sum of vectors from *subspaces* $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ ($\boldsymbol{x} = \sum_{i=1}^{n} \boldsymbol{x}_i$, $\boldsymbol{x} \in \mathbf{v}$, $\boldsymbol{x}_i \in \mathbf{v}_i$), then $\mathbf{v}$ itself is called the *sum of the subspaces:*

$$\mathbf{v} = \sum_{i=1}^{n} \mathbf{v}_i$$

The *direct sum* of vector spaces such that $\mathbf{v}_i \cap \mathbf{v}_j = 0$, $i \neq j$ is denoted

$$\mathbf{v} = \mathbf{v}_1 \dot{+} \mathbf{v}_2 \dot{+} \cdots = \dot{\sum_{i=1}^{k}} \mathbf{v}_i$$

The set $\{\mathbf{v}_i\}$ is called a *direct decomposition* of $\mathbf{v}$.

The *projection theorem* states that

$$\mathbf{v} = \mathbf{w} \dot{+} \mathbf{w}^{\perp}, \quad \boldsymbol{x} \in \mathbf{w}, \text{ and } \boldsymbol{y} \in \mathbf{w}^{\perp} \quad \text{implies} \quad (\boldsymbol{x}, \boldsymbol{y}) = 0$$

$$\dim(\mathbf{v}_1 + \mathbf{v}_2) = \dim \mathbf{v}_1 + \dim \mathbf{v}_2 - \dim(\mathbf{v}_1 \cap \mathbf{v}_2)$$

$$\dim \mathbf{v} = \dim(\mathbf{w} \dot{+} \mathbf{w}^{\perp}) = \dim \mathbf{w} + \dim \mathbf{w}^{\perp}$$

Two spaces $\mathbf{v}$ and $\mathbf{w}$ are *dual* to each other if the basis vectors of $\mathbf{v}$ are $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n$, the basis vectors of $\mathbf{w}$ are $\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_n$, and

$$(\boldsymbol{e}_i, \boldsymbol{f}_j) = \delta_{ij} \qquad (i, j = 1, 2, \ldots)$$

The set $\boldsymbol{e}_i$ ($i = 1, \ldots, n$) from a *basis* for the space $\mathbf{v}$ if every vector $\boldsymbol{x} \in \mathbf{v}$ can be expressed as a linear combination of these vectors. The *dimension* of the space is the maximal number of linearly independent vectors in the space. In an n-dimensional linear vector space any set of n linearly independent vectors qualifies as a basis for the vector space.

EUCLIDIAN SPACE AND MATRIX REPRESENTATION

Consider two separate spaces $\mathbf{E}_m$ and $\mathbf{E}_n$ with bases $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_m$ and $\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_n$, respectively, given by

$$(\boldsymbol{e}_k)_l = (l\text{th component of } \boldsymbol{e}_k) = \delta_{kl} \qquad (k, l = 1, 2, \ldots, m)$$
$$(\boldsymbol{f}_j)_i = (i\text{th component of } \boldsymbol{f}_j) = \delta_{ij} \qquad (i, j = 1, 2, \ldots, n)$$

Let the operator $\mathbf{A}$ be a *transport,* or *linear mapping,* from $\mathbf{E}_m$ to $\mathbf{E}_n$ (see Fig. 1):

$$y = \mathbf{A}\boldsymbol{x}, \qquad \boldsymbol{x} \in \mathbf{E}_m, \quad \boldsymbol{y} \in \mathbf{E}_n$$

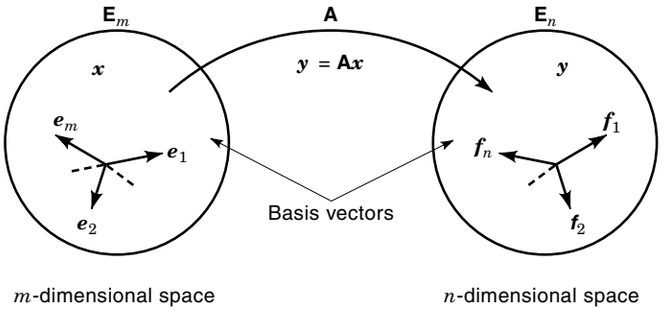


Figure 1. Matrix representation.

Let

$$\mathbf{A}\boldsymbol{e}_i = \boldsymbol{g}_i \qquad (i = 1, 2, \ldots, m)$$

Since $\boldsymbol{g}_i \in \mathbf{E}_n$, it can be represented as a linear combination of the $\boldsymbol{f}_j$:

$$\boldsymbol{g}_i = \sum_{j=1}^{n} a_{ij}\boldsymbol{f}_j$$

The operator $\mathbf{A}$ determines the numbers a_{ij}. We have

$$x = \sum_{i=1}^{m} x_i \boldsymbol{e}_i, \qquad y = \sum_{j=1}^{n} y_j \boldsymbol{f}_j \tag{1}$$

where x_i and y_j are the ith and jth components of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and

$$\begin{aligned} y = \mathbf{A}x &= \sum_{i=1}^{m} x_i \mathbf{A}e_i = \sum_{i=1}^{m} x_i \boldsymbol{g}_i \\ &= \sum_{i=1}^{m} x_i \sum_{j=1}^{n} a_{ij}\boldsymbol{f}_j = \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} x_i \right) \boldsymbol{f}_j \end{aligned} \tag{2}$$

From (1) and (2),

$$y_j = \sum_{i=1}^{m} a_{ij} x_i \qquad (j = 1, 2, \ldots, n) \tag{3}$$

The action of the operator $\mathbf{A}$ can be fully computed from the numbers a_{ij} $(i = 1, \ldots, m; j = 1, \ldots, n)$. These numbers, when arranged as a table of n rows and m columns, constitute an $n \times m$ matrix $\underline{\mathbf{A}}$. Thus Eq. (3) can be written

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \tag{4}$$

or

$$\underbrace{\underline{\boldsymbol{y}} = \underline{\mathbf{A}}\underline{\boldsymbol{x}}}_{\text{matrix relation}} \quad \leftrightarrow \quad \underbrace{\boldsymbol{y} = \mathbf{A}\boldsymbol{x}}_{\text{space relation}} \tag{5}$$

Thus, $\underline{\boldsymbol{x}}$, $\underline{\boldsymbol{y}}$, $\underline{\mathbf{A}}$ are the matrix representations of the vectors $\boldsymbol{x}$, $\boldsymbol{y}$ and the operator $\mathbf{A}$ with respect to the basis $\{\boldsymbol{e}_i\}_1^m$ in $\mathbf{E}_m$ and the basis $\{\boldsymbol{f}_j\}_1^n$ in $\mathbf{E}_n$. Basis vectors are analogous to coordinates. However, vectors and operators exist indenendently of the basis assigned to them.

A vector space whose vectors belong to some larger space is called a *subspace.* This concept is very useful in developing the *canonical form* of a matrix. Let $\mathbf{A}$ be a mapping of $\mathbf{E}_n$ onto itself. A subspace $\mathbf{E}_{n_i}$ of $\mathbf{E}_n$ is *invariant* with respect to $\mathbf{A}$ if $\mathbf{A}\boldsymbol{x} \in \mathbf{E}_{n_i}$ implies $\boldsymbol{x} \in \mathbf{E}_{n_i}$. The structure of a *invariant mapping* (matrix) can be very usefully exploited by means of its invariant subspaces:

$$\mathbf{E}_n = \sum_{i=1}^{k} \mathbf{E}_{n_i} \qquad \left(i = 1, \ldots, k; \quad \sum_{j=1}^{k} n_j = n \right)$$

The basis of $\mathbf{E}_{n_i}$ consists of $\boldsymbol{e}_{ij}$ $(i = 1, \ldots, k; j = 1, \ldots, n_j)$.

In the basis chosen, $\mathbf{A}$ can be represented by a *quasidiagonal* form

$$\underline{\mathbf{A}} = \begin{bmatrix} \underline{\mathbf{A}}_1 & & & \\ & \underline{\mathbf{A}}_2 & & \\ & & \ddots & \\ & & & \underline{\mathbf{A}}_n \end{bmatrix}$$

All other entries besides boxes along the main diagonal are zeros.

MATRIX ALGEBRA

A scalar is a special case of a matrix with one row and one column. Following is a review of matrix theory fundamentals:

1. *Column matrix* (or vector):

$$\underline{\boldsymbol{x}} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad (n \times 1 \text{ matrix})$$

2. *Row matrix* (or vector):

$$\underline{\mathbf{x}}^{\mathrm{T}} = [x_1, \ldots, x_n] \qquad (1 \times n \text{ matrix})$$

3. *Matrix* of order $n \times m$:

$$\underline{\mathbf{A}} = (a_{ij}) \qquad (i = 1, \ldots, n; \quad j = 1, \ldots, m)$$

4. *Addition of matrices:*

$$\underline{\mathbf{A}} + \underline{\mathbf{B}} = (a_{ij} + b_{ij}) \qquad (i = 1, \ldots, n;\ j = 1, \ldots, m)$$

5. *Multiplication:* If $\underline{\mathbf{A}}$ is $n \times p$ and $\underline{\mathbf{B}}$ is $p \times m$,

$$\underline{\mathbf{A}}\,\underline{\mathbf{B}} = \sum_{k=1}^{p} a_{ik} b_{kj} \qquad (i = 1, \ldots, n; \quad j = 1, \ldots, m)$$

6. *Adjoint:* Let $\underline{\mathbf{A}}^{\mathrm{T}}$ be the transpose of $\underline{\mathbf{A}}$, $\underline{\mathbf{A}}^{\mathrm{T}} = (a_{ji})$ ($\underline{\mathbf{A}}$ with rows and columns exchanged). Then the *adjoint matrix* of $\underline{\mathbf{A}}$ is

$$\overline{\underline{\mathbf{A}}^{\mathrm{T}}} = \underline{\mathbf{A}}^{*}$$

$\underline{\mathbf{A}}$ is a *unitary matrix* if

$$\underline{\mathbf{A}}^{-1} = \underline{\mathbf{A}}^{*}$$

a *symmetric matrix* if

$$\underline{\mathbf{A}}^{\mathrm{T}} = \underline{\mathbf{A}}$$

and a *Hermitian matrix* (useful in physics) if

$$\overline{\underline{\mathbf{A}}^{\mathrm{T}}} = \underline{\mathbf{A}} = \underline{\mathbf{A}}^{*}$$

The *commutator* of $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ is

$$\underline{\mathbf{AB}} - \underline{\mathbf{BA}} = [\underline{\mathbf{A}}, \underline{\mathbf{B}}]$$

$[\underline{\mathbf{A}}, \underline{\mathbf{B}}] = 0$ implies $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ are Hermitian.

7. *Inverse matrix:* We shall assume knowledge of the determinant of a matrix and its elementary properties. Let $\Delta(\underline{\mathbf{A}})$ be the determinant of $\underline{\mathbf{A}}$, where $\underline{\mathbf{A}}$ is an $n \times n$ square matrix. Let A^{ij} be the *ij cofactor* of a_{ij}, that is, the determinant of the matrix $\underline{\mathbf{A}}$ after striking out the ith row and the jth column, multiplied by $(-1)^{i+j}$. Then

$$\sum_{j=1}^{n} a_{ij} A^{ij} = \Delta(\underline{\mathbf{A}})$$

(the *Laplace expansion*), and

$$\sum_{j=1}^{n} a_{ij} A^{kj} = \Delta(\underline{\mathbf{A}})\delta_{ik}$$

The *inverse matrix* $\underline{\mathbf{A}}^{-1}$ is given by

$$(\underline{\mathbf{A}}^{-1})_{ij} = [\Delta(\underline{\mathbf{A}})]^{-1} A^{ji}, \qquad \underline{\mathbf{A}}^{-1} = [\Delta(\underline{\mathbf{A}})]^{-1}(\mathrm{Adj}\boldsymbol{A})$$

and we have

$$\underline{\mathbf{AA}}^{-1} = \underline{\mathbf{A}}^{-1}\underline{\mathbf{A}} = \underline{\mathbf{I}} \qquad \text{(identity)}$$

$$(\underline{\mathbf{AA}}^{-1})_{ij} = \sum_{k=1}^{n} a_{ik} (\underline{\mathbf{A}}^{-1})_{kj} = [\Delta(\underline{\mathbf{A}})]^{-1} \sum_{k=1}^{n} a_{ik} A^{kj} = \delta_{ij}$$

8. The *determinant of a product of matrices* is $\Delta(\underline{\mathbf{AB}}) = \Delta(\underline{\mathbf{A}})\Delta(\underline{\mathbf{B}})$.
9. A *singular matrix* $\underline{\mathbf{A}}$ is one such that

$$\Delta(\underline{\mathbf{A}}) = 0$$

10. Sometimes it is useful to represent a $n \times m$ matrix as a collection of n row vectors or m column vectors:

$$\underline{\mathbf{A}} = \begin{bmatrix} b_1^T \\ b_2^T \\ \vdots \\ b_n^T \end{bmatrix} \quad \text{or} \quad \underline{\mathbf{A}} = [\underline{\boldsymbol{a}}_1 \quad \dots \quad \underline{\boldsymbol{a}}_m]$$

b_i^T is a $1 \times m$ row vector, and $\underline{\boldsymbol{a}}_j$ is an $n \times 1$ column vector ($i = 1, \ldots, n; j = 1, \ldots, m$).

11. *Minors:* Choose any k rows and any k columns from the matrix $\underline{\mathbf{A}}$ and form a matrix. The determinant of this matrix, with rows and columns in their natural order, is called a *minor* of $\underline{\mathbf{A}}$ of the kth order.
12. *Projection matrices:* If $\underline{\mathbf{P}}$ is Hermitian and $\underline{\mathbf{P}}^n = \underline{\mathbf{P}}$ ($n = 2, \ldots$), then $\underline{\mathbf{P}}$ is called a *projection matrix.* Any arbitrary vector $\underline{\boldsymbol{x}}$ can be decomposed into $\underline{\boldsymbol{y}}$ and $\underline{\boldsymbol{z}}$ such that

$$\underline{\mathbf{Py}} = \underline{\mathbf{y}}, \qquad \underline{\mathbf{Pz}} = \underline{\mathbf{0}}, \qquad \underline{\mathbf{x}} = \underline{\mathbf{y}} + \underline{\mathbf{z}}, \qquad \underline{\mathbf{z}} = (\underline{\mathbf{I}} - \underline{\mathbf{P}})\,\underline{\mathbf{x}}$$

The vector space $\mathbf{v}$ can be decomposed into $\mathbf{w}$ and $\mathbf{w}^{\perp}$, where

$$(\underline{\mathbf{P}}\underline{\boldsymbol{x}} = \underline{\boldsymbol{y}}) \in \mathbf{w} \text{ and } (\underline{\mathbf{I}} - \underline{\mathbf{P}})\underline{\boldsymbol{x}} = \underline{\boldsymbol{z}} \in \mathbf{w}^{\perp}, \qquad \mathbf{w} + \mathbf{w}^{\perp} = \mathbf{v}$$

RANK OF A MATRIX

The rank is a very useful concept in the solution of simultaneous equations. It can be defined in many different (but equivalent) ways. In particular, it is the largest order of a nonvanishing minor, and it is the maximum number of linearly independent rows (or of linearly independent columns). Thus, given an $n \times m$ matrix $\underline{\mathbf{A}}$, the rank $r \le n, m$.

Kernel and Range

Let $\underline{\mathbf{A}}$ be a transformation on $\mathbf{E}_n$ from $\mathbf{E}_m$. The *kernel* of $\underline{\mathbf{A}}$ is the totality of $\underline{\boldsymbol{x}} \in \mathbf{E}_m$ for which $\underline{\mathbf{A}}\underline{\boldsymbol{x}} = \underline{\mathbf{0}}$. The *range* of $\underline{\mathbf{A}}$ is the totality of vectors $\underline{\mathbf{A}}\underline{\boldsymbol{x}} \in \mathbf{E}_n$. These are denoted as Ker $\underline{\mathbf{A}}$ and rng $\underline{\mathbf{A}}$. Let dim stand for dimension. Then dim Ker $\underline{\mathbf{A}}$ is also known as the *nullity* of $\underline{\mathbf{A}}$. Furthermore, dim rng $\underline{\mathbf{A}}$ is the rank of $\underline{\mathbf{A}}$.

Sylvester's law of nullity states that

$$\dim \mathrm{Ker}\, \underline{\mathbf{A}} + \dim \mathrm{rng}\, \underline{\boldsymbol{A}} = \dim \mathbf{E}_n$$

Systems of Linear Algebraic Equations

Let $\underline{\mathbf{A}}$ be an $n \times m$ matrix, $\underline{\mathbf{x}}$ be an $m \times 1$ matrix, and $\underline{\boldsymbol{b}}$ be an $n \times 1$ matrix forming a system of equations

$$\underline{\mathbf{A}}\underline{\boldsymbol{x}} = \underline{\boldsymbol{b}}$$

Let $\underline{\mathbf{B}} = [\underline{\mathbf{A}}, \underline{\boldsymbol{b}}]$ be the $n \times (m + 1)$ augmented matrix.

This system has a solution only if $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ have same rank r. Only r of the m components of $\underline{\boldsymbol{x}}$ can be uniquely determined, and $m - r$ can be assigned at will.

Nonsingular Matrices

For an $n \times n$ matrix $\underline{\mathbf{A}}$ to be nonsingular (invertible), its determinant $\Delta(\mathbf{A})$ must be nonzero, which implies that its rows (or columns) are linearly independent. This means the rank of $\underline{\mathbf{A}}$ is n. For an invertible $\underline{\mathbf{A}}$, $\underline{\mathbf{A}}\underline{\boldsymbol{x}} = \underline{\mathbf{0}}$ implies no linearly independent solutions besides $\underline{\boldsymbol{x}} = \underline{\mathbf{0}}$.

EIGENVALUES AND EIGENVECTORS OF MATRICES

We shall consider only $n \times n$ square matrices.

Eigenvalue and Eigenvector

Suppose $\underline{\mathbf{A}}\underline{\boldsymbol{x}} = \lambda\underline{\boldsymbol{x}}$. Then the scalar λ is known as an *eigenvalue* and $\underline{\boldsymbol{x}}$ as an *eigenvector* of $\underline{\mathbf{A}}$. We have

$$(\lambda\underline{\mathbf{I}} - \underline{\mathbf{A}})\underline{\boldsymbol{x}} = \underline{\mathbf{A}}(\lambda)\underline{\boldsymbol{x}} = \underline{\mathbf{0}} \tag{6}$$

implying

$$\sum_{j=1}^{n} (\lambda\delta_{ij} - a_{ij})\underline{\boldsymbol{x}}_j = \underline{\mathbf{0}} \qquad (i = 1, \ldots, n)$$

If the rank of $\underline{\mathbf{A}}(\lambda)$ is $r \leq n$, then it has r linearly independant nontrivial eigenvectors and a maximum of r distinct eigenvalues. For at least one nontrivial solution of (6) we must have the scalar equation

$$\det \underline{\mathbf{A}}(\lambda) = \Delta_{\underline{\mathbf{A}}}(\lambda) = |(\lambda \underline{\mathbf{I}} - \underline{\mathbf{A}})| = 0 \tag{7}$$

$\Delta_{\underline{\mathbf{A}}}(\lambda)$ is called the *characteristic polynomial* in λ of degree n, and equation (7) is called the *characteristic equation.* We have

$$\begin{aligned} \Delta_{\underline{\mathbf{A}}}(\lambda) = P(\lambda) &= \begin{bmatrix} \lambda - a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ -a_{n1} & \cdots & \lambda - a_{nn} \end{bmatrix} \\ &= \lambda^n + a_1\lambda^{n-1} + \cdots + a_n \end{aligned} \tag{8}$$

According to the fundamental theorem of algebra (8) has n roots $\lambda_1, \ldots, \lambda_n$, not necessarily all distinct. These roots λ_i are the eigenvalues of $\underline{\mathbf{A}}$ belonging to the corresponding eigenvector $\underline{\boldsymbol{x}}_i$.

Elementary Symmetric Functions

Let

$$P(\lambda) = a_0\lambda^n + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \cdots + a_n = \prod_{i=1}^{n} (\lambda - \lambda_i) \tag{9}$$

Then

$$\begin{aligned} a_0 &= 1 \\ (-1)a_1 &= \sum_{i=1}^{n} \lambda_i \\ (-1)^2 a_2 &= \frac{1}{2!} \sum_{i,j=1}^{n}{}' \lambda_i\lambda_j \\ &\vdots \\ (-1)^m a_m &= \frac{1}{m!} \sum_{i_1,i_2,\ldots i_m=1}^{n}{}' \lambda_{i_1}\lambda_{i_2} \cdots \lambda_{i_m} \\ &\cdot \\ (-1)^n a_n &= \prod_{i=1}^{n} \lambda_i \end{aligned}$$

where a prime on the summation implies a sum only over distinct subscripts.

Two important quantities associated with $\underline{\mathbf{A}}$, its trace (also called *Spur*) and determinant, can then be expressed as

$$\begin{aligned} \mathrm{Tr}\,\underline{\mathbf{A}} &= \sum_{i=1}^{n} a_{ii} = \sum_{i=1}^{n} \lambda_i \\ \det \underline{\mathbf{A}} &= \Delta_{\underline{\mathbf{A}}} = \Delta_{\underline{\mathbf{A}}}(0) = \prod_{i=1}^{n} \lambda_i \end{aligned}$$

Generalized Eigenvectors of Multiplicity k

Let

$$(\lambda \underline{\mathbf{I}} - \underline{\mathbf{A}})^k \underline{\boldsymbol{x}} = \underline{\mathbf{0}}, \quad (\lambda \underline{\mathbf{I}} - \underline{\mathbf{A}})^{k-1} \neq \underline{\mathbf{0}} \qquad (k \leq n) \tag{10}$$

The vector $\underline{\boldsymbol{x}}$ is called a *generalized eigenvector* of $\underline{\mathbf{A}}$ of multiplicity k.

If the generalized eigenvector $\underline{\boldsymbol{x}}_i$ of order k belongs to the eigenvalue λ_i, then the chain of k generalized eigenvector $\{\underline{\boldsymbol{x}}_i, (\lambda \underline{\mathbf{I}} - \underline{\mathbf{A}})\underline{\boldsymbol{x}}_i, \ldots, (\lambda \underline{\mathbf{I}} - \mathbf{A})^{k-1}\underline{\boldsymbol{x}}_i\}$ are linearly independent and can be utilized as k linearly independent eigenvectors of $\underline{\mathbf{A}}$.

Observe that:

1. Generalized eigenvectors of a matrix corresponding to different eigenvalues are linearly independent.
2. Eigenvalues of a Hermitian matrix are real, and eigenvectors corresponding to different eigenvalues are orthogonal. This result plays a important role in physics, particularly in quantum mechanics.

Norm of a Matrix

A *norm* of a matrix $\underline{\mathbf{A}}$, denoted by $\|\underline{\mathbf{A}}\|$, corresponding to the "greatest stretching" of vectors under its mapping. Three main useful norms are

$$\|\underline{\mathbf{A}}\|_m = \max_i \sum_j |a_{ij}|, \qquad \|\boldsymbol{x}\|_m = \max_i |\mathbf{x}_i| \qquad (m - \text{norm}) \tag{11a}$$

$$\|\underline{\mathbf{A}}\|_l = \max_j \sum_i |a_{ij}|, \qquad \|\boldsymbol{x}\|_l = \sum_j |\mathbf{x}_j| \qquad (l - \text{norm}) \tag{11b}$$

$$\|\underline{\mathbf{A}}\|_k = \left(\sum_{i,j} |a_{ij}|^k \right)^{1/k}, \qquad \|\boldsymbol{x}\|_k = \left[\sum_j |\mathbf{x}_j|^k \right]^{1/k} \qquad (k - \text{norm}) \tag{11c}$$

Geometric Series

For any matrix $\underline{\mathbf{A}}$,

$$\underline{\mathbf{I}} + \underline{\mathbf{A}} + \underline{\mathbf{A}}^2 + \cdots = \sum_{k=1}^{\infty} \underline{\mathbf{A}}^k = (\underline{\mathbf{I}} - \underline{\mathbf{A}})^{-1}, \qquad \|\underline{\mathbf{A}}\| < 1$$

$$\|(\underline{\mathbf{I}} - \underline{\mathbf{A}})^{-1}\| \leq (1 - \|\underline{\mathbf{A}}\|)^{-1}$$

If

$$f(\lambda) = \lambda^n + a_1\lambda^{n-1} + \cdots + a_n$$

then

$$f(\mathbf{A}) = \underline{\mathbf{A}}^n + a_1\underline{\mathbf{A}}^{n-1} + \cdots + a_n\underline{\mathbf{I}}$$

Eigenvalues of a Function of $\underline{\mathbf{A}}$

If λ_i is an eigenvalue of $\underline{\mathbf{A}}$ (denoted as $\underline{\mathbf{A}} \to \lambda_i$), then

$$\begin{aligned} \underline{\mathbf{A}}^{-1} &\longrightarrow \lambda_i^{-1}, & \underline{\mathbf{A}}^k &\longrightarrow \lambda_i^k \\ \mathbf{A}^{\mathrm{T}} &\longrightarrow \lambda_i, & f(\underline{\mathbf{A}}) &\longrightarrow f(\lambda_i) \\ & & \overline{\underline{\mathbf{A}}} &\longrightarrow \overline{\lambda}_i \end{aligned}$$

Sylvesters Theorem

For a quadratic form $[\underline{\boldsymbol{x}}^{\mathrm{T}}\,\underline{\mathbf{A}}\underline{\boldsymbol{x}}]$ to be positive definite it is necessary and sufficient that all the principal minors (along the main diagonal) of $\underline{\mathbf{A}}$ be positive.

DIAGONALIZATION OF MATRICES

An $n \times n$ matrix $\underline{\mathbf{A}}$ can be diagonalized if and only if it has n linearly independent eigenvectors. This is always possible if all of its n eigenvalues are all distinct. Then

$$\underline{\mathbf{A}}\underline{\boldsymbol{x}}_i = \lambda_i \underline{\boldsymbol{x}}_i \qquad (i = 1, \ldots, n)$$

where $\underline{\boldsymbol{x}}_i$ is the eigenvector belonging to λ_i; thus

$$\underline{\mathbf{A}}[\underline{\boldsymbol{x}}_1 \quad \cdots \quad \underline{\boldsymbol{x}}_n] = [\underline{\boldsymbol{x}}_1 \quad \cdots \quad \underline{\boldsymbol{x}}_n] \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

Let

$$\underline{\mathbf{P}} = [\underline{\boldsymbol{x}}_1 \quad \cdots \quad \underline{\boldsymbol{x}}_n] \qquad \text{(modal matrix)}$$

$$\underline{\mathbf{\Lambda}} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \qquad \text{(a diagonal matrix)}$$

Then

$$\underline{\mathbf{AP}} = \underline{\mathbf{P}}\underline{\mathbf{\Lambda}}, \qquad \underline{\mathbf{A}} = \underline{\mathbf{P}}^{-1}\underline{\mathbf{\Lambda}}\underline{\mathbf{P}}$$

In general two matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ are similar if one can find a nonsingular matrix $\underline{\mathbf{P}}$ such that

$$\underline{\mathbf{A}} = \underline{\mathbf{P}}^{-1}\underline{\mathbf{B}}\underline{\mathbf{P}}$$

Observe that:

1. Similar matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ have the same eigenvalues, equal determinants, and the same characteristic polynomials:

$$\underline{\mathbf{A}} \longrightarrow \lambda_i \quad \text{also means} \quad \underline{\mathbf{B}} \longrightarrow \lambda_i$$

$$\Delta_{\underline{\mathbf{A}}} = \Delta_{\underline{\mathbf{B}}}$$

$$\Delta_{\underline{\mathbf{A}}}(\lambda) = \Delta_{\underline{\mathbf{B}}}(\lambda)$$

2. Every Hermitian matrix is diagonalizable, and its modal matrix $\underline{\mathbf{P}}$ is unitary:

$$\underline{\mathbf{P}}^*\underline{\mathbf{P}} = \underline{\mathbf{I}}$$

3. If $\underline{\mathbf{A}}_c$ is a *companion matrix*

$$\underline{\mathbf{A}}_c = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \ldots & -a_1 \end{bmatrix}$$

then

$$\Delta_{\mathbf{A}_c}(\lambda) = a_0\lambda^n + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \cdots + a_n = \prod_{i=1}^{n}(\lambda - \lambda_i)$$

If the eigenvalues of $\underline{\mathbf{A}}_c$ are distinct, then

$$\underline{\mathbf{P}} = \underline{\mathbf{V}}_n = \begin{bmatrix} 1 & \cdots & 1 \\ \lambda_1 & \cdots & \lambda_n \\ \vdots & & \vdots \\ \lambda_1^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix}$$

the *Vandermonde matrix,* is nonsingular, and

$$\det \underline{\mathbf{V}}_n = \prod_{i=2}^{n}\left(\prod_{j=1}^{i-1}(\lambda_i - \lambda_j)\right)$$

4. If a matrix has eigenvalues of multiplicity greater than one, then for diagonalization these eigenvalues should induce the same number of linearly independent eigenvectors as the multiplicity; otherwise the similarity transformation produces not the diagonal but the *Jordan* form. As discussed earlier, we produce a set of generalized eigenvectors for the same eigenvalues, which are linearly independent and transform a matrix into *Jordan canonical form.*

THE JORDAN CANONICAL FORM

When the characteristic polynomial of a matrix has multiple roots, it may not be possible to diagonalize the matrix. Nevertheless, it is possible to transform the matrix into a canonical form called *Jordan canonical form,* via similarity transformations. We shall limit ourself to the procedure for arriving at this canonical form.

Let

$$\Delta_{\underline{\mathbf{A}}}(\lambda) = |(\lambda\underline{\mathbf{I}} - \underline{\mathbf{A}})| = \prod_{i=1}^{r}(\lambda - \lambda_i)^{k_i}, \qquad \sum_{i=1}^{r} k_i = n$$

The matrix $\underline{\mathbf{A}}$ can be transformed to a matrix $\underline{\mathbf{J}}$ with canonical superboxes $\underline{\mathbf{J}}_i$ $(i = 1, \ldots, r)$: $\underline{\mathbf{J}} = \sum_{i=1}^{r} \underline{\mathbf{J}}_i$. These superboxes $\underline{\mathbf{J}}_i$ are further divided into boxes $\underline{\mathbf{J}}_{ij}$ $(j = 1, \ldots, r_i;\ r_i \le k_i)$: $\underline{\mathbf{J}}_i = \sum_{j=1}^{r_i} \underline{\mathbf{J}}_{ij}$. Namely,

$$\underline{\mathbf{J}} = \begin{bmatrix} \underline{\mathbf{J}}_1 & & & \\ & \ddots & & \underline{\mathbf{0}} \\ & & \underline{\mathbf{J}}_i & \\ & \underline{\mathbf{0}} & & \ddots \\ & & & \underline{\mathbf{J}}_r \end{bmatrix}$$

$$\underline{\mathbf{J}}_i = \begin{bmatrix} \underline{\mathbf{J}}_{i1} & & & \\ & \ddots & & \\ & & \underline{\mathbf{J}}_{ij} & \\ & & & \ddots \\ & & & \underline{\mathbf{J}}_{ir_i} \end{bmatrix}$$

The dimension of the box $\underline{\mathbf{J}}_i$ is $k_i \times k_i$ ($\sum_{i=1}^{r} k_i = n$). The dimension of the ijth box within $\underline{\mathbf{J}}_i$ is $l_{ij} \times l_{ij}$ ($\sum_{j=1}^{r_i} l_{ij} = k_i$).

The following procedure is used to determine

$$\begin{aligned} &\lambda_i && (i = 1, \ldots, r) \\ &k_i && (i = 1, \ldots, r) \\ &l_{ij} && (i = 1, \ldots, r; \quad j = 1, \ldots, r_i) \end{aligned}$$

Step 1. Determine the characteristic polynomial $\Delta_{\underline{\mathbf{A}}}(\lambda)$ of $\underline{\mathbf{A}}$ [with order $\det(\lambda\underline{\mathbf{I}} - \underline{\mathbf{A}})$] and its roots λ_i ($i = 1, \ldots, r$).

Step 2. Determine the multiplicity indices k_i ($i = 1, \ldots, r$) such that $(\lambda - \lambda_i)^{k_i}$ is a factor of $\Delta_{\underline{\mathbf{A}}}(\lambda)$ but $(\lambda - \lambda_i)^{k_i+1}$ is not.

Step 3. Consider all the minors of order $n - j$ ($i = 1, \ldots, r; j = 1, \ldots, r_i$). If the greatest common divisor (gcd) of any one of these minors contains a factor $(\lambda - \lambda_i)^{k_{ij}}$ but not $(\lambda - \lambda_i)^{k_{ij+1}}$, then $l_{ij} = k_{i,j-1} - k_{i,j}$ ($i = 1, \ldots, r; j = 1, \ldots, r_i; k_{i,0} = k_i$)

The minors of order $n - k_i - 1$ contain no factor $\lambda - \lambda_i$. each Jordan subbox $\underline{\mathbf{J}}_{ij}$ appears as

$$\underline{\mathbf{J}}_{ij} = \begin{bmatrix} \lambda_i & 1 & 0 & \ldots & 0 \\ 0 & \lambda_i & 1 & \ldots & 0 \\ 0 & 0 & \lambda_i & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_i \end{bmatrix}, \qquad l_{ij} \times l_{ij}$$

In practice we use the method of *elementary divisors* to arrive at the structure of Jordan canonical form. We transform $\underline{\mathbf{A}}$ into Jordan form $\underline{\mathbf{J}}$ via a similarity transformation $\underline{\mathbf{P}}$:

$$\underline{\mathbf{A}} = \underline{\mathbf{P}}\,\underline{\mathbf{J}}\,\underline{\mathbf{P}}^{-1}, \qquad \underline{\mathbf{A}}^k = \underline{\mathbf{P}}\,\underline{\mathbf{J}}^k\,\underline{\mathbf{P}}^{-1}$$

The modal matrix $\underline{\mathbf{P}}$ is made up of the chain of generalized eigenvectors

$$\underline{\boldsymbol{x}}_{ij},\ (\lambda_i\underline{\mathbf{I}} - \underline{\mathbf{A}}\,\underline{\boldsymbol{x}}_{ij}),\ \ldots,\ (\lambda_i\underline{\mathbf{I}} - \underline{\mathbf{A}}^{r_i-1})\underline{\boldsymbol{x}}_{ij} \qquad (i = 1, \ldots, r, \quad j = 1, \ldots, r_i)$$

Every square matrix $\underline{\mathbf{A}}$ can be transformed into Jordan form.

The *minimal polynomial* of $\underline{\mathbf{J}}$ (or $\underline{\mathbf{A}}$) is $\underline{\mathbf{P}}_m(\lambda) = \prod_{i=1}^{r_{ij}}(\lambda - \lambda_i)^{l_{i1}}$, where l_{i1} is the size of the largest Jordan subbox associated with λ_i

Using Dg for block-diagonal matrices, we have

$$\begin{aligned} \underline{\mathbf{J}} &= \mathrm{Dg}[\underline{\mathbf{J}}_1, \underline{\mathbf{J}}_2, \ldots, \underline{\mathbf{J}}_i, \ldots, \underline{\mathbf{J}}_n] \\ \underline{\mathbf{J}}_i &= \mathrm{Dg}[\underline{\mathbf{J}}_{i1}, \underline{\mathbf{J}}_{ij}, \ldots, \underline{\mathbf{J}}_{ir_i}] \qquad (i = 1, \ldots, r) \end{aligned}$$

$(\lambda - \lambda_i)^{l_{ij}}$ are known as *elementary divisors* of $\underline{\mathbf{A}}$ ($i = 1, \ldots, r, j = 1, \ldots, r_i$)

$(\lambda_i\underline{\mathbf{I}} - \underline{\mathbf{A}})$ acts as an *elevator matrix.* It raises an eigenvector to the next higher eigenvector until the last vector in the chain is reached, and then annihilates it.

CAYLEY–HAMILTON THEOREM

This remarkable theorem states: "A matrix satisfies its own characteristic equation". Specifically, if

$$\Delta_{\underline{\mathbf{A}}}(\lambda) = p(\lambda) = |(\lambda I - \underline{\mathbf{A}})| = \lambda^n + a_1\lambda^{n-1} + \cdots$$

where $\underline{\mathbf{A}}$ is an $n \times n$ matrix, then $\underline{\mathbf{A}}\underline{x} \in \mathbf{E}_n$ when $\underline{\boldsymbol{x}} \in \mathbf{E}_n$. Then $p(\underline{\mathbf{A}})\underline{\boldsymbol{x}} = \underline{\mathbf{0}}$, implying

$$p(\underline{\mathbf{A}}) \equiv \underline{\mathbf{A}}^n + a_1\underline{\mathbf{A}}^{n-1} + \cdots + a_n\underline{\mathbf{I}} = \underline{\mathbf{0}}$$

Proof:

$$[\underline{\mathbf{A}}(\lambda)]^{-1} = (\lambda\underline{\mathbf{I}} - \underline{\mathbf{A}})^{-1} = \frac{1}{\Delta_{\underline{\mathbf{A}}}(\lambda)}\underline{\mathbf{A}}^*(\lambda) = \frac{1}{p(\lambda)}\underline{\mathbf{B}}(\lambda) \tag{12}$$

where $\underline{\mathbf{B}}(\lambda) = \underline{\mathbf{A}}^*(\lambda)$ is a polynomial matrix in λ of degree $(n - 1)$

$$\underline{\mathbf{B}}(\lambda) \equiv \underline{\mathbf{B}}_1\lambda^{n-1} + \underline{\mathbf{B}}_2\lambda^{n-2} + \cdots + \underline{\mathbf{B}}_n = \sum_{i=0}^{n-1} \underline{\mathbf{B}}_{n-i}\lambda^i \tag{13}$$

From (12) and (13),

$$(\lambda\underline{\mathbf{I}} - \underline{\mathbf{A}})\underline{\mathbf{B}}(\lambda) = p(\lambda)\underline{\mathbf{I}}$$

Equating powers of λ on both sides,

$$\begin{aligned} \underline{\mathbf{0}} - \underline{\mathbf{A}}\underline{\mathbf{B}}_\mathbf{n} &= a_n\underline{\mathbf{I}} \\ \underline{\mathbf{B}}_\mathbf{n} - \underline{\mathbf{A}}\underline{\mathbf{B}}_\mathbf{n-1} &= a_{n-1}\underline{\mathbf{I}} \\ &\vdots \\ \underline{\mathbf{B}}_2 - \underline{\mathbf{A}}\underline{\mathbf{B}}_1 &= a_1\underline{\mathbf{I}} \\ \underline{\mathbf{B}}_1 - \underline{\mathbf{0}} &= \underline{\mathbf{I}} \end{aligned} \tag{14}$$

Multiplying these equations by $\underline{\mathbf{I}}, \underline{\mathbf{A}}, \ldots, \underline{\mathbf{A}}^n$ respectively and adding,

$$\underline{\mathbf{0}} \equiv \underline{\mathbf{A}}^n + a_1\underline{\mathbf{A}}^{n-1} + \cdots + a_n\underline{\mathbf{I}} \equiv p(\underline{\mathbf{A}}) \tag{15}$$

This theorem is very significant in system theory, for it implies that all matrices $\underline{\mathbf{A}}^k$ ($k \geq n$) can be expressed as a linear combination of matrices $\underline{\mathbf{A}}^j$ ($j < n$).

COMPUTATION OF A POLYNOMIAL FUNCTION OF THE MATRIX $\underline{\mathbf{A}}$

Let

$$F(\underline{\mathbf{A}}) = \sum_{k=1}^{m} c_k\underline{\mathbf{A}}^k, \qquad m \geq n \tag{16}$$

where λ_i ($i = 1, \ldots, n$) are eigenvalues of $\underline{\mathbf{A}}$. Then

$$\frac{F(\lambda)}{\Delta_{\underline{\mathbf{A}}}(\lambda)} = Q(\lambda) + \frac{R(\lambda)}{\Delta_{\underline{\mathbf{A}}}(\lambda)} \tag{17}$$

by long division, where $R(\lambda)$ is polynomial of degree less than n. Then

$$F(\lambda) = Q(\lambda)\Delta_{\underline{\mathbf{A}}}(\lambda) + R(\lambda)$$
$$F(\lambda_i) = R(\lambda_i), \quad \Delta_{\underline{\mathbf{A}}}(\lambda_i) = 0, \qquad i = 1, \ldots, n$$

Compute the coefficients of $R(\lambda_i)$ from $F(\lambda_i)$. If λ_i is an eigenvalue of multiplicity m_i, then not only does $\Delta_{\underline{\mathbf{A}}}(\lambda_i) = 0$, but the first $m_i - 1$ derivatives of $\Delta_{\underline{\mathbf{A}}}(\lambda_i)$ with respect to λ computed at $\lambda = \lambda_i$ also vanish, resulting in

$$\frac{d^k}{d\lambda^k} F(\lambda)\bigg|_{\lambda=\lambda_i} = \frac{d^k}{d\lambda^k} R(\lambda)\bigg|_{\lambda=\lambda_i} \qquad (k = 0, 1, \ldots, m_{i-1})$$

For the matrix exponential we have

$$e^{\underline{\mathbf{A}}t} = \sum_{k=0}^{\infty} \frac{\underline{\mathbf{A}}^k t^k}{k!}$$

(not generally recommended for computing),

$$e^{\underline{\mathbf{A}}t} = \sum_{i=0}^{n-1} \alpha_i(t)\underline{\mathbf{A}}^i, \qquad \alpha_0(0) = 1, \quad \alpha_i(0) = 0 \; (i = 2, \ldots)$$

and

$$e^{(\underline{\mathbf{A}}+\underline{\mathbf{B}})t} = e^{\underline{\mathbf{A}}t} e^{\underline{\mathbf{B}}t} \qquad \text{if} \quad \underline{\mathbf{AB}} = \underline{\mathbf{BA}}$$

For a series

$$g(\lambda) = \sum_{k=0}^{\infty} g_k \lambda^k$$

For a series $|\lambda| \le r \le 1$ implies convergence.

$$g(\underline{\mathbf{A}}) = \sum_{k=0}^{\infty} g_k \underline{\mathbf{A}}^k, \qquad \underline{\mathbf{A}} \text{ with eigenvalues } \lambda_i$$

$|\lambda_i| \le r \le 1$ ($i = 1, 2, \ldots, n$) implies convergence.

From complex integration

$$f(\boldsymbol{A}) = \frac{1}{2\pi j} \oint_c (\lambda \underline{\mathbf{I}} - \underline{\mathbf{A}})^{-1} f(\lambda)\, d\lambda, \qquad |\lambda_i| \le c$$

For

$$\underline{\mathbf{J}} = \begin{bmatrix} \lambda_1 & 1 & & \\ & \lambda_1 & 1 & \\ & & \lambda_1 & \ddots \\ & & & \ddots \end{bmatrix}$$

We obtain

$$f(\underline{\mathbf{J}}) = \begin{bmatrix} f(\lambda_1) & \dfrac{f'(\lambda_1)}{1!} & \dfrac{f''(\lambda_1)}{2!} & \cdots \\ & f(\lambda_1) & \dfrac{f'(\lambda_1)}{1!} & \ddots \\ & & f(\lambda_1) & \\ & & & \ddots \end{bmatrix}$$

$$e^{\underline{\mathbf{J}}t} = \begin{bmatrix} e^{\lambda_1 t} & te^{\lambda_1 t} & t^2 e^{\lambda_1 t} & \cdots \\ & e^{\lambda_1 t} & te^{\lambda_1 t} & \\ & & e^{\lambda_1 t} & \\ & & & \ddots \end{bmatrix}$$

$$\underline{\mathbf{\Lambda}} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

$$f(\underline{\mathbf{\Lambda}}) = \begin{bmatrix} f(\lambda_1) & & & \\ & f(\lambda_2) & & \\ & & \ddots & \\ & & & f(\lambda_n) \end{bmatrix}$$

$$e^{\underline{\mathbf{\Lambda}}t} = \begin{bmatrix} e^{\lambda_1 t} & & & \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ & & & e^{\lambda_n t} \end{bmatrix}$$

$$\underline{\mathbf{A}} = \underline{\mathbf{P}}^{-1}\underline{\mathbf{\Lambda}}\underline{\mathbf{P}}, \qquad f(\underline{\mathbf{A}}) = \underline{\mathbf{P}}^{-1}\mathbf{f}(\underline{\mathbf{\Lambda}})\underline{\mathbf{P}}$$
$$\underline{\mathbf{A}} = \underline{\mathbf{S}}^{-1}\underline{\mathbf{J}}\underline{\mathbf{S}}, \qquad f(\underline{\mathbf{J}}) = \underline{\mathbf{S}}^{-1}\mathbf{f}(\underline{\mathbf{J}})\underline{\mathbf{S}}$$

If an $n \times n$ matrix $\underline{\mathbf{A}}$ has minimal polynomial of degree $m < n$, then

$$e^{\underline{\mathbf{A}}t} = \alpha_0(t)\underline{\mathbf{I}} + \alpha_1(t)\underline{\mathbf{A}} \cdots \alpha_{m-1}(t)\underline{\mathbf{A}}^{m-1}$$

where $\alpha_j(t)$ ($j = 0, \ldots m - 1$) can be computed from the eigenvalues, distinct or multiple.

A matrix $\underline{\mathbf{A}}$ is called *stable* if the real parts of its eigenvalues λ_i ($i = 1, \ldots, n$) are negative.

For the *Riccati equation*

$$\underline{\mathbf{AS}} + \underline{\mathbf{SA}}^{\mathrm{T}} = -\underline{\mathbf{Q}}, \qquad \underline{\mathbf{S}}, \; \underline{\mathbf{Q}} \text{ symmetric}$$

we have the solution

$$\underline{\mathbf{S}} = \int_{\mathbf{0}}^{\infty} \underline{\mathbf{e}}^{\underline{\mathbf{A}}t} \underline{\mathbf{Q}} \underline{\mathbf{e}}^{\underline{\mathbf{A}}^{\mathrm{T}}t}\, dt$$

COMPANION MATRICES

A companion matrix has the form

$$\underline{\mathbf{A}}_c = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix}$$

$$\Delta_{\underline{\mathbf{A}}_c}(\lambda) = |\lambda \underline{\mathbf{I}} - \underline{\mathbf{A}}_c| = \lambda^n + a_1\lambda^{n-1} + \cdots + a_n = p(\lambda)$$

The polynomial $\Delta_{\underline{\mathbf{A}}}(\lambda)$ can be associated with the companion matrix $\underline{\mathbf{A}}_c$.

The following special properties are associated with companion matrices:

1. If λ_i is an eigenvalue of multiplicity one (distinct), the associated eigenvector is

$$\underline{\boldsymbol{p}}_i^{\mathrm{T}} = [1 \quad \lambda_i \quad \lambda_i^2 \quad \cdots \quad \lambda_i^{n-1}]$$

2. If λ_i is an eigenvalue of multiplicity $k \le n$ [$(\lambda - \lambda_i)^k$ is, factor of $\Delta_{\underline{\mathbf{A}}}(\lambda)$ but $(\lambda - \lambda_i)^{k+1}$ is not], then this eigenvalue has k *generalized* eigenvectors and one and only one Jordan block of size $k \times k$ belonging to the eigenvalue λ_i. This implies that companion matrix is nonderogatory, and we have

$$\begin{aligned} \underline{\boldsymbol{p}}_{i1}^{\mathrm{T}} &= [1 \quad \lambda_1 \quad \lambda_1^2 \quad \dots \quad \lambda_1^{n-1}] \\ \underline{\boldsymbol{p}}_{i1}^{\mathrm{T}} &= [0 \quad 1 \quad 2\lambda_1 \quad \dots \quad (n-1)\lambda_1^{n-2}] \\ &\vdots \\ \underline{\boldsymbol{p}}_{in}^{\mathrm{T}} &= \left[0 \quad 0 \quad 0 \quad \dots \quad \prod_{j=1}^{k-1}(n-j)\lambda_i^{n-k}\right] \end{aligned}$$

3. An nth-order Linear differential equation.

$$x^{(n)} + a_1 x^{(n-1)} \dots + a_{n-1}\dot{x} + a_n x = 0$$

can be written as

$$\dot{\underline{\boldsymbol{x}}} = \underline{\mathbf{A}}_c \underline{\boldsymbol{x}}$$

where $\underline{\mathbf{A}}_c$ is a companion matrix.

4. A matrix $\underline{\mathbf{A}}$ is similar to the companion matrix $\underline{\mathbf{A}}_c$ [of its characteristic polynomial $\Delta_{\underline{\mathbf{A}}}(\lambda)$] if and only if the minimal and the characteristic polynomial of $\underline{\mathbf{A}}$ are the same. This implies $\underline{\mathbf{A}}$ is nonderogatory.

CHOLESKY DECOMPOSITION (ALSO KNOWN AS LU DECOMPOSITION)

This is a convenient scheme for machine computation of $\underline{\mathbf{A}}\underline{\boldsymbol{x}} = \underline{\boldsymbol{b}}$, where $\underline{\mathbf{A}}$ is $n \times n$ of rank n, and $\underline{\boldsymbol{b}}$ is $n \times 1$.

Write $\underline{\mathbf{A}} = (a_{ij})$ as $\underline{\mathbf{A}} = \underline{\mathbf{L}}\underline{\mathbf{U}}$, where $\underline{\mathbf{L}}$ is lower triangular and $\underline{\mathbf{U}}$ is upper triangular:

$$\underline{\mathbf{L}} = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}, \qquad \underline{\mathbf{U}} = \begin{bmatrix} 1 & c_{12} & \dots & c_{1n} \\ 0 & 1 & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

l_{ij} and u_{ij} are computed as

$$\begin{aligned} l_{i1} &= a_{i1},\ u_{1j} = \frac{a_{1j}}{l_{11}} && (i = 1, \dots, n;\ j = 1, \dots, n) \\ l_{ij} &= a_{ij} - \sum_{k=1}^{j-1}(l_{ik}u_{kj}) && (i \ge j < 1) \\ u_{ij} &= \frac{1}{l_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}\right) && (j > i > 1), \qquad u_{ii} = 1 \end{aligned}$$

Knowing $\underline{\mathbf{L}}$ and $\underline{\mathbf{U}}$, solve the two sets of equations

$$\underline{\mathbf{U}}\underline{\mathbf{x}} = \underline{\boldsymbol{y}}, \qquad \underline{\mathbf{L}}\underline{\boldsymbol{y}} = \underline{\mathbf{b}}$$

where $\underline{\mathbf{A}}$ is symmetric. The computation of $\underline{\mathbf{U}}$ is simplified as

$$u_{ij} = \frac{1}{l_{ii}} l_{ji} \qquad (i \le j)$$

JACOBI AND GAUSS–SEIDEL METHODS

When all the diagonal elements of $\underline{\mathbf{A}}$ are nonzero, we can decompose $\underline{\mathbf{A}}$ as

$$\underline{\mathbf{A}} = \underline{\mathbf{L}} + \underline{\mathbf{D}} + \underline{\mathbf{U}}$$

with

$\underline{\mathbf{U}}$ upper triangular with zero on the diagonal
$\underline{\mathbf{L}}$ lower triangular with zero on the diagonal
$\underline{\mathbf{D}}$ diagonal

The iterative schemes for solving $\underline{\mathbf{A}}\underline{\boldsymbol{x}} = \underline{\boldsymbol{b}}$, with initial guess $\underline{\boldsymbol{x}}^{(0)}$, are

$$\underline{\boldsymbol{x}}^{(i+1)} = \underline{\mathbf{D}}^{-1}\underline{\boldsymbol{b}} - \underline{\mathbf{D}}^{-1}(\underline{\mathbf{L}} + \underline{\mathbf{U}})\underline{\boldsymbol{x}}^{(i)} \quad (i = 0, 1, 2, 3, \dots) \quad \text{(Jacobi)}$$

$$\underline{\boldsymbol{x}}^{(i+1)} = (\underline{\mathbf{L}} + \underline{\mathbf{D}})^{-1}\underline{\boldsymbol{b}} - (\underline{\mathbf{L}} + \underline{\mathbf{D}})^{-1}\underline{\mathbf{U}}\underline{\boldsymbol{x}}^{(i)} \quad \text{(Gauss–Seidel)}$$

LEAST-SQUARES BEST-FIT PROBLEM (ALSO KNOWN AS THE PSEUDOINVERSE PROBLEM)

Given:

$\underline{\mathbf{A}}\underline{\boldsymbol{x}} = \underline{\boldsymbol{b}}$, subject to the condition $\underline{\mathbf{B}}\underline{\boldsymbol{x}} = 0$
$\underline{\mathbf{A}}$ $n \times p$, rank $\underline{\mathbf{A}} = p$;
$\underline{\mathbf{B}}$ $r \times p$, rank $\underline{\mathbf{B}} = r$ $(r \le p \le n)$

we are to solve for $\underline{\boldsymbol{x}}$.

Define $(\underline{\mathbf{A}}^{\mathrm{T}}\underline{\mathbf{A}})^{-1}\underline{\mathbf{A}}^{\mathrm{T}} = \underline{\mathbf{A}}^{+}$ (*pseudoinverse*), $(\underline{\mathbf{A}}^{\mathrm{T}}\underline{\mathbf{A}})^{-1}\underline{\mathbf{B}}^{\mathrm{T}} = \underline{\mathbf{B}}_1$. The least-squares solution is

$$\hat{\underline{\boldsymbol{x}}} = [\underline{\mathbf{A}}^{+} - \underline{\mathbf{B}}_1(\underline{\mathbf{B}}\underline{\mathbf{B}}_1)^{-1}\underline{\mathbf{B}}\underline{\mathbf{A}}^{+}]\underline{\boldsymbol{b}}$$

HERMITIAN (OR SYMMETRIC REAL) MATRICES AND DEFINITE FUNCTIONS

1. Let $\underline{\mathbf{A}}$ be Hermitian. If for all $\underline{\boldsymbol{x}}$ we have

$$\overline{\underline{\mathbf{x}}}^{\mathrm{T}}\underline{\mathbf{A}}\underline{\boldsymbol{x}} = \underline{\boldsymbol{x}}^{*}\underline{\mathbf{A}}\underline{\boldsymbol{x}} > 0$$

then $\underline{\mathbf{A}}$ is *positive definite*. If for all $\underline{\boldsymbol{x}}$ we have

$$\underline{\boldsymbol{x}}^{*}\underline{\mathbf{A}}\underline{\boldsymbol{x}} \geq 0$$

then $\underline{\mathbf{A}}$ is *positive semidefinite*. If for some $\underline{\boldsymbol{x}}$ we have $\underline{\boldsymbol{x}}^{*}\mathrm{A}\underline{x} > 0$ and for other $\underline{\boldsymbol{x}}$ we have $\underline{\boldsymbol{x}}^{*}\mathrm{A}\underline{x} < 0$, then $\underline{\mathbf{A}}$ is *indefinite*.

2. Hermitian (or symmetric real) matrices have distinct eigenvalues, and their eigenvectors are mutually orthogonal. If in addition the matrix is positive definite, then all its eigenvalues are necessarily positive. If λ_1 is the largest and λ_n is the smallest eigenvalue of $\underline{\mathbf{A}}$, then

$$\lambda_n(\underline{\boldsymbol{x}}^{*}\underline{\boldsymbol{x}}) \leq (\underline{\boldsymbol{x}}^{*}\underline{\mathbf{A}}\underline{\boldsymbol{x}}) \leq \lambda_1(\underline{\boldsymbol{x}}^{*}\underline{\boldsymbol{x}})$$

In fact, any Hermitian (or real symmetric) matrix can be diagonalized by a similarity transformation $\underline{\mathbf{P}}$ in which all the columns are mutually orthonormal (called a *unitary* matrix). All the eigenvalues of a Hermitian (or symmetric real) positive definite matrix are strictly positive. The coefficients of the characteristic polynomial $|(\lambda\underline{\mathbf{I}} - \underline{\mathbf{A}})|$ of a positive definite matrix alternate in sign, yielding a necessary and sufficient condition for positive definiteness. The principal diagonal minors of the determinant of a positive definite Hermitian matrix must be strictly positive. If two Hermitian matrices commute, then they can be simultaneously diagonalized.

3. For the *simultaneous diagonalization* of two real matrices $\underline{\mathbf{R}} > 0$ and $\underline{\mathbf{Q}} \leq 0$, choose a nonsingular $\underline{\mathbf{W}}$, the *square-root matrix of* $\underline{\mathbf{R}}$, such that $\underline{\mathbf{R}} = \underline{\mathbf{W}}^{\mathrm{T}}\underline{\mathbf{W}}$. Choose an orthogonal matrix $\underline{\mathbf{O}}$ such that $\underline{\mathbf{O}}^{\mathrm{T}}\underline{\overline{\mathbf{W}}}^{\mathrm{T}}\underline{\mathbf{Q}}\underline{\mathbf{W}}\underline{\mathbf{O}} = \underline{\mathbf{D}}$ ($\underline{\mathbf{D}} \leq 0$ is a diagonal matrix).
4. *Liapunov Stability Theorem.* Given an $n \times n$ real matrix $\underline{\mathbf{A}}$ with eigenvalues λ_i, if there exists a matrix $\underline{\mathbf{S}} \geq 0$ such that $\underline{\mathbf{A}}^{\mathrm{T}}\underline{\mathbf{S}} + \underline{\mathbf{S}}\underline{\mathbf{A}} \leq 0$, then $\mathrm{Re}\ \lambda_i < 0$ ($i = 1, \ldots, n$).

SOME USEFUL FACTS AND IDENTITIES

1. $(\underline{\mathbf{A}}^{-1} - \underline{\mathbf{B}}^{-1})^{-1} = \underline{\mathbf{A}} - \underline{\mathbf{C}}\underline{\mathbf{A}}$, where $\underline{\mathbf{C}} = (\underline{\mathbf{A}} - \underline{\mathbf{B}})^{-1}$.
2. $(\underline{\mathbf{I}} - \underline{\mathbf{A}}\underline{\mathbf{B}})^{-1} = \underline{\mathbf{I}} - \underline{\mathbf{A}}(\underline{\mathbf{I}} + \underline{\mathbf{B}}\underline{\mathbf{A}})^{-1}\underline{\mathbf{B}}$, $\underline{\mathbf{B}}\underline{\mathbf{A}}$ nonsingular (*Woodbury's form*).

If $\underline{\mathbf{B}} = \underline{\boldsymbol{x}}$, an $n \times 1$ vector, and $\underline{\mathbf{A}} = \underline{\boldsymbol{y}}^{\mathrm{T}}$, a $1 \times n$ column vector, then

$$(\underline{\mathbf{I}} + \underline{\boldsymbol{x}}\underline{\boldsymbol{y}}^{\mathrm{T}})^{-1} = \underline{\mathbf{I}} - \frac{1}{\beta}\underline{\boldsymbol{x}}\underline{\boldsymbol{y}}^{\mathrm{T}}, \qquad \beta = (1 + \underline{\mathbf{x}}^{\mathrm{T}}\underline{\mathbf{y}})$$

(the *Sherman–Morrison formula*),

$$(\underline{\mathbf{A}} + \underline{\boldsymbol{x}}\underline{\boldsymbol{y}}^{\mathrm{T}})^{-1} = \underline{\mathbf{A}}^{-1} - \frac{1}{\alpha}\underline{\mathbf{A}}^{-1}\underline{\boldsymbol{x}}\underline{\boldsymbol{y}}^{\mathrm{T}}\underline{\mathbf{A}}^{-1}, \qquad \alpha = 1 + T_r(\underline{\boldsymbol{x}}\underline{\boldsymbol{y}}^{\mathrm{T}}\underline{\mathbf{A}}^{-1})$$

3. Suppose $\underline{\mathbf{A}} = \sum_{i=1}^{n} \lambda_i \underline{\mathbf{x}}_i\underline{\mathbf{y}}_i^{\mathrm{T}}$ is an $n \times n$ real matrix with distinct eigenvalues λ_i ($i = 1, \ldots, n$), $\underline{\mathbf{x}}_i$ the corresponding eigenvectors of $\underline{\mathbf{A}}$, and $\underline{\mathbf{y}}_i$ the corresponding eigenvectors of $\underline{\mathbf{A}}^{\mathrm{T}}$. If $\underline{\mathbf{A}}$ is Hermitian, then we have $\underline{\mathbf{y}}_i = \underline{\mathbf{x}}_i^{\mathrm{T}} = \underline{\mathbf{x}}_i^{*}$.
4. $\underline{\mathbf{A}} = \underline{\boldsymbol{x}}\underline{\boldsymbol{y}}^{\mathrm{T}}$ implies $\underline{\mathbf{A}}$ is of rank one.
5. *Gerschgorin Circles*. Given an $n \times n$ nonsingular matrix $\underline{\mathbf{A}} = (a_{ij})$ with eigenvalues λ_k ($k = 1, \ldots, n$), then

$$|a_{ij}| > \sum_{i \neq j} |a_{ij}| \qquad (i = 1, \ldots, n)$$

$$|\lambda_k - a_{ii}| \leq \sum_{i \neq j} (a_{ij}) \qquad \text{for at least one } k, \quad i = 1, \ldots, n$$

6. *Bordering Matrices.* These matrices are useful in sequential filtering problems. Let

$$\tilde{\underline{\mathbf{A}}} = \begin{bmatrix} \underline{\mathbf{A}} & \underline{\boldsymbol{x}} \\ \underline{\boldsymbol{y}}^{\mathrm{T}} & \alpha \end{bmatrix}$$

Then

$$\tilde{\underline{\mathbf{A}}}^{-1} = \begin{bmatrix} \left(\underline{\mathbf{A}} - \frac{1}{\alpha}\underline{\boldsymbol{x}}\underline{\boldsymbol{y}}^{\mathrm{T}}\right)^{-1} & -\frac{1}{\beta}\underline{\mathbf{A}}^{-1}\underline{\boldsymbol{x}} \\ -\frac{1}{\beta}\underline{\boldsymbol{y}}^{\mathrm{T}}\underline{\mathbf{A}}^{-1} & -\frac{1}{\beta} \end{bmatrix}, \qquad \beta = \alpha - \underline{\boldsymbol{y}}^{\mathrm{T}}\underline{\mathbf{A}}^{-1}\underline{\boldsymbol{x}}$$

If $\underline{\mathbf{A}}$ is Hermitian (diagonalizable, $\underline{\mathbf{A}} = \underline{\mathbf{P}}\underline{\Lambda}\underline{\mathbf{P}}^{*}$, $\underline{\mathbf{P}}$ unitary) and $\underline{\boldsymbol{y}} = \underline{\boldsymbol{x}}$, then the eigenvalues $\tilde{\lambda}$ of $\tilde{\underline{\mathbf{A}}}$ are computed from

$$\underline{\boldsymbol{x}}^{*}\underline{\mathbf{P}}(\tilde{\lambda}\underline{\mathbf{I}} - \underline{\Lambda})^{-1}\underline{\mathbf{P}}^{*}\underline{\boldsymbol{x}} = \tilde{\lambda} - \alpha \qquad (\tilde{\underline{\mathbf{A}}} \text{ is also Hermitian})$$

If $\underline{\mathbf{A}} > 0$ and $\alpha > \underline{\boldsymbol{y}}^{\mathrm{T}}\underline{\mathbf{A}}^{-1}\underline{\boldsymbol{x}}$, $\underline{\boldsymbol{y}} = \underline{\boldsymbol{x}}$, then $\tilde{\underline{\mathbf{A}}} > 0$.

7. *Kronecker Product.* For $m \times n$ $\underline{\mathbf{A}}$ and $p \times q$ $\underline{\mathbf{B}}$,

$$\underline{\mathbf{A}} \otimes \underline{\mathbf{B}} = \begin{bmatrix} a_{11}\underline{\mathbf{B}} & \cdots & a_{1n}\underline{\mathbf{B}} \\ \vdots & & \vdots \\ a_{m1}\underline{\mathbf{B}} & \cdots & a_{mn}\underline{\mathbf{B}} \end{bmatrix}$$

is an $mp \times nq$ matrix called the *Kronecker product.* There are mn blocks of this matrix, and the ijth block is $a_{ij}\underline{\mathbf{B}}$. We have

$$(\underline{\mathbf{A}} \otimes \underline{\mathbf{B}})(\underline{\mathbf{C}} \otimes \underline{\mathbf{D}}) = (\underline{\mathbf{A}}\underline{\mathbf{C}} \otimes \underline{\mathbf{B}}\underline{\mathbf{C}})$$

provided $\underline{\mathbf{A}}\underline{\mathbf{C}}$ and $\underline{\mathbf{B}}\underline{\mathbf{D}}$ exist

Furthermore,

$$(\underline{\mathbf{A}}\otimes\underline{\mathbf{B}})^{\mathrm{T}}=(\underline{\mathbf{A}}^{\mathrm{T}}\otimes\underline{\mathbf{B}}^{\mathrm{T}}),\qquad(\underline{\mathbf{A}}\otimes\underline{\mathbf{B}})^{-1}=(\underline{\mathbf{A}}^{-1}\otimes\underline{\mathbf{B}}^{-1})$$

Finally, we can express the Liapunov matrix equation

$$\underline{\mathbf{AS}}+\underline{\mathbf{A}}^{\mathrm{T}}=\underline{\mathbf{Q}} \tag{18}$$

(all matrices are $n\times n$) in Kronecker product form. $\underline{\mathbf{S}}$ and $\underline{\mathbf{Q}}$ are symmetric. We leave $\underline{\mathbf{A}}$ alone and express $\underline{\mathbf{Q}}$ and $\underline{\mathbf{S}}$ as direct sums of n vectors each, yielding

$$\underline{\mathbf{Q}}=[\underline{\boldsymbol{q}}_1|\cdots|\underline{\boldsymbol{q}}_n],\quad \underline{\mathbf{S}}=[\underline{\boldsymbol{s}}_1|\ldots|\underline{\boldsymbol{s}}_n]$$

$$\underline{\mathbf{q}}=\begin{bmatrix}\underline{\boldsymbol{q}}_1\\ \vdots\\ \underline{\boldsymbol{q}}_n\end{bmatrix},\qquad \underline{\mathbf{s}}=\begin{bmatrix}\underline{\boldsymbol{s}}_1\\ \vdots\\ \underline{\boldsymbol{s}}_n\end{bmatrix}$$

The matrix equation (18) takes the form

$$(\underline{\mathbf{I}}\otimes\underline{\mathbf{A}}+\underline{\mathbf{A}}\otimes\underline{\mathbf{I}})\underline{\mathbf{s}}=\underline{\mathbf{q}}$$

8. *Hadamard Product.* If $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ are $n\times n$, their *Hadamard product* is defined as

$$\underline{\mathbf{H}}=\underline{\mathbf{A}}*\underline{\mathbf{B}},\qquad \underline{\mathbf{H}}=(h_{ij})=(a_{ij}b_{ij})\quad(i,\,j=1,\,\ldots,\,n)$$

9. *Tridiagonal Form.* If an $n\times n$ matrix $\underline{\mathbf{A}}$ is symmetric, it can be transformed via similarity transformation into a *tridiagonal* form having nonzero entries only on, directly below, or directly above the main diagonal.

10. *Binet–Cauchy Theorem.* A very useful theorem in electrical network theory states the algorithm for computing the determinant of the product $\underline{\mathbf{AB}}$ where $\underline{\mathbf{A}}$ is $m\times n$ and $\underline{\mathbf{B}}$ is $n\times m$, $m<n$. Define the *major* of $\underline{\mathbf{A}}$ (or of $\underline{\mathbf{B}}$) as the determinant of the submatrix of maximum order (in this case m). Then according to *Binet–Cauchy theorem,*

$$\det(\underline{\mathbf{AB}})=\sum_{\text{all majors}}(\text{products of corresponding majors of }\underline{\mathbf{A}}\text{ and }\underline{\mathbf{B}})$$

11. *Lancaster's Formula.* One has

$$p(\underline{\boldsymbol{x}})=e^{-f(\underline{\boldsymbol{x}})},\qquad f(\underline{\boldsymbol{x}})=\frac{1}{2}\underline{\boldsymbol{x}}^{\mathrm{T}}\underline{\mathbf{R}}^{-1}\underline{\boldsymbol{x}}>0$$

$$\int_{-\infty}^{\infty}p(\underline{\boldsymbol{x}})\,d\underline{\boldsymbol{x}}=(2\pi)^{-n/2}\Delta_{\underline{\mathbf{R}}}$$

12. *Singular-Value Decomposition.* Suppose $\underline{\mathbf{A}}$ is an $n\times m$ real matrix with $n>m$, with rank $r\le m$. Form
   - $\underline{\mathbf{S}}=\underline{\mathbf{A}}\underline{\mathbf{A}}^{\mathrm{T}}$, an $n\times n$ matrix with orthogonal eigenvector $\underline{\boldsymbol{e}}_1,\ldots,\underline{\boldsymbol{e}}_n$
   - $\underline{\mathbf{R}}=\underline{\mathbf{A}}^{\mathrm{T}}\underline{\mathbf{A}}$, an $m\times m$ matrix with orthogonal eigenvector $\underline{\boldsymbol{f}}_1,\ldots,\underline{\boldsymbol{f}}_m$
   - $\underline{\mathbf{U}}=[\underline{\boldsymbol{e}}_1|\cdots|\underline{\boldsymbol{e}}_n]$, $\underline{\mathbf{V}}=[\underline{\boldsymbol{f}}_1|\cdots|\underline{\boldsymbol{f}}_m]$
   - $\underline{\mathbf{\Sigma}}=Dg[\sigma_1,\,\sigma_2,\,\ldots,\,\sigma_r,\,\mathbf{0},\,\mathbf{0}]$, $\sigma_1>\sigma_2>\ldots>\sigma_r$ nonnegative
     $=Dg[\text{square roots of eigenvalues of }\underline{\mathbf{A}}^{\mathrm{T}}\underline{\mathbf{A}}]$

   Then the *singular–value decomposition* of $\underline{\mathbf{A}}$ is written as $\underline{\mathbf{A}}=\underline{\mathbf{U\Sigma V}}$. The solution to the linear equation $\underline{\mathbf{A}}\underline{\boldsymbol{x}}=\underline{\boldsymbol{b}}$ is

$$\underline{\boldsymbol{x}}=\underline{\boldsymbol{x}}_a+\underline{\boldsymbol{x}}_b,\qquad \underline{\mathbf{x}}_a=\sum_{i=1}^{r}(\underline{\boldsymbol{e}}_i^{\mathrm{T}}\underline{\boldsymbol{b}})\sigma_i^{-1}\underline{\boldsymbol{f}}_i,$$

$$\underline{\boldsymbol{x}}_b=\sum_{i=r+1}^{m}c_i\underline{\boldsymbol{f}}_i\ (c_i\text{ arbitrary})$$

   $\underline{\boldsymbol{x}}_b$ represents the auxiliary part of $\underline{\boldsymbol{x}}$, which can be taken as zero.

13. *Schur–Cohen Criteria.* In order that the roots of a polynomial

$$p(\lambda)=a_0\lambda^n+a_1\lambda^{n-1}+\cdots+a_n$$

lie within the unit circle, it is necessary and sufficient that the following conditions be satisfied:

$$(-1)^np(-1)>0$$
$$p(1)>0$$
$$\det(\underline{\mathbf{X}}_i+\underline{\mathbf{Y}}_i)>0$$
$$\det(\underline{\mathbf{X}}_i-\underline{\mathbf{Y}}_i)>0$$

$$\underline{\mathbf{X}}_i=\begin{bmatrix}a_0 & a_1 & \ldots & a_{i-1}\\ & a_0 & \ldots & a_{i-2}\\ & & \ddots & \vdots\\ 0 & & & a_0\end{bmatrix}$$

$$\underline{\mathbf{Y}}_i=\begin{bmatrix}& & & a_n\\ & & a_n & a_{n-1}\\ & \iddots & & \vdots\\ a_n & & \ldots & a_{n-i+1}\end{bmatrix}$$

14. *Hankel, Toeplitz, and Circulant Matrices.* A matrix $\underline{\mathbf{H}}$ is *Hankel* if its (i,j)th entry depends only on the value of $i+j$. Similiarly, a matrix $\underline{\mathbf{T}}$ is *Toeplitz* if its (i,j)th entry depends only on the value of $i-j$. A *circulant matrix* $\underline{\mathbf{C}}$ is defined by $(\underline{\mathbf{C}})_{ij}=c_{j+1-i}$, where subscripts are mod n. Thus 4×4 matrices of those kinds are of the form

$$\underline{\mathbf{H}}=\begin{bmatrix}h_1 & h_2 & h_3 & h_4\\ h_2 & h_3 & h_4 & h_5\\ h_3 & h_4 & h_5 & h_6\\ h_4 & h_5 & h_6 & h_7\end{bmatrix}$$

$$\underline{\mathbf{T}}=\begin{bmatrix}t_0 & t_{-1} & t_{-2} & t_{-3}\\ t_1 & t_0 & t_{-1} & t_{-2}\\ t_2 & t_1 & t_0 & t_{-1}\\ t_3 & t_2 & t_1 & t_0\end{bmatrix}$$

$$\underline{\mathbf{C}}=\begin{bmatrix}c_1 & c_2 & c_3 & c_4\\ c_4 & c_1 & c_2 & c_3\\ c_3 & c_4 & c_1 & c_2\\ c_2 & c_3 & c_4 & c_1\end{bmatrix}$$

Such matrices play an important role in system theory involving state-space realizations.

Nehari's Theorem. Hankel matrices can be used to compute an important bound on a function $f(t)$. Given $f(t)$, compute its

Fourier coefficients c_i ($i = -n, \ldots, -1, 0, 1, \ldots n$) and the associated complex symmetric $(n + 1) \times (n + 1)$ Hankel matrix $\underline{\mathbf{H}}$ such that $(\underline{\mathbf{H}})_{ij} = (c_{i+j})$ ($i = -n, -n + 1, \ldots, 0; j = 0, 1, \ldots, n$). A very useful result due to Nehari states that

$$|\underline{\boldsymbol{x}}^* \underline{\mathbf{H}} \underline{\boldsymbol{x}}| \leq k \underline{\boldsymbol{x}}^* \underline{\boldsymbol{x}}$$

where k is the bound of the function $f(t)$.

BIBLIOGRAPHY

A. C. Aitken, *Determinants and Matrices,* New York: Interscience, 1942. This is excellent reading.

R. Bellman, *Introduction to Matrix Algebra,* New York: McGraw-Hill, 1960. Very comprehensive coverage of current applications, very readable, lots of references.

G. D. Dudley, *Mathematical Foundations for Electromagnetic Theory,* Piscataway, NJ: IEEE Press, 1994, has a very good chapter on linear vector spaces.

F. R. Gantmacher, *The Theory of Matrices,* Vols. 1, 2, translated from Russian by K. A. Hirsch, New York: Chelsea, 1959. This is a very complete work.

R. A. Horn and C. R. Johnson, *Matrix Analysis,* Cambridge, UK: Cambridge Univ. Press, 1993. Must reading for Hermitian and symmetric matries.

T. Kailath, *Linear Systems,* Englewood Cliffs, NJ: Prentice-Hall, 1980. This is a very complete collection of matrix theory for the study of linear dynamical systems.

M. Marcus and H. Minc, *A Survey of Matrix Theory and Matrix Inequalities,* New York: Dover, 1964. A very good survey, very precise and concise.

K. S. Miller, *Some Eclectic Matrix Theory,* Krieger Publishing Co., has a very good treatment of matrix inverses.

A. D. Myskis, *Advanced Mathematics for Engineers: Special Course,* translated from Russian by V. M. Volosov and I. G. Volosova, Moscow: Mir, 1975.

B. Noble, *Applied Linear Algebra,* Englewood Cliffs, NJ: Prentice-Hall, 1969.

L. S. Pontryagin, *Ordinary Differential Equations,* translated from Russian by L. Kacinskas and W. B. Counts, Reading, MA: Addison-Wesley, 1962. Excellent chapter on linear algebra related to matrix differential equations.

V. I. Smirnov, *Linear Algebra and Group Theory,* translated from Russian and revised by R. A. Silverman, New York: McGraw-Hill, 1961. Excellent reading, particularly for the Jordan canonical form.

H. S. Wilf, *Mathematics for the Physical Sciences,* New York: Wiley, 1962.

J. H. Wilkinson, *The Algebraic Eigenvalue Problem,* London: Oxford Univ. Press, 1965. A very valuable book for computation methods.

N. PURI
Rutgers University

LINEAR ALGEBRAIC SOLVERS. See PARALLEL NUMERICAL ALGORITHMS AND SOFTWARE.

/ /

Lyapunov Methods
Standard Article

Anthony N. Michel[1]
[1]University of Notre Dame, Notre Dame, IN

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2434
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (258K)

Abstract

The sections in this article are

Dynamical Systems Determined By Ordinary Differential Equations

Lyapunov and Lagrange Stability Concepts

Lyapunov Functions

Lyapunov Stability Results—Motivation

The Principal Lyapunov and Lagrange Stability Theorems

Some Extensions and Further Results

Some Notes and References

| | |

Copyright © 1999-2008 All Rights Reserved.

LYAPUNOV METHODS

Phenomena that change are most suitably described in terms of their states evolving in time. These are called dynamical systems. Such systems, which occur in nature or are man-made, are frequently endowed with natural states, called equilibria (or rest positions) or operating points. When after a disturbance, the system states return to an equilibrium (operating point), one speaks of a stable equilibrium. In the case of man-made objects, a great deal of engineering is concerned with the design of systems with stable operating points (equilibria). The most universally accepted notion of stability is Lyapunov stability. In the present article we introduce this concept and we present results for analyzing the Lyapunov stability properties of an equilibrium. Collectively, such results are called Lyapunov Method.

For a given set of initial conditions, the characterization of a physical system is usually given in terms of the evolution in time of a motion in state space. If we let t_0 and $x(t_0) = x_0$ denote initial time and initial state, respectively, we can represent the motion by a function $p(\cdot, x_0, t_0)$ from R^+ to X, that is, $p(\cdot, x_0, t_0): [t_0, \infty) \to X$ where $[t_0, \infty) \subset R^+ = [0, \infty)$ denotes real time and X denotes the state space (some metric space). Now if we let initial state x_0 vary over a specified set A in X ($x_0 \in A \subset X$) and if we let initial time t_0 vary over a specified set R_0^+ in R^+ ($t_0 \in R_0^+ \subset R^+$), there will result a family of motions that we will denote by S, provided that $p(t_0, x_0, t_0) = x_0$. The resulting five-tuple $\{X, S, A, R^+, R_0^+\}$ is called a dynamical system. When there is no room for confusion, we will simply speak of a dynamical system S, rather than a dynamical system $\{X, S, A, R^+, R_0^+\}$.

In the above discussion, the evolution of motions is along real time t in R^+ (i.e., $t \in R^+$). In this case, we speak of continuous-time dynamical systems. In many applications, motions may also take place along discrete instants, for example, along the nonnegative integers, $Z^+ = \{0, 1, 2, 3, \cdots\}$, resulting in a discrete-time dynamical system that we denote by $\{X, S, A, Z^+, Z_0^+\}$, where Z_0^+ is in Z^+ (i.e., $Z_0^+ \subset Z^+$}. Still, in other types of applications, some components of the motions may evolve along R^+, while others may evolve, e.g., along Z^+, so that the entire motion will evolve along a subset T of $R^+ \times Z^+$ (i.e., $T \subset R^+ \times Z^+$). The resulting dynamical systems $\{X, S, A, T, T_0\}$ are called hybrid dynamical systems.

Examples of continuous-time dynamical systems are systems whose motions are determined by the solutions of systems of ordinary differential equations and systems of ordinary differential inequalities while examples of discrete-time dynamical systems include systems whose motions are determined by the solutions of ordinary difference equations and systems of ordinary difference inequalities. All of these are examples of finite dimensional dynamical systems.

If a dynamical system is not finite dimensional, it is said to be infinite dimensional. Examples of infinite dimensional dynamical systems include those whose motions are determined by the solutions of delay differential equations, functional differential equations, partial differential equations, Volterra integrodifferential equations, and the like.

In addition to the above, dynamical systems may also be determined by "equation free" characterizations (discrete event systems, systems determined by Petri nets, and the like), and by mixtures of equations [hybrid dynamical systems, such as, digital control systems consisting of a continuous-time plant and a digital (discrete-time) controller].

Dynamical systems that represent processes that are either manufactured or can be found in nature are usually endowed with one or more "operating points." Mathematically, these are represented by *invariant sets.* A set M in A (i.e., $M \subset A \subset X$) is said to be an invariant set (with respect to S) if whenever a motion at t_0 starts out in M will remain in M forever (i.e., if $p(t_0, x_0, t_0) = x_0 \in M$, then $p(t, x_0, t_0) \in M$ for all $t \geq t_0 \geq 0$). If in particular, M consists of one single point, say x_e, then x_e is called an equilibrium of the dynamical system S. In this case $M = \{x_e\}$ and $p(t, x_e, t_0) = x_e$ for all $t \geq t_0$.

In the following, we will confine ourselves to equilibria. A discussion for general invariant sets would follow along similar lines, involving obvious modifications.

The qualitative behavior of motions of a dynamical system in the vicinity of an operating point (i.e., in the vicinity of an equilibrium) is of great interest in applications and gives rise to the various stability notions of an equilibrium in the sense of Lyapunov.

Suppose that x_e is an equilibrium of a dynamical system S. If by choosing all the initial points of the motions in a sufficiently small neighborhood of x_e, we can force the motions to stay sufficiently close to x_e for all $t \geq t_0 \geq 0$ (in terms of the metric of X), the equilibrium x_e is said to be stable (in the sense of Lyapunov). If x_e is stable, and if by choosing all initial points of the motions in some neighborhood of x_e at $t = t_0$, we can force the motions to tend to x_e as t becomes arbitrarily large (i.e., as $t \to \infty$), then x_e is said to be asymptotically stable (in the sense of Lyapunov). The set of initial points for which the above statement is true is called the domain of attraction of x_e. If the above statement is true for all initial points (i.e., for all motions), then x_e is said to be globally asymptotically stable. In this case, x_e is the only equilibrium of the dynamical

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

system. If x_e is asymptotically stable and if "the motions tend to x_e exponentially" (with respect to the metric of X), then x_e is said to be exponentially stable. Finally, if x_e is not stable, it is said to be unstable.

Other closely related qualitative attributes of dynamical systems concern various notions of boundedness of motions. These comprise the Lagrange stability of dynamical systems.

In the qualitative analysis of dynamical systems, Lyapunov methods play a central role. The aim is to ascertain qualitative properties of families of motions near an equilibrium point (in the sense discussed above) without having to actually determine explicit expressions for the motions of a dynamical system. This is fortunate, for in general, there are no known techniques that yield explicit expressions for such motions. It is for this reason that one frequently speaks of the Direct Method of Lyapunov (of stability analysis). In addition to determining various stability properties of an equilibrium, the Direct Method of Lyapunov can also be used in determining various boundedness properties of motions of dynamical systems (Lagrange stability).

The Direct Method of Lyapunov employs auxiliary scalar-valued functions of the system state, called Lyapunov functions, which frequently are viewed as (generalized) energy functions for dynamical systems, or as (generalized) distance functions [from the motions (at time t) to an equilibrium] for dynamical systems. Stability properties of an equilibrium (or boundedness properties of motions) are then deduced from the behavior of the Lyapunov functions evaluated along the motions of a dynamical system. In general, this can be accomplished without explicitly determining expressions for the motions of a given dynamical system; hence, the term the Direct Method of Lyapunov.

To make the above discussion more precise, we will in the following confine ourselves to finite-dimensional, continuous-time dynamical systems whose motions are determined by systems of ordinary differential equations. In this case, the state space is given by $X = R^n$, the metric on X is determined by any one of the equivalent norms, $|\cdot|$, on R^n, and we will assume that $R_0^+ = R^+$.

DYNAMICAL SYSTEMS DETERMINED BY ORDINARY DIFFERENTIAL EQUATIONS

We shall concern ourselves with dynamical systems that are determined by the solutions of first-order ordinary differential equations of the form

$$\dot{x} = f(x, t) \tag{E}$$

where $x = (x_1, \cdots, x_n)^T \in R^n$ (i.e., x is a real n-vector), $t \in R^+ = [0, \infty)$ (i.e., $t \geq 0$), $\dot{x}$ denotes differentiation with respect to t (i.e., $\dot{x} = (\dot{x}_1, \cdots, \dot{x}_n)^T$, $\dot{x}_i = dx_i/dt$, $i = 1, \cdots, n$), and f is a continuous function of $R^n \times R^+$ into R^n (i.e., $f(x, t) = [f_1(x_1, \cdots, x_n, t), \cdots, f_n(x_1, \cdots x_n, t)]^T = [f_1(x, t), \cdots, f_n(x, t)]^T$ where it is assumed that $f_i(x, t)$ is continuous on $R^n \times R^+$, $i = 1, \cdots, n$). Unless otherwise stated, we assume that for every (x_0, t_0), the initial-value problem

$$\dot{x} = f(x, t),\ x(t_0) = x_0 \tag{I}$$

possesses a unique solution $p(t, x_0, t_0)$ with $p(t_0, x_0, t_0) = x_0$, which is defined for all $t \geq t_0$ and which depends continuously on the initial conditions (x_0, t_0).

A point x_e in R^n is called an equilibrium point of (E) if $f(x_e, t) = 0$ for all $t \geq 0$. Other terms for equilibrium point include stationary point, singular point, critical point, and rest position. We note that if x_e is an equilibrium point of (E), then for any $t_0 \geq 0$, $p(t, x_e, t_0) = x_e$ for all $t \geq t_0$ [i.e., x_e is a unique solution of (E) with initial conditions given by $p(t_0, x_e, t_0) = x_e$].

As a specific example, consider the simple pendulum that is described by equations of the form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -k \sin x_1,\ k > 0 \end{aligned} \tag{1}$$

where x_1 denotes angular displacement and x_2 denotes angular velocity of a mass subjected to gravitational force and rotating about a fixed point. By convention, we let $x_1 = 0$ when the mass position is in the most downward position. Physically, the pendulum has two equilibrium points. One of these is when $x_1 = x_2 = 0$ (when the mass is in the most downward position) and the second point is when $x_1 = \pm\pi$ and $x_2 = 0$. However, the model of the pendulum, represented by system (1), has countably infinitely many equilibrium points which are located at the points $(\pi n, 0)$, $n = 0, \pm 1, \pm 2, \cdots$.

An equilibrium point x_e of (E) is called an isolated equilibrium point if there is an $r > 0$ such that $B(x_e, r) \subset R^n$ contains no equilibrium points of (E) other than x_e itself. Here, $B(x_e, r) = \{x \in R^n : |x - x_e| < r\}$, where $|\cdot|$ denotes any one of the equivalent norms on R^n. (Thus, $B(x_e, r)$ denotes a sphere in R^n with center at x_e and radius equal to $r > 0$.)

All equilibrium points of system (1) are isolated equilibria in R^2. On the other hand, for a dynamical system described by the system of equations

$$\begin{aligned} \dot{x}_1 &= -ax_1 + bx_1x_2 \\ \dot{x}_2 &= -bx_1x_2 \end{aligned} \tag{2}$$

where $a > 0$, $b > 0$ are constants, every point on the positive x_2-axis is an equilibrium point for system (2).

It should be noted that there are systems with no equilibrium points at all, as is the case, for example, in the system of equations

$$\begin{aligned} \dot{x}_1 &= c + \sin(x_1 + x_2) + x_1 \\ \dot{x}_2 &= c + \sin(x_1 + x_2) - x_1 \end{aligned} \tag{3}$$

where $c \geq 2$ is a constant.

There are many important classes of systems that possess only one equilibrium. For example, consider the linear homogeneous system of equations given by

$$\dot{x} = A(t)x, \tag{LH}$$

where $A(t) = [a_{ij}(t)]$ denotes a real $n \times n$ matrix whose elements $a_{ij}(t)$ are continuous functions from R^+ into R (i.e., $a_{ij} : R^+ \to R$). The system (LH) has a unique equilibrium at the origin ($x_e = (x_1, x_2)^T = (0, 0)^T = 0$) if $A(t_0)$ is nonsingular for all $t_0 \geq 0$. Also, the autonomous system of equations

$$\dot{x} = f(x) \tag{A}$$

where $f: R^n \to R^n$ is assumed to be continuously differentiable with respect to all of its arguments, and where

$$J(x_e) = \frac{\partial f}{\partial x}(x)\,|_{x=x_e} \tag{4}$$

denotes the $n \times n$ Jacobian matrix defined by $\partial f/\partial x = [\partial f_i/\partial x_j]$ has an isolated equilibrium at x_e if $f(x_e) = 0$ and $J(x_e)$ is nonsingular.

Unless otherwise stated, we shall assume henceforth that a given equilibrium point is an isolated equilibrium. Also, we shall assume, unless otherwise stated, that in a given discussion, the equilibrium of interest is located at the origin of R^n. This assumption can be made without any loss of generality. To see this, assume that $x_e \neq 0$ is an equilibrium of system (E) [i.e., $f(x_e, t) = 0$ for all $t \geq 0$]. Let $w = x - x_e$. Then $w = 0$ is an equilibrium of the transformed system

$$\dot{w} = F(w, t) \tag{5}$$

where

$$F(w, t) = f(w + x_e, t) \tag{6}$$

Since system (6) establishes a one-to-one correspondence between the solutions of system (E) and system (5), we may assume henceforth that system (E) possesses the equilibrium of interest located at the origin. The equilibrium $x_e = 0$ will sometimes be referred to as the trivial solution of system (E).

LYAPUNOV AND LAGRANGE STABILITY CONCEPTS

We now state and interpret several definitions of stability of an equilibrium point, in the sense of Lyaponov.

The equilibrium $x_e = 0$ of system (E) is stable if for every $\epsilon > 0$ and $t_0 \geq 0$, there exists a $\delta(\epsilon, t_0) > 0$ such that

$$|p(t, x_0, t_0)| < \epsilon \text{ for all } t \geq t_0 \tag{7}$$

whenever

$$|x_0| < \delta(\epsilon, t_0) \tag{8}$$

[In system (7) and system (8), $|\cdot|$ denotes any one of the equivalent norms on R^n.]

In Fig. 1 we depict the behavior of the solutions (motions) in the vicinity of a stable equilibrium for the case $x \in R^2$. The interpretation of this figure is that when $x_e = 0$ is stable, then by choosing the initial points in a sufficiently small spherical neighborhood, we can force the graph of the solution for $t \geq t_0$ to lie entirely inside a given cylinder.

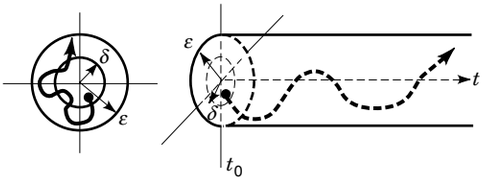


Figure 1. Qualitative behavior of a trajectory in the vicinity of a stable equilibrium.

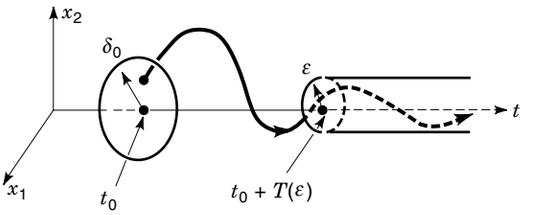


Figure 2. Qualitative behavior of a trajectory in the vicinity of an attractive equilibrium.

In the above definition of stability, δ depends on ϵ and t_0 [i.e., $\delta = \delta(\epsilon, t_0)$]. If δ is independent of t_0 [i.e., $\delta = \delta(\epsilon)$], then the equilibrium $x = 0$ of system (E) is said to be uniformly stable.

The equilibrium $x_e = 0$ of system (E) is said to be asymptotically stable if (1) it is stable, and (2) for every $t_0 \geq 0$ there exists an $\eta(t_0) > 0$ such that $\lim_{t\to\infty} p(t, x_0, t_0) = 0$ whenever $|x_0| < \eta$. Furthermore, the set of all $x_0 \in R^n$ such that $p(t, x_0, t_0) \to 0$ as $t \to \infty$ for some $t_0 \geq 0$ is called the domain of attraction of the equilibrium $x_e = 0$ of system (E). Also, if for system (E) condition (2) is true, then the equilibrium $x_e = 0$ is said to be attractive.

The equilibrium $x = 0$ of system (E) is said to be uniformly asymptotically stable if (1) it is uniformly stable, and (2) there is a $\delta_0 > 0$ such that for every $\epsilon > 0$ and any $t_0 \in R^+$, there exists a $T(\epsilon) > 0$, independent of t_0, such that $|p(t, x_0, t_0)| < \epsilon$ for all $t \geq t_0 + T(\epsilon)$ whenever $|x_0| < \delta_0$.

In Fig. 2 we depict pictorially property [system (2)] for uniform asymptotic stability. The interpretation of this figure is that by choosing the initial points in a sufficiently small spherical neighborhood at $t = t_0$, we can force the graph of the solution to lie inside a given cylinder for all $t > t_0 + T(\epsilon)$. Condition (2) can be rephrased by saying that there exists a $\delta_0 > 0$ such that $\lim_{t\to\infty} p(t + t_0, x_0, t_0) = 0$, uniformly in (x_0, t_0) for $t_0 \geq 0$ and for $|x_0| \leq \delta_0$.

In applications we are frequently interested in a special case of uniform asymptotic stability: the equilibrium $x_e = 0$ of system (E) is exponentially stable if there exists an $\alpha > 0$, and for $\epsilon > 0$, there exists a $\delta(\epsilon) > 0$, such that $|p(t, x_0, t_0)| \leq \epsilon e^{\alpha(t-t_0)}$ for all $t \geq t_0$ whenever $|x_0| < \delta(\epsilon)$ and $t_0 \geq 0$.

In Fig. 3, the behavior of a solution in the vicinity of an exponentially stable equilibrium $x_e = 0$ is shown.

The equilibrium $x_e = 0$ of system (E) is said to be unstable if it is not stable. It is important to note that if $x_e = 0$ is an

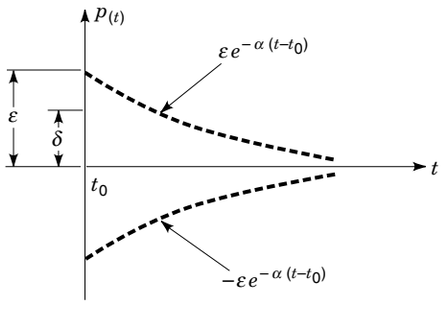


Figure 3. A trajectory envelope in the vicinity of an exponentially stable equilibrium.

unstable equilibrium, it still can happen that all the solutions tend to zero with increasing t. Thus, instability and attractivity of an equilibrium are compatible concepts. Note that the equilibrium $x_e = 0$ is necessarily unstable if every neighborhood of the origin contains initial points corresponding to unbounded solutions (i.e., solutions whose norm $|p(t, x_0, t_0)|$ grows to infinity on a sequence $t_m \to \infty$). However, it can happen that a system with unstable equilibrium $x_e = 0$ [see system (E)] may have only bounded solutions.

The above concepts pertain to local properties of an equilibrium. We now consider some global characterizations.

A solution $p(t, x_0, t_0)$ of system (E) is bounded if there exists a $\beta > 0$ such that $|p(t, x_0, t_0)| < \beta$ for all $t \geq t_0$, where β may depend on each solution. System (E) is said to possess Lagrange stability if for each $t_0 \geq 0$ and x_0 the solution $p(t, x_0, t_0)$ is bounded.

The solutions of system (E) are uniformly bounded if for any $\alpha > 0$ and $t_0 \in R^+$, there exists a $\beta = \beta(\alpha) > 0$ (independent of t_0) such that if $|x_0| < \alpha$, then $|p(t, x_0, t_0)| < \beta$ for all $t \geq t_0$.

The solutions of system (E) are uniformly ultimately bounded (with bound B) if there exists $B > 0$ and if corresponding to any $\alpha > 0$ and $t_0 \in R^+$, there exists a $T = T(\alpha)$ (independent of t_0) such that $|x_0| < \alpha$ implies that $|p(t, x_0, t_0)| < B$ for all $t \geq t_0 + T$.

In contrast to the boundedness properties given in the preceding three paragraphs, the concepts introduced earlier as well as those stated in the following are usually referred to as stability, respectively, instability, in the sense of Lyapunov.

The equilibrium $x_e = 0$ of system (E) is asymptotically stable in the large (or globally asymptotically stable) if it is stable and if every solution of system (E) tends to zero as $t \to \infty$. In this case, the domain of attraction of the equilibrium $x_e = 0$ of system (E) is all of R^n. Note that in this case, $x_e = 0$ is the only equilibrium of system (E).

The equilibrium $x_e = 0$ of system (E) is uniformly asymptotically stable in the large if (1) it is uniformly stable, and (2) for any $\alpha > 0$ and any $\epsilon > 0$ and $t_0 \in R^+$, there exists $T(\epsilon, \alpha) > 0$, independent of t_0, such that if $|x_0| < \alpha$, then $|p(t, x_0, t_0)| < \epsilon$ for all $t \geq t_0 + T(\epsilon, \alpha)$.

Finally, the equilibrium $x = 0$ of system (E) is exponentially stable in the large if there exists $\alpha > 0$ and for any $\beta > 0$, there exists $k(\beta) > 0$ such that $|p(t, x_0, t_0)| \leq k(\beta)|x_0|e^{\alpha(t-t_0)}$ for all $t \geq t_0$ whenever $|x_0| < \beta$. In the following, we cite several specific examples.

1. The scalar equation

$$\dot{x} = 0 \tag{9}$$

has for any initial condition $x(0) = x_0 = c$ the solution $p(t, c, 0) = c$. All solutions are equilibria for system (9). The trivial solution $x_e = 0$ is stable; in fact, it is uniformly stable.

2. The scalar equation

$$\dot{x} = ax,\ a > 0 \tag{10}$$

has for every initial condition $x(0) = x_0 = c$ the solution $p(t, c, 0) = ce^{at}$, and $x_e = 0$ is the only equilibrium of system (10). This equilibrium is unstable.

3. The scalar equation

$$\dot{x} = -ax,\ a > 0 \tag{11}$$

has for every initial condition $x(0) = x_0 = c$ the solution $p(t, c, 0) = ce^{-at}$, and $x_e = 0$ is the only equilibrium of system (11). This equilibrium is exponentially stable in the large.

4. The scalar equation

$$\dot{x} = \frac{-1}{t+1}x \tag{12}$$

has for every initial condition $x(t_0) = x_0 = c$, $t_0 \geq 0$, a unique solution of the form $p(t, c, t_0) = [(1 + t_0)c]/(t + 1)$, and $x_e = 0$ is the only equilibrium of system (12). This equilibrium is uniformly stable and asymptotically stable in the large, but it is not uniformly asymptotically stable.

5. By making use of the general properties of the solutions of linear autonomous homogeneous systems of equations given by

$$\dot{x} = Ax,\ t \geq 0 \tag{L}$$

where $A = [a_{ij}]$ is a real $n \times n$ matrix, the following has been established:

(i) The equilibrium $x_e = 0$ of system (L) is stable if all eigenvalues of A have nonpositive real parts and every eigenvalue of A that has a zero real part is a simple zero of the characteristic polynomial of A.

(ii) The equilibrium $x = 0$ of system (L) is asymptotically stable if and only if all eigenvalues of A have negative real parts. In this case, there exist constants $k > 0$, $\sigma > 0$ such that $|p(t, x_0, t_0)| \leq k|x_0|e^{-\sigma(t-t_0)}$ for all $t \geq t_0 \geq 0$.

LYAPUNOV FUNCTIONS

The general Lyapunov and Lagrange stability results for dynamical systems described by system (E) involve the existence of real-valued functions $v : D \to R$. In the case of local results (e.g., stability, instability, asymptotic stability, and exponential stability of an equilibrium $x_e = 0$), we shall usually only require that $D = B(h) \subset R^n$ for some $h > 0$, or $D = B(h) \times R^+$. (Recall that $B(h) = \{x \in R^n : |x| < h\}$ where $|x|$ denotes any one of the equivalent norms of x on R^n and $R^+ = [0, \infty)$.) On the other hand, in the case of global results [e.g., asymptotic stability in the large and exponential stability in the large of the equilibrium $x_e = 0$, and uniform boundedness of solutions of system (E)], we have to assume that $D = R^n$ or $D = R^n \times R^+$. Unless stated otherwise, we shall always assume that $v(0, t) = 0$ for all $t \in R^+$ [respectively, $v(0) = 0$].

Now let $p(t)$ be an arbitrary solution of system (E) and consider the function $t \mapsto v(p(t), t)$. If v is continuously differentiable with respect to all of its arguments, then we obtain, by the chain rule, the derivative of v with respect to t along the solutions of system (E), $\dot{v}_{(E)}$, as

$$\dot{v}_{(E)}(p(t), t) = \frac{\partial v}{\partial t}(p(t), t) + \nabla v(p(t), t)^T f(p(t), t) \tag{13}$$

where ∇v denotes the gradient vector of v with respect to x. Note that for a solution $p(t, x_0, t_0)$ of system (E) we have

$$v(p(t), t) = v(x_0, t_0) + \int_{t_0}^{t} \dot{v}_{(E)}(p(\tau, x_0, t_0), \tau)\, d\tau \tag{14}$$

The above observations motivate the following: $\dot{v}_{(E)}: R^n \times R^+ \to R$ (respectively, $\dot{v}_{(E)}: B(h) \times R^+ \to R$), defined by

$$\begin{aligned}\dot{v}_{(E)}(x,t) &= \frac{\partial v}{\partial t}(x,t) + \sum_{i=1}^{n} \frac{\partial v}{\partial x_i}(x,t) f_i(x,t) \\ &= \frac{\partial v}{\partial t}(x,t) + \nabla v(x,t)^T f(x,t)\end{aligned} \tag{15}$$

is called the derivative of v, with respect to t, along the solutions of system (E).

It is important to note that in system (15), the derivative of v with respect to t, along the solutions of system (E), is evaluated without having to solve system (E). The significance of this will become clear later. We also note that when $v: R^n \to R$ (resp., $v: B(h) \to R$), then system (15) reduces to $\dot{v}_{(E)}(x, t) = \nabla v(x)^T f(x, t)$. Also, in the case of autonomous systems (A), if $v: R^n \to R$ (resp., $v: B(h) \to R$), we have

$$\dot{v}_{(A)}(x) = \nabla v(x)^T f(x) \tag{16}$$

Occasionally, we shall require only that v be continuous on its domain of definition and that it satisfy locally a Lipschitz condition with respect to x. In such cases we define the upper right-hand derivative of v with respect to t along the solutions of system (E) by

$$\dot{v}_{(E)}(x,t) = \limsup_{\theta \to 0+} (1/\theta)\{v[x + \theta \cdot f(x,t), t + \theta] - v(x,t)\} \tag{17}$$

When v is continuously differentiable, then system (17) reduces to system (15).

In characterizing v-functions of the type discussed above, we will employ Kamke comparison functions, which are defined as follows: a continuous function $\psi: [0, r_1] \to R^+$ (resp., $\psi: R^+ \to R^+$) is said to belong to class K (i.e., $\psi \in K$), if $\psi(0) = 0$ and if ψ is strictly increasing on $[0, r_1]$ (resp., on $[0, \infty)$). If $\psi: R^+ \to R^+$, if $\psi \in K$, and if $\lim_{r \to \infty} \psi(r) = \infty$, then ψ is said to belong to class KR.

We are now in a position to characterize v-functions in a variety of ways. In the following, we assume that $v: R^n \times R^+ \to R$ (resp., $v: B(h) \times R^+ \to R$), that $v(0, t) = 0$ for all $t \in R^+$, and that v is continuous.

a. v is said to be positive definite if for some $r > 0$, there exists a $\psi \in K$ such that $v(x, t) \geq \psi(|x|)$ for all $t \geq 0$ and for all $x \in B(r)$.
b. v is decrescent if there exists a $\psi \in K$ such that $|v(x, t)| \leq \psi(|x|)$ for all $t \geq 0$ and for all $x \in B(r)$ for some $r > 0$.
c. $v: R^n \times R^+ \to R$ is radially unbounded if there exists a $\psi \in KR$ such that $v(x, t) \geq \psi(|x|)$ for all $x \in R^n$ and for all $t \geq 0$.
d. v is negative definite if $-v$ is positive definite.
e. v is positive semidefinite if $v(x, t) \geq 0$ for all $x \in B(r)$ for some $r > 0$ and for all $t \geq 0$.
f. v is negative semidefinite if $-v$ is positive semidefinite.

The definitions corresponding to the above concepts when $v: R^n \to R$ or $v: B(h) \to R$ [where $B(h) \subset R^n$ for some $h > 0$] involve obvious modifications. We now consider some specific examples.

a. $v: R^3 \to R$ given by $v(x) = x^T x = x_1^2 + x_2^2 + x_3^2$ is positive definite and radially unbounded.
b. $v: R^3 \to R$ given by $v(x) = x_1^2 + (x_2 + x_3)^2$ is positive semidefinite, but not positive definite.
c. $v: R^2 \to R$ given by $v(x) = x_1^2 + x_2^2 - (x_1^2 + x_2^2)^3$ is positive definite but not radially unbounded.
d. $v: R^3 \to R$ given by $v(x) = x_1^2 + x_2^2$ is positive semidefinite but not positive definite.
e. $v: R^2 \to R$ given by $v(x) = x_1^4/(1 + x_1^4) + x_2^4$ is positive definite but not radially unbounded.
f. $v: R^2 \times R^+ \to R$ given by $v(x, t) = (1 + \cos^2 t)x_1^2 + 2x_2^2$ is positive definite, decrescent, and radially unbounded.
g. $v: R^2 \times R^+ \to R$ given by $v(x, t) = (x_1^2 + x_2^2)\cos^2 t$ is positive semidefinite and decrescent.
h. $v: R^2 \times R^+ \to R$ given by $v(x, t) = (1 + t)(x_1^2 + x_2^2)$ is positive definite and radially unbounded but not decrescent.
i. $v: R^2 \times R^+$ given by $v(x, t) = x_1^2/(1 + t) + x_2^2$ is decrescent and positive semidefinite but not positive definite.
j. $v: R^2 \times R^+ \to R$ given by $v(x, t) = (x_2 - x_1)^2(1 + t)$ is positive semidefinite but not positive definite or decrescent.

Of special interest are quadratic forms $v: R^n \to R$ given by

$$v(x) = x^T Bx = \sum_{i,k=1}^{n} b_{ik} x_i x_k \tag{18}$$

where $B = [b_{ij}]$ is a real, symmetric $n \times n$ matrix. Since B is symmetric, it is diagonizable and all its eigenvalues are real. Let λ_m and λ_M denote the smallest and largest eigenvalues of B and let $|x|$ denote the Euclidean norm of x. It has been shown that

$$\lambda_m |x|^2 \leq v(x) \leq \lambda_M |x|^2 \tag{19}$$

for all $x \in R^n$. From system (19) these facts follow now immediately:

a. v is definite (i.e., either positive definite or negative definite) if and only if all eigenvalues are nonzero and have the same sign.
b. v is semidefinite (i.e., either positive semidefinite or negative semidefinite) if and only if the nonzero eigenvalues of B have the same sign.
c. v is indefinite (i.e., in every neighborhood of the origin $x = 0$, v assumes positive and negative values) if and only if B possesses both positive and negative eigenvalues.

It has also been shown that v given by system (18) is positive definite (and radially unbounded) if and only if all principal minors of the matrix B are positive, that is, if and only if

$$\det \begin{bmatrix} b_{11} & \cdots & b_{1k} \\ \vdots & & \vdots \\ b_{k1} & \cdots & b_{kk} \end{bmatrix} > 0, \quad k = 1, \ldots, n$$

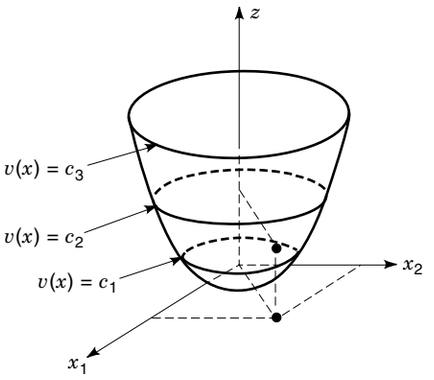


Figure 4. Surface described by a quadratic form.

Furthermore, v given by system (18) is negative definite if and only if

$$(-1)^k \det \begin{bmatrix} b_{11} & \cdots & b_{1k} \\ \vdots & & \vdots \\ b_{k1} & \cdots & b_{kk} \end{bmatrix} > 0, \quad k = 1, \ldots, n$$

It turns out that quadratic forms [system (18)] have some interesting geometric properties, as is shown next. Let $n = 2$ and assume that both eigenvalues of B are positive, which means that v is positive definite and radially unbounded. In R^3, the surface determined by the equation

$$z = v(x) = x^T Bx \tag{20}$$

describes a cup-shaped surface as shown in Fig. 4. Note in this figure that corresponding to every point on this cup-shaped surface there exists one and only one point in the x_1x_2 plane. Note also that the loci defined by $C_i = \{x \in R^2 : v(x) = c_i \geq 0\}$, c_i = constant, determine closed curves in the x_1x_2 plane as shown in Fig. 5. These are called level curves. Note that $C_0 = \{0\}$ corresponds to the case $z = c_0 = 0$. Further, note also that this function v can be used to cover the entire R^2 plane with closed curves by selecting for z all values in R^+.

In the more general case, when $x \in R^n$, $n > 2$, and B is positive definite, the preceding discussion concerning quadratic forms [system (18)] still holds; however, in this case, the closed curves C_i must be replaced by closed hypersurfaces in R^n and simple visualizations as shown in Figs. 4 and 5 are no longer possible.

LYAPUNOV STABILITY RESULTS—MOTIVATION

Before presenting the Lyapunov and Lagrange stability results, we will give geometric interpretations for some of these. To this end we consider dynamical systems determined by two first-order ordinary differential equations of the form

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) \\ \dot{x}_2 &= f_2(x_1, x_2) \end{aligned} \tag{21}$$

and we assume that for every (x_0, t_0), $t_0 \geq 0$, system (21) has a unique solution $p(t, x_0, t_0)$ with $p(t_0, x_0, t_0) = x_0$. We also assume that $(x_1, x_2)^T = (0, 0)^T$ is the only equilibrium in $B(h)$ for some h > 0.

Next, let v be a positive definite, continuously differentiable function with nonvanishing gradient ∇v on $0 < |x| \leq h$. Then $v(x) = c$, $c \geq 0$, defines for sufficiently small constants $c > 0$ a family of closed curves C_i, which cover the neighborhood $B(h)$ as shown in Fig. 6. Note that the origin $x = 0$ is located in the interior of each curve and $C_0 = \{0\}$.

Now suppose that all solutions (motions) of system (21) originating from points on the circular disk $|x| \leq r_1 < h$ cross the curves $v(x) = c$ from the exterior toward the interior when we proceed along these solutions in the direction of increasing values of t. Then we can conclude that these solutions approach the origin as t increases (i.e., the equilibrium $x = 0$ in this case is asymptotically stable).

In terms of the given v function, we have the following interpretation. For a given solution $p(t, x_0, t_0)$ to cross the curve $v(x) = r$, $r = v(x_0)$, the angle between the outward normal vector $\nabla v(x_0)$ and the derivative of $p(t, x_0, t_0)$ at $t = t_0$ must be greater than $\pi/2$, that is,

$$\dot{v}_{(21)}(x_0) = \nabla v(x_0)^T f(x_0) < 0$$

For this to happen at all points, we must have $\dot{v}_{(21)}(x) < 0$ for $0 < |x| \leq r_1$. The same results can be arrived at from an analytic point of view. The function $V(t) \triangleq v[p(t, x_0, t_0)]$ decreases monotonically as t increases. This implies that the derivative $\dot{v}[p(t, x_0, t_0)]$ along the solution $p(t, x_0, t_0)$ must be negative definite in $B(r)$ for $r > 0$ sufficiently small.

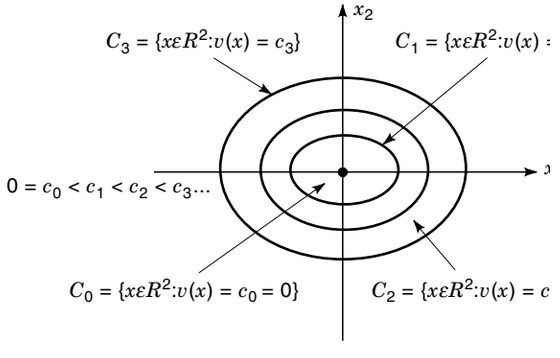


Figure 5. Level curves determined by a quadratic form.

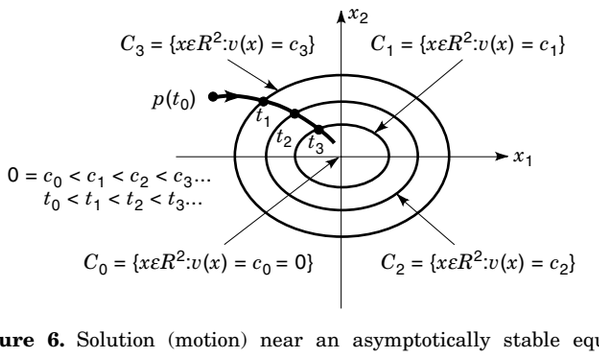


Figure 6. Solution (motion) near an asymptotically stable equilibrium.

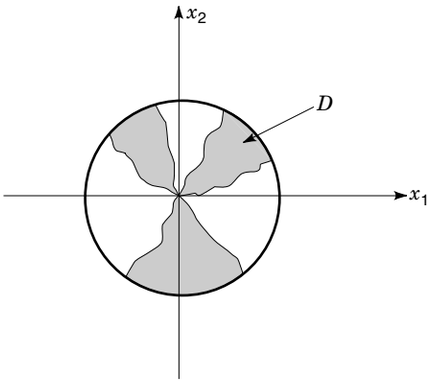


Figure 7. Instability of an equilibrium.

Proceeding, let us next assume that system (21) has only one equilibrium, $x_e = 0$, and that v is positive definite and radially unbounded. In this case, the relation $v(x) = c$, $c \in R^+$, can be used to cover all of R^2 by closed curves of the type shown in Fig. 6. If for arbitrary initial conditions (x_0, t_0), the solution of system (21), $p(t, x_0, t_0)$, behaves as already discussed, then it follows that the derivative of v along this solution, $\dot{v}(p[t, x_0, t_0,)]$, will be negative definite in R^2.

The foregoing discussion was given in terms of an arbitrary solution of system (21). This suggests the following results:

1. If there exists a positive definite function v such that $\dot{v}_{(21)}$ is negative definite, then the equilibrium $x_e = 0$ of system (21) is asymptotically stable.
2. If there exists a positive definite and radially unbounded function v such that $\dot{v}_{(21)}$ is negative definite for all $x \in R^2$, then the equilibrium $x_e = 0$ of system (21) is asymptotically stable in the large.

Continuing by making reference to Fig. 7, let us now assume that we can find for system (21) a continuously differentiable function $v : R^2 \to R$ that is indefinite and that has the properties discussed in the following. Since v is indefinite, there exist in each neighborhood of the origin points for which $v > 0$, $v < 0$, and $v(0) = 0$. Confining our attention to $B(k)$, where $k > 0$ is sufficiently small, we let $D = \{x \in B(k) : v(x) < 0\}$. Note that D may consist of several subdomains. The boundary of D, ∂D, as shown in Fig. 7, consists of points in $\partial B(k)$ and of points determined by $v(x) = 0$. Let us assume that in the interior of D, v is bounded. Suppose that $\dot{v}_{(21)}$ is negative definite in D and that $p(t)$ is a solution of system (21) that originates somewhere on the boundary of D [i.e., $p(t_0, x_0, t_0) = x_0 \in \partial D$] with $v(x_0) = 0$. Then this solution will penetrate the boundary of D at points where $v = 0$ as t increases, and it can never again reach a point where $v = 0$. Indeed, as t increases, this solution will penetrate the set of points determined by $|x| = k$, since by assumption, $\dot{v}_{(21)} < 0$ along this solution and since $v < 0$ in D. But this shows that the equilibrium $x_e = 0$ of system (21) is unstable. This discussion leads us yet to another conjecture:

3. Assume there exists a continuously differentiable function $v : R^2 \to R$ with the following properties:
   (i) there exist points x arbitrarily close to the origin such that $v(x) < 0$, which form the domain D which is bounded by the set of points determined by $v = 0$ and the disk $|x| = k$;
   (ii) in the interior of D, v is bounded; and
   (iii) in the interior of D, $\dot{v}_{(21)}$ is negative.

   Then the equilibrium $x_e = 0$ of system (21) is unstable.

THE PRINCIPAL LYAPUNOV AND LAGRANGE STABILITY THEOREMS

It turns out that results of the type presented in the previous section for system (21) are true for general systems given by system (E). This is true for the case of Lyapunov stability and Lagrange stability. These results comprise the Lyapunov Method, or the Second Method of Lyapunov, or the Direct Method of Lyapunov of qualitative analysis of dynamical systems. The reason for the latter name is clear: results of the kind considered here allow us to make qualitative statements about entire families of solutions of system (E) without actually solving this equation.

In the following, we summarize most of the important Lyapunov and Lagrange stability results for dynamical systems determined by system (E). Their proofs can be found in many texts on ordinary differential equations or on the stability of dynamical systems. We shall cite some of these sources when discussing the literature on the present subject.

In each of the following statements, we shall assume the existence of a continuously differentiable function $v : B(h) \times R^+ \to R$ for some $h > 0$, or $v : R^n \times R^+ \to R$, as needed.

1. If v is positive definite and $\dot{v}_{(E)}$ is negative semidefinite (or identically zero), then the equilibrium $x_e = 0$ of system (E) is stable.

2. If v is positive definite and decrescent and $\dot{v}_{(E)}$ is negative semidefinite (or identically zero), then the equilibrium $x_e = 0$ of system (E) is uniformly stable.

3. If v is positive definite and decrescent and $\dot{v}_{(E)}$ is negative definite, then the equilibrium $x_e = 0$ of system (E) is uniformly asymptotically stable.

4. If v is positive definite, decrescent, and radially unbounded and $\dot{v}_{(E)}$ is negative definite for all $(x, t) \in R^n \times R^+$, then the equilibrium $x_e = 0$ of system (E) is uniformly asymptotically stable in the large.

5. If there exist three positive constants c_1, c_2, c_3 such that

$$\begin{aligned} c_1|x|^2 \le v(x,t) \le c_2|x|^2 \\ \dot{v}_{(E)}(x,t) \le -c_3|x|^2 \end{aligned} \tag{22}$$

for all $t \in R^+$ and all $x \in B(r)$ for some $r > 0$, then the equilibrium $x_e = 0$ of system (E) is exponentially stable.

6. If there exist three positive constants c_1, c_2, c_3 such that system (22) holds for all $t \in R^+$ and all $x \in R^n$, then the equilibrium $x_e = 0$ of system (E) is exponentially stable in the large.

7. If v is decrescent and $\dot{v}_{(E)}$ is positive definite (resp., negative definite) and if in every neighborhood of the origin there are points x such that $v(x, t_0) > 0$ (resp., $v(x, t_0) < 0$), then the equilibrium $x_e = 0$ of system (E) is unstable (at $t = t_0 \ge 0$).

8. Assume that v is bounded on $D = \{(x, t) : x \in B(h), t \ge t_0\}$ and satisfies the following: (i) $\dot{v}_{(E)}(x, t) = \lambda v(x, t) + w(x, t)$,

where $\lambda > 0$ is a constant and $w(x, t)$ is either identically zero or positive semidefinite; (ii) in the set $D_1 = \{(x, t) : t = t_1, x \in B(h_1)\}$ for fixed $t_1 \geq t_0$ and with arbitrarily small h_1, there exist values x such that $v(x, t_1) > 0$. Then the equilibrium $x_e = 0$ of system (E) is unstable.

9. Assume that v satisfies the following properties:

(i) For every $\epsilon > 0$ and for every $t \geq 0$, there exist points $\bar{x} \in B(\epsilon)$ such that $v(\bar{x}, t) < 0$. We call the set of all points (x, t) such that $x \in B(h)$ and such that $v(x, t) < 0$ the "domain $v < 0$." This domain is bounded by the hypersurfaces that are determined by $|x| = h$ and $v(x, t) = 0$, and it may consist of several component domains.

(ii) In at least one of the component domains D of the "domain $v < 0$," v is bounded from below and $0 \in \partial D$ for all $t \geq 0$.

(iii) In the domain D, $\dot{v}_{(E)} \leq -\psi(|v|)$, where $\psi \in K$.

Then the equilibrium $x_e = 0$ of system (E) is unstable.

The next two results are typical of Lagrange-type stability results. In both of these results we assume that v is continuously differentiable and is defined on $|x| \geq R$, where R may be large, and $0 \leq t < \infty$.

10. Assume there exist $\psi_1, \psi_2 \in KR$ such that $\psi_1(|x|) \leq v(x, t) \leq \psi_2(|x|)$ and $\dot{v}_{(E)}(x, t) \leq 0$ for all $|x| \geq R$ and for all $0 \leq t < \infty$. Then the solutions of system (E) are uniformly bounded.

11. Assume there exist $\psi_1, \psi_2 \in KR$ and $\psi_3 \in K$ such that $\psi_1(|x|) \leq v(x, t) \leq \psi_2(|x|)$ and $\dot{v}_{(E)}(x, t) \leq -\psi_3(|x|)$ for all $|x| \geq R$ and $0 \leq t < \infty$. Then the solutions of system (E) are uniformly ultimately bounded.

We now apply some of the above results to some specific examples.

The system given by

$$\dot{x}_1 = x_2,\ \dot{x}_2 = -x_2 - e^{-t}x_1 \tag{23}$$

has an equilibrium at $(x_1, x_2)^T = (0, 0)^T$. We choose for system (23) the positive definite function $v(x_1, x_2, t) = x_1^2 + e^t x_2^2$ and obtain $\dot{v}_{(23)}(x_1, x_2, t) = -e^t x_2^2$ which is negative semidefinite. The result in item 1 above applies and we conclude that the equilibrium $x_e = 0$ of system (23) is stable.

We consider the simple pendulum considered earlier which is described by the equations

$$\dot{x}_1 = x_2,\ \dot{x}_2 = -k \sin x_1 \tag{24}$$

where $k > 0$ is a constant. As noted earlier, system (24) has an isolated equilibrium at $(x_1, x_2)^T = (0, 0)^T$. Choose $v(x_1, x_2) = \frac{1}{2}x_2^2 + k \int_0^{x_1} \sin \eta d\eta$, which is continuously differentiable and positive definite. We note that since v is independent of t, it is automatically decrescent. Furthermore, $\dot{v}_{(24)}(x_1, x_2) = (k \sin x_1)\dot{x}_1 + x_2\dot{x}_2 = (k \sin x_1)x_2 + x_2(-k \sin x_1) = 0$. The result in item 2 above applies and we conclude that the equilibrium $x_e = 0$ of system (24) is uniformly stable.

The system given by

$$\begin{aligned} \dot{x}_1 &= (x_1 - k_2 x_2)(x_1^2 + x_2^2 - 1) \\ \dot{x}_2 &= (k_1 x_1 + x_2)(x_1^2 + x_2^2 - 1) \end{aligned} \tag{25}$$

has an isolated equilibrium at $(x_1, x_2)^T = (0, 0)^T$. For system (25) we choose $v(x) = k_1 x_1^2 + k_2 x_2^2$ and obtain $\dot{v}_{(25)}(x_1, x_2) = 2(k_1 x_1^2 + k_2 x_2^2)(x_1^2 + x_2^2 - 1)$. If $k_1 > 0$, $k_2 > 0$, then v is positive definite (and decrescent) and $\dot{v}_{(25)}$ is negative definite over the domain $x_1^2 + x_2^2 < 1$. Accordingly, the result in item 3 above applies and we conclude that the equilibrium $(x_1, x_2)^T = (0, 0)^T$ is uniformly asymptotically stable.

The system given by

$$\begin{aligned} \dot{x}_1 &= x_2 + c x_1(x_1^2 + x_2^2) \\ \dot{x}_2 &= -x_1 + c x_2(x_1^2 + x_2^2) \end{aligned} \tag{26}$$

where c is a real constant, has only one equilibrium, which is located at the origin. For system (26) we choose the positive definite, decrescent, and radially unbounded function $v(x_1, x_2) = x_1^2 + x_2^2$ to obtain $\dot{v}_{(26)}(x_1, x_2) = 2c(x_1^2 + x_2^2)^2$. When $c = 0$, the result in item 2 above applies and we conclude that the equilibrium $(x_1, x_2)^T = (0, 0)^T$ is uniformly stable. If $c < 0$, then the result in item 4 above applies and we conclude that the trivial solution of system (26) is uniformly asymptotically stable in the large. If $c > 0$, then the result in item 7 above applies and we conclude that the trivial solution of system (26) is unstable.

For the system

$$\begin{aligned} \dot{x}_1 &= -a(t)x_1 - b x_2 \\ \dot{x}_2 &= b x_1 - c(t)x_2 \end{aligned} \tag{27}$$

b is a real constant and a and c are real and continuous functions defined for $t \geq 0$ and satisfying $a(t) \geq \delta > 0$ and $c(t) \geq \delta > 0$ for all $t \geq 0$. We assume that a, b and c are such that $x = 0$ is the only equilibrium for system (27). If we choose $v(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2)$, then $\dot{v}_{(27)}(x, t) = -a(t)x_1^2 - c(t)x_2^2 \leq -\delta(x_1^2 + x_2^2)$ for all $(x_1, x_2)^T \in R^2$ and for all $t \geq 0$. The result in item 6 above applies and we conclude that the equilibrium $(x_1, x_2)^T = (0, 0)^T$ of system (27) is exponentially stable in the large.

Consider the system

$$\begin{aligned} \dot{x}_1 &= x_1 + x_2 + x_1 x_2^4 \\ \dot{x} &= x_1 + x_2 - x_1^2 x_2 \end{aligned} \tag{28}$$

which has an isolated equilibrium $(x_1, x_2)^T = (0, 0)^T$. Choosing $v(x_1, x_2) = (x_1^2 - x_2^2)/2$, we obtain $\dot{v}_{(28)}(x_1, x_2) = \lambda v(x_1, x_2) + w(x_1, x_2)$, where $w(x_1, x_2) = x_1^2 x_2^4 + x_1^2 x_2^2$ and $\lambda = 2$. The result in item 8 above applies and we conclude that the equilibrium $(x_1, x_2)^T = (0, 0)^T$ is unstable.

Consider the system

$$\begin{aligned} \dot{x}_1 &= x_1 + x_2 \\ \dot{x}_2 &= x_1 - x_1 + x_1 x_2 \end{aligned} \tag{29}$$

which has an isolated equilibrium at the origin $(x_1, x_2)^T = (0, 0)^T$. Choosing $v(x_1, x_2) = -x_1 x_2$ we obtain $\dot{v}_{(29)}(x_1, x_2) = -x_1^2 - x_2^2 - x_1^2 x_2$. Let $D = \{(x_1, x_2)^T \in R^2 : x_1 > 0, x_2 > 0$, and $x_1^2 + x_2^2 < 1\}$. Then for all $(x_1, x_2)^T \in D$, $v(x_1, x_2) < 0$ and $\dot{v}_{(29)}(x_1, x_2) < 2v(x_1, x_2)$. We see that the result in item 9 above applies and conclude that the equilibrium $(x_1, x_2)^T = (0, 0)^T$ is unstable.

Consider the system

$$\dot{x} = -x - \sigma,\ \dot{\sigma} = -\sigma - f(\sigma) + x \tag{30}$$

where $f(\sigma) = \sigma(\sigma^2 - 6)$. This system has three isolated equilibria located at $x = \sigma = 0$, $x = -\sigma = 2$, and $x = -\sigma = -2$. Choosing the radially unbounded and decrescent function $v(x, \sigma) = \frac{1}{2}(x^2 + \sigma^2)$, we obtain $\dot{v}_{(30)}(x, \sigma) = -x^2 - \sigma^2(\sigma^2 - 5) \leq -x^2 - (\sigma^2 - \frac{5}{2})^2 + \frac{25}{4}$. Note also that $\dot{v}_{(30)}(x, \sigma)$ is negative for all (x, σ) such that $x^2 + \sigma^2 > R^2$, where, for example, $R = 10$ is an acceptable choice. Therefore, in accordance with the results given in items 10 and 11 above, all solutions of system (30) are uniformly bounded, in fact, uniformly ultimately bounded.

We conclude the present section by noting that the results given above in items 1–11 are also true when v is continuous, rather than continuously differentiable. In this case, $\dot{v}_{(E)}$ is interpreted as in system (17).

SOME EXTENSIONS AND FURTHER RESULTS

The body of work concerned with the Lyapunov Method is vast. In the following, we present a few additional rather well-known results.

For the case of autonomous systems given by

$$\dot{x} = f(x) \tag{A}$$

$f(0) = 0$, it is sometimes possible to relax the conditions on $\dot{v}_{(A)}$ (given in the previous section) when investigating the asymptotic stability of the equilibrium $x_e = 0$, by insisting that $\dot{v}_{(A)}$ be only negative semidefinite. In doing so, we require the following concept: a set $\Gamma \subset R^n$ is said to be invariant with respect to system (A) if every solution of system (A) starting in Γ remains in Γ for all time.

The following theorem is one of the results that comprise the Invariance Theory in the stability analysis of dynamical systems determined by system (A): Assume that there exists a continuously differentiable, positive definite, and radially unbounded function $v : R^n \to R$ such that

(i) $\dot{v}_{(A)}(x) \leq 0$ for all $x \in R^n$, and
(ii) the set $\{0\}$ is the only invariant subset of the set $E = \{x \in R^n : \dot{v}_{(A)}(x) = 0\}$.

Then the equilibrium $x_e = 0$ of system (A) is asymptotically stable in the large.

We apply the above invariance theorem in the analysis of the Lienard Equation given by

$$\dot{x}_1 = x_2,\ \dot{x}_2 = -f(x_1)x_2 - g(x_1) \tag{31}$$

where it is assumed that f and g are continuously differentiable for all $x_1 \in R$, $g(x_1) = 0$ if and only if $x_1 = 0$, $x_1 g(x_1) > 0$ for all $x_1 \neq 0$ and $x_1 \in R$,

$$\lim_{|x_1| \to \infty} \int_0^{x_1} g(\eta)\, d\eta = \infty$$

and $f(x_1) > 0$ for all $x_1 \in R$. Under these assumptions, the origin $(x_1, x_2)^T = (0, 0)^T$ is the only equilibrium of system (31).

Let us now choose the v function

$$v(x_1, x_2) = \frac{1}{2}x_2^2 + \int_0^{x_1} g(\eta)\, d\eta$$

which is positive definite and radially unbounded. Along the solutions of system (31) we have $\dot{v}_{(31)}(x_1, x_2) = -x_2^2 f_2(x_1) \leq 0$ for all $(x_1, x_2)^T \in R^2$. It is easy to see that in the present case the set E is the entire x_1-axis and that the largest invariant subset of the set E with respect to system (31) is the set $\{(0, 0)^T\}$. In view of the Invariance Stability Theorem given above, the origin $(x_1, x_2)^T = (0, 0)^T$ is asymptotically stable in the large.

The power, generality, and elegance of the Lyapunov Method must be obvious by now. However, this method has also weaknesses, the greatest drawback being that there exist no rules for choosing v-functions (Lyapunov functions). However, for the case of linear systems given by

$$\dot{x} = Ax \tag{L}$$

it is possible to construct Lyapunov functions in a systematic manner, in view of the following result.

Assume that the matrix A has no eigenvalues on the imaginary axis. Then there exists a Lyapunov function v of the form

$$v(x) = x^T Bx,\ B = B^T \tag{32}$$

whose derivative $\dot{v}_{(L)}$, given by

$$\dot{v}_{(L)} = -x^T Cx$$

where

$$-C = A^T B + BA \tag{33}$$

is definite (i.e., negative definite or positive definite).

In particular, the above results states that if all eigenvalues of A have negative real parts (i.e., the matrix A is stable), then for system (L), our earlier Lyapunov result for asymptotic stability in the large constitutes also the necessary conditions for asymptotic stability. In the same spirit, an instability results for system (L) can also be established. We will not pursue this, however.

In view of the above result, if for example, all eigenvalues of A have negative real parts, then the v-function [system (32)] is easily constructed by assuming a positive definite matrix $C = C^T$ and by solving the Lyapunov matrix equation [(system 33)] for the $n(n + 1)/2$ unknown elements of the symmetric matrix B (which in this case will be positive definite).

To simplify matters, we consider in the following autonomous systems described by

$$\dot{x} = f(x) \tag{A}$$

and we assume that $x_e = 0$ is an equilibrium of system (A) [i.e., $f(0) = 0$]. Now, when the origin is not the only equilibrium of system (A) and if $x_e = 0$ is asymptotically stable, then $x_e = 0$ cannot possibly be globally asymptotically stable. [There may be other reasons why an asymptotically stable equilibrium $x_e = 0$ of system (A) might not be globally asymptotically stable.] Under such conditions, it is of great interest to determine an estimate of the domain of attraction of the equilibrium $x_e = 0$ of system (A).

Now for purposes of discussion, let us assume that for system (A) there exists a Lyapunov function v that is positive

definite and radially unbounded. Also, let us assume that over some domain $D \subset R^n$ containing the origin, $\dot{v}_{(A)}(x)$ is negative, except at the origin, where $\dot{v}_{(A)} = 0$. Let $C_i = \{x \in R^n : v(x) \leq c_i\}$, $c_i > 0$. Using similar reasoning as was done in the analysis of the system (21), we can now show that as long as $C_i \subset D$, C_i will be a subset of the domain of attraction of $x_e = 0$. Thus, if $c_i > 0$ is the largest number for which this is true, then it follows that C_i will be contained in the domain of attraction of $x_e = 0$. The set C_i obtained in this manner will be the best estimate for the domain of attraction of $x_e = 0$ that can be obtained using our particular choice of v-function.

Above we pointed out that for system (L) there actually exist converse Lyapunov (asymptotic stability and instability) theorems. It turns out that for virtually every Lyapunov and Lagrange Stability Theorem given earlier, a converse can be established. Unfortunately, these Lyapunov converse theorems are of not much help in constructing v-functions in specific cases. For purposes of illustration, we cite in the following an example of such a converse theorem.

If f and $\partial f/\partial x$ are continuous on the set $B(r) \times R^+$ for some $r > 0$, and if the equilibrium $x_e = 0$ of system (E) is uniformly asymptotically stable, then there exists a Lyapunov function v which is continuously differentiable on $B(r_1) \times R^+$ for some $r_1 > 0$ such that v is positive definite and decrescent, and such that $\dot{v}_{(E)}$ is negative definite.

We conclude this section by addressing the following question: under what conditions does it make sense to linearize a nonlinear system about an equilibrium $x_e = 0$ and then deduce the stability properties of $x_e = 0$ from the corresponding linear system? Results that answer questions of this kind comprise Lyapunov's First Method or Lyapunov's Indirect Method.

To simplify our discussion, we consider autonomous systems (A),

$$\dot{x} = f(x) \tag{A}$$

we assume that f is continuously differentiable, and we assume that $f(0) = 0$, which means that $x_e = 0$ is an equilibrium for system (A).

A linearization process of system (A) about the equilibrium $x_e = 0$ results in the representation of system (A) by

$$\dot{x} = Ax + F(x) \tag{34}$$

where

$$A = \frac{\partial f}{\partial x}(0)$$

denotes the Jacobian of $f(x)$ evaluated at $x = 0$, and where

$$\lim_{|x| \to 0} \frac{|F(x)|}{|x|} = 0 \tag{35}$$

Associated with system (34) [respectively, system (A)], we have the system

$$\dot{y} = Ay \tag{36}$$

which is called the linearization of system (A).

Now suppose that the matrix A in system (36) is stable (i.e., all eigenvalues of A have negative real parts). According to results given above, we can construct in this case a Lyapunov function of the form (32) for system (36). Utilizing this Lyapunov function in the analysis of the nonlinear system (34) [and hence, of the original system (A)], and applying Lyapunov's asymptotic stability theorem that was presented earlier, the following result is established:

Assume that for the real $n \times n$ matrix A all eigenvalues have negative real parts and let $F : R^n \to R^n$ be continuous and satisfy system (35). Then the equilibrium $x_e = 0$ of system (34) [and hence, of system (A)] is asymptotically stable.

An instability theorem in the spirit of the above result has also been established. In fact, for system (E), theorems along the lines of the above results have been established as well.

We close the present section by considering the following version of the Lienard equation,

$$\dot{x}_1 = x_2,\ \dot{x}_2 = -x_1 - f(x_1)x_2 \tag{37}$$

where f is assumed to be continuously differentiable and $f(0) > 0$. The origin is clearly an equilibrium of system (37),

$$J(0) = A = \begin{bmatrix} 0 & 1 \\ -1 & -f(0) \end{bmatrix}$$

and the eigenvalues of A are given by $\lambda_1,\ \lambda_2 = [-f(0) \pm \sqrt{f(0)^2 - 4}]/2$. These have clearly negative real parts. Furthermore, it is easily verified that system (35) is satisfied. It follows that the trivial solution of system (37) is asymptotically stable. It must be emphasized, however, that this analysis by the First Method of Lyapunov does not yield any information whatsoever about the domain of attraction of the equilibrium $x_e = 0$ of system (37). This is true, in general.

SOME NOTES AND REFERENCES

Reference 1 is a translation of a paper from the Russian that originally had appeared in 1893 in a mathematics journal in Kharkow (*Comm. Soc. Math. Kharkow*). In this paper, A. M. Lyapunov developed a highly original approach for the stability analysis of an equilibrium of systems described by ordinary differential equations, which today bears his name under several variants: Lyapunov's Second Method, Lyapunov's Direct Method, Lyapunov's Method, and so forth. (In the present article, Lyapunov's First Method is also included.) It is interesting to note that since the motivation for his work was an analysis of the motions of planets, Lyapunov was actually more interested in the stability (and instability) of an equilibrium, rather than in asymptotic stability. For an account of early work on this subject, the reader may want to consult the book by Bennett (2), and some of the sources cited therein.

Since 1893, the Lyapunov approach has been extended, generalized, and improved in numerous ways, and the literature on this subject has experienced phenomenal growth, especially in recent times. Results that are in the spirit of those presented herein have been discovered for general dynamical systems (3–5), and for more specific classes of infinite dimensional systems (6) and finite dimensional systems (7,8).

Perhaps the greatest driving force behind the development of Lyapunov's Method was the significant progress that has been made since World War II in feedback control systems.

For a brief description of this, the reader may wish to consult Ref. 9 and the sources cited in that paper.

One of the early important problems in feedback control concerns the absolute stability of regulator systems. It is fair to say that most of the progress that was made toward solving this class of problems was accomplished by means of Lyapunov's approach (10–12). Another important class of feedback problems treated primarily by the Lyapunov Method was the systematic stability analysis of complex, large-scale dynamical systems (13–15), with specific applications in such diverse areas as power systems (16) and artificial neural networks (17). These two classes of systems are only a small sample where the Lyapunov approach has been effective in stability analysis. There are many other such classes, too numerous to cite here. The reader may want to consult some of the contemporary texts in control systems to obtain additional insights into this subject (18–20).

BIBLIOGRAPHY

1. A. M. Lyapunov, Problème général de la stabilité du movement, *Ann. Fac. Sci. Toulouse,* **9**: 203–474, 1907.
2. S. Bennett, *A History of Control Engineering,* London: Peter Peregrinus, 1979.
3. V. I. Zubov, *Methods of A. M. Lyapunov and their Applications,* Groningen: Noordhoff, 1964.
4. W. Hahn, *Stability of Motion,* New York: Springer-Verlag, 1967.
5. A. N. Michel and K. Wang, *Qualitative Theory of Dynamical Systems,* New York: Marcel Dekker, 1995.
6. V. Lakshmikantham and S. Leela, *Differential and Integral Inequalities,* Vols. I and II, New York: Academic Press, 1969.
7. J. P. LaSalle and S. Lefschetz, *Stability by Liapunov's Direct Method,* New York: Academic Press, 1961.
8. R. K. Miller and A. N. Michel, *Ordinary Differential Equations,* New York: Academic Press, 1982.
9. A. N. Michel, Stability: the common thread in the evolution of feedback control, *IEEE Control Systems Magazine,* **16**: 50–60, 1996.
10. M. A. Aizerman and F. R. Gantmacher, *Absolute Stability of Regulator Systems,* San Francisco: Holden-Day, 1964.
11. S. Lefschetz, *Stability of Nonlinear Control Systems,* New York: Academic Press, 1965.
12. K. S. Narendra and J. H. Taylor, *Frequency Domain Stability for Absolute Stability,* New York: Academic Press, 1973.
13. A. N. Michel and R. K. Miller, *Qualitative Analysis of Large Scale Dynamical Systems,* New York: Academic Press, 1977.
14. D. D. Siljak, *Large-Scale Dynamical Systems: Stability and Structure,* New York: North Holland, 1978.
15. Lj. T. Grujic, A. A. Martynyuk, and M. Ribbens-Pavella, *Large Scale Systems Stability under Structural and Singular Perturbations,* New York: Springer-Verlag, 1987.
16. M. A. Pai, *Power System Stability,* New York: North-Holland, 1981.
17. A. N. Michel and J. A. Farrell, Associative memories via artifical neural networks, *IEEE Control Systems Magazine,* **10**: 6–17, 1990.
18. H. K. Khalil, *Nonlinear Systems,* New York: Macmillan, 1992.
19. M. Vidyasagar, *Nonlinear Systems Analysis,* Englewood Cliffs, NJ: Prentice Hall, 1993.
20. P. J. Antsaklis and A. N. Michel, *Linear Systems,* New York: McGraw-Hill, 1997.

ANTHONY N. MICHEL
University of Notre Dame

LYAPUNOV STABILITY. See LYAPUNOV METHODS.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Minimization
Standard Article

V. C. Ramesh[1]
[1]Illinois Institute of Technology, Chicago, IL

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2439
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (109K)

Abstract

The sections in this article are

Problem Formulation

A Fuzzy Model

Obtaining Fuzzy Tolerance Parameters

The Algorithm

Conclusion

| | |

Copyright © 1999-2008 All Rights Reserved.

MINIMIZATION

This article discusses optimization problems involved in real-time control of systems with a human operator in the loop. Such real-time systems involve multiple objective functions, and there are multiple optimal solutions. Often these objectives are conflicting; an example is planning for contingencies as well as the "base case" (current operating point). The solution to such multiobjective problems is typically a trade-off surface (known as a Pareto surface) whose axes are the various objective functions. The Pareto surface is to be presented to the human operator who will make the final decisions regarding control actions. Also, ideally the operator will be able to modify the parameters of the optimization process interactively to obtain desired results.

Mathematical optimization techniques such as nonlinear programming (NLP) have been used historically as the building blocks of real-time control systems. However, their inadequacies have been felt most acutely in the modeling of realistic control actions that do not fit well in the traditional optimization frameworks. The deficiencies are particularly notable in the following areas:

1. Modeling uncertainties in input data pertaining to system operation.
2. Modeling of "soft constraints."
3. Filtering and ranking control actions that the operator is expected to perform.
4. Modeling discrete control actions.
5. Modeling the "level of risk" that the operator is willing to take while planning for contingencies.

Let us consider these deficiencies in turn. Uncertainties in input data result from many sources. Chief among them are uncertainties introduced by inaccurate and/or imprecise sensory data. Such uncertainties are particularly evident in control systems designed for geographically distributed physical networks such as electric power grids and air-traffic control. Traditionally, such uncertainties have been handled through the use of probabilistic techniques, resulting in stochastic optimization models. Recently, fuzzy logic has emerged as a feasible alternative for modeling data uncertainties in optimization of physical systems (1).

Several decades ago, March and Simon (2) argued that human decision-makers usually "satisfice" rather than "optimize." Traditional optimization models treat most problem constraints as rigid constraints whose violations are impermissible. This is not satisfactory for two reasons. First, the uncertainties in the parameters of the underlying physical system naturally lead to situations where violations are tolerated for gains in the objective. Second, one of the popular ways of modeling multiple conflicting objectives is the "constraint method" which converts secondary objectives to constraints (with specified tolerances that are to be minimized) (3). Thus, such "soft constraints" need to be modeled explicitly in the optimization problem. Penalty factor techniques and fuzzy-logic-based methods have been shown to be effective for modeling such soft constraints (4,5).

Operators can perform only a limited number of control actions in time-critical situations. Hence, the solutions of optimization methods need to be such that a small number of actions with different priorities are suggested. Again, traditional optimization techniques are not completely suitable. While postprocessing of solutions using technologies such as expert systems is feasible, this will likely compromise the optimality of the solutions. It is preferable to start from a model that incorporates such filtering and prioritization.

Discrete control actions are often more effective than their continuous counterparts when quick changes are needed. However, such actions are typically avoided in many optimization models because they result in mixed-integer nonlinear programming (MINLP) models that are often very difficult to solve. This is unfortunate since the results of enhanced models can lead to more efficient operations. Operators may be forced to use such actions without the aid of optimization models. What is needed is a fresh look at integration of discrete actions in continuous models without necessarily leading to rigorous MINLP problems. Techniques such as fuzzy logic can provide help in this respect.

Discrete actions can introduce multiple minima that reflect practical solution alternatives. Finding the global minimum can be a daunting task and is the subject of an entire field of the relatively new field of global optimization. However, this possibility needs to be carefully considered since it can result in significantly better solutions.

Planning for contingencies constitutes an important aspect of control and operation of complex real-time systems. However, contingency planning has been difficult to tackle through traditional optimization models because of the inability of these models to account for the subjective "risk preferences" of the operators. Risk assessment and management is a fundamental component of contingency planning. The trade-off is between economics and "security" against contingencies.

The first two deficiencies have been addressed in the general fuzzy logic literature, and fuzzy techniques have been proposed for system operation with uncertain data and soft constraints. The last three deficiencies have not received as much attention, and they are still unresolved problems. In this article, we concentrate on the modeling issues pertaining to the last deficiency. We refer the reader to Ref. 6 for an elaborate discussion on the other two deficiencies (i.e., the third and the fourth deficiency).

PROBLEM FORMULATION

The conventional optimization problem for a control system can be formulated as:

$$\min_{Z} f(Z) \quad \text{s.t.} \quad G(Z) = 0, \qquad H(Z) \leq 0 \tag{1}$$

where

U is the vector of control variables
X is the vector of state variables
$Z = [u, X]^{\mathrm{T}}$ is the vector of all the decision variables

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

G is the set of system equations
H is the set of operation limits
$f(Z)$ is the objective function (usually the cost) which is to be minimized

To account for the contingencies that can occur in the system (i.e., for contingency constrained optimization), the problem can be formulated as the following decomposed multiobjective problem.

Base-Case Subproblem

$$\begin{aligned} &\underset{Z_0}{\text{Min}}\, f(Z_0) \\ &\underset{Z_0}{\text{Min}}\, \|U_0 - U_k^*\|^2, \qquad k = 1, 2, \ldots, N \\ &\text{s.t.} \quad G_0(Z_0) = 0, \qquad H_0(Z_0) \le 0 \end{aligned} \tag{2}$$

where

$Z_0 = [U_0, X_0]^T$ is the base-case decision vector
U_k^* is the latest value of the kth subproblem control variables
G_0 is the set of system equations for the base case
H_0 is the set of operating limits for the base case
N is the number of contingencies

***N* Contingency Subproblems**

$$\underset{Z_k}{\text{Min}}\, \|U_0^* - U_k\|^2 \quad \text{s.t.} \quad G_k(Z_k) = 0, \qquad H_k(Z_k) \le 0 \tag{3}$$

where

U_0^* is the latest value of the base case control variables
$Z_k = [U_k, X_k]^T$ is the decision vector for the kth subproblem
G_k is the set of system equations for the kth subproblem
H_k is the set of operating limits for the kth subproblem

A FUZZY MODEL

Let $f(Z_0) \le C_0 + \delta_c$ represent an imprecise upper bound on the maximum permissible operating cost, where C_0 is the optimal cost obtained by solving the general constrained optimization problem without contingency constraints, and δ_c is the "tolerance" parameter that is a measure of the fuzziness in this constraint. So the fuzzy goal is to keep $f(Z_0)$ "as close to" C_0 as possible, but no greater than $C_0 + \delta_c$.

Let μ_c be the membership function that represents the extent to which a given $f(Z_0)$ satisfies the fuzzy goal. Such a membership function can take any value in [0, 1]. The higher its value, the greater the degree of satisfaction of the fuzzy goal by the given $f(Z_0)$. If we assume that the operator's satisfaction decreases linearly with deviation from C_0, we can use the following mathematical formulation for this (linear) membership function:

$$\mu_c(f(Z_0)) = \begin{cases} 1 & \text{if } f(Z_0) \le C_0 \\ \dfrac{(C_0 + \delta_c - f(Z_0))}{\delta_c} & \text{if } C_0 < f(Z_0) < C_0 + \delta_c \\ 0 & \text{if } f(Z_0) \ge C_0 + \delta_c \end{cases} \tag{4}$$

Let η_k represent either of the functions $\|U_0^* - U_k\|^2$ or $\|U_k^* - U_0\|^2$. Let Δ be the vector of the ramp limits (the increase or decrease rate) for the control variables. Then the fuzzy ramping constraints can be formulated as $\eta_k < \|\Delta\|^2 + \delta_\Delta$, where δ_Δ is a parameter that specifies the tolerance on the violation of this constraint. The corresponding fuzzy goal is to keep η_k "as close to" $\|\delta_\Delta\|^2$ as possible but no greater than $\|\Delta\|^2 + \delta_\Delta$. We can define a linear membership function $\mu_\Delta(\eta_k)$, as follows:

$$\mu_\Delta(\eta_k) = \begin{cases} 1 & \text{if } \eta_k \le \|\Delta\|^2 \\ \dfrac{(\|\Delta\|^2 + \delta_\Delta - \eta_k)}{\delta_c} & \text{if } \|\Delta\|^2 < \eta_k < \|\Delta\|^2 + \delta_\Delta \\ 0 & \text{if } \eta_k \ge \|\Delta\|^2 + \delta_\Delta \end{cases} \tag{5}$$

Both the membership functions are displayed in Fig. 1.

Let $\eta_k(U_0)$ represent $\|U_k^* - U_0\|^2$, and let $\eta_k(U_k)$ represent $\|U_0^* - U_k\|$. Our fuzzy formulation of the decomposition presented in the previous section is the following:

Base-Case Subproblem

$$\begin{aligned} &\underset{Z_0}{\text{Max}}\, \text{Min}\{\mu_c(f(Z_0)),\, \mu_\Delta(\eta_k(U_0)),\, k = 1, \ldots, N\} \\ &\qquad \text{s.t.} \quad G_0(Z_0) = 0, \qquad H_0(Z_0) \le 0 \end{aligned} \tag{6}$$

In Eq. (6), the Min operator is used to represent the intersection of the $N + 1$ fuzzy sets corresponding to the two membership functions. The resulting Max–Min formulation aims to find an operating point that maximizes the degree of satisfaction of the least satisfied fuzzy relation, for a given set of N postcontingency operating points. This is one way to seek a compromise between the $N + 1$ fuzzy relations.

***N* Contingency Subproblems**

$$\begin{aligned} &\underset{Z_k}{\text{Max}}\, \mu_\Delta(\eta_k(U_0)) \\ &\text{s.t.} \quad G_k(Z_k) = 0, \qquad H_k(Z_k) \le 0 \end{aligned} \tag{7}$$

In solving each contingency subproblem, we seek to find a postcontingency operating point that is "closest" (per the Euclidean norm) to the given base case, in control space. This is tantamount to maximizing the degree of satisfaction of the corresponding membership function.

To solve this fuzzy model using standard optimization methods, we need to convert Eq. (4) to an equivalent "crisp"

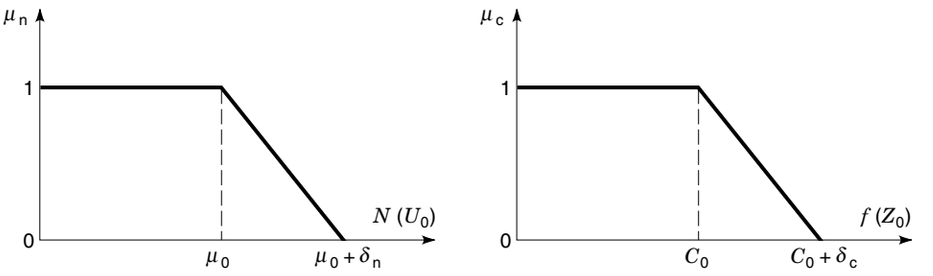


Figure 1. Membership functions for our fuzzy goals.

formulation. To do this, we introduce $N + 1$ (one for each subproblem) membership variables, β_k, as follows:

Base-Case Subproblem

$$\begin{aligned} &\underset{Z_0,\beta_0}{\text{Max}}\ \beta_0 \\ &\text{s.t.}\quad f(Z_0) + \delta_c\beta_0 \le C_0 + \delta_c \\ &\qquad \eta_k(U_0) + \delta_\Delta\beta_0 \le \|\Delta\|^2 + \delta_\Delta, \qquad k = 1, \ldots, N \\ &\qquad G_0(Z_0) = 0, \qquad H_0(Z_0) \le 0, \qquad 0 \le \beta_0 \le 1 \end{aligned} \tag{8}$$

N Contingency Subproblems

$$\begin{aligned} &\underset{Z_k,\beta_k}{\text{Max}}\ \beta_k \\ &\text{s.t.}\quad \eta_k(U_k) + \delta_\Delta\beta_k \le \|\Delta\|^2 + \delta_\Delta\|^2 + \delta_\Delta, \qquad k = 1, \ldots, N \\ &\qquad G_k(Z_k) = 0, \qquad H_k(Z_k) \le 0, \qquad 0 \le \beta_k \le 1 \end{aligned} \tag{9}$$

OBTAINING FUZZY TOLERANCE PARAMETERS

In this section, we describe a procedure for obtaining the parameters, δ_c and δ_Δ, used in the model presented above. This procedure is based on the work of Zimmermann (1).

1. Solve the base case in Eq. (2) for the first objective (ignoring the second one) to get U_0^* and $f(U_0^*)$. $C_0 = f(U_0^*)$.
2. For each contingency k, using U_0^* obtained from 1:
   i. Solve Eq. (3) to get U_k^*.
   ii. Solve the second objective in Eq. (2) subject to the constraints to get U_{0k}^*.
   iii. Calculate $f(U_{0k}^*)$.

Then, $C_0 + \delta_c = \text{Max}\{f(U_{0k}^*), k = 1, \ldots, N\}$ and $\|\Delta\|^2 + \delta_\Delta = \text{Max}\{\|U_k^* - U_0^*\|^2, k = 1, \ldots, N\}$.

The idea is to set the upper bound on the base-case cost equal to the maximum deviation from the optimal cost that is necessary to minimize the correction time of any of the contingencies. Similarly, the upper bound on the correction time is equal to the maximum of the correction times of any of the contingencies, corresponding to the least-cost base-case point. Δ is determined from the maximum correction times specified by the operator.

We would also expect to query the operator regarding our choice of a linear function to reflect the rate of decrease in the degrees of satisfaction. This can be done by picking a couple of points within the bounds and asking the operator for the change in the satisfaction compared with one of the bounds. If a linear function is not found to be appropriate, it can be replaced by one of the other functions reported in the fuzzy logic literature such as hyperbolic or exponential (7). Once the bounds are known, this is relatively easy to do.

THE ALGORITHM

Let $\Delta\beta_k$ represent the change in β_k between two successive iterations. Let ϵ represent a "termination" parameter. Then the algorithm for our fuzzy approach is as follows:

Step 1. Given $\|\Delta\|^2$, use the above procedure to get δ_c, δ_Δ, and U_0^* corresponding to C_0.

Step 2. Given U_k^*, solve the base-case subproblem in Eq. (8) to obtain Z_0^* and β_0.

Step 3. Given U_0^*, solve the N contingency subproblems in Eq. (9) to obtain Z_k^* and β_k for $k = 1, \ldots, N$.

Step 4. If $\|\Delta\beta_i\| < \epsilon$, for all $i = 0, \ldots, N$, stop; else go to step 2.

The use of the above algorithm for a real-world example (an electric power system control problem) is described in Ref. 8.

CONCLUSION

This article discusses a method for real-time optimization problems. The models described here have been applied to control of electric power networks and are discussed in detail in Refs. 8–10. We believe that modeling deficiencies in many such optimization problems can be fruitfully addressed using fuzzy logic.

BIBLIOGRAPHY

1. H. J. Zimmermann, *Fuzzy Set Theory and Its Applications,* Hingham, MA: Kluwer-Nijhoff, 1985.
2. J. G. March and H. A. Simon, *Organizations,* New York: Wiley, 1958.
3. M. Zeleny, *Multiple Criteria Decision Making,* New York: McGraw-Hill, 1982.
4. V. C. Ramesh and S. N. Talukdar, A parallel asynchronous decomposition for on-line contingency planning, *Proc. PICA,* 1995, pp. 243–248.
5. S. N. Talukdar and V. C. Ramesh, A multi-agent technique for contingency constrained optimal power flows, *IEEE Trans. Power Syst.,* **9** (2): 855–861, 1994.
6. V. C. Ramesh and X. Li, Strategies for improved contingency planning, *Inf. Syst. Eng.,* **2** (3–4): 183–193, 1996.

7. M. Sakawa, *Fuzzy Sets and Interactive Multiobjective Optimization,* New York: Plenum Press, 1993.
8. V. C. Ramesh and X. Li, A fuzzy multiobjective approach to contingency constrained OPF, *IEEE Trans. Power Syst.,* **12**: 1348–1354, 1997.
9. V. C. Ramesh and Xuan Li, Optimal power flow with fuzzy emission constraints, *Elec. Mach. Power Syst.,* **25** (8): 897–906, 1997.
10. V. C. Ramesh and X. Li, Towards intelligent optimization models for operator assistance, *Eng. Intell. Syst.,* **4** (4): 227–233, 1996.

V. C. RAMESH
Illinois Institute of Technology

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Minmax Techniques
Standard Article

Germano Lambert-Torres[1],
João Onofre Pereira Pinto[2],
Luiz Eduardo Borges da Silva[3]
[1]Escola Federal de Engenharia de Itajubá, Itajubá, MG, Brazil
[2]The University of Tennessee at Knoxville, Knoxville, TN
[3]Escola Federal de Engenharia de Itajubá, Itajubá, MG, Brazil

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2438
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (529K)

Abstract

The sections in this article are

Basic Operations With Ordinary Sets

Fuzzy Sets

Basic Concepts of Fuzzy Statements

Ordinary and Fuzzy Relations

Composition of Two Relations

An Illustrative Example

| | |

Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

MINMAX TECHNIQUES

In real life, people need to make decisions based on facts such as measurements, scheduling, short- and long-term forecasting, and guessing. These decisions may be related to management, security, economics, and education, and so forth. However, in the last few years, the decision process has become more complex due to the large amount of information associated with each step of the decision-making process.

Usually, in order to help human operators make the right decision at the right time, there is a collection of operational instructions, standards, computer programs, and other information. However, sometimes these regulations and standards represent the main drawback for good and reliable decision making, because they have been obtained from the analysis of a specific situation or a certain particular study and hence do not apply to a different event. In this case the available information tends to push the operator to make a wrong decision. To cope with the decision-making problem in a complex environment, new mathematical tools have been developed to help create more flexible, friendly, and easy-to-build computerized decision-making systems. Among these tools, the fuzzy set theory plays a very important role today when the values assumed by the variables are linguistic values such as "small," "big," "warm," "cold," and "close." In the last few years, the reported new applications of this theory have reached an impressive number of areas. Industrial automation, equipment automation, expert systems, medical diagnostics, and control systems are some of the areas to which fuzzy sets are intensively applied today. The fuzzy set theory (1) was proposed as a step toward modeling the pervasive imprecision of the real world.

This article presents the theory of fuzzy sets and develops the fuzzy technique. *Fuzzy technique* is the name given to the process through which fuzzy set theory is applied to problems of the real world. Initially, some basic aspects of ordinary sets are presented, and then brief concepts of fuzzy set theory are addressed. Next, the fuzzy technique is presented and sequentially developed. Finally, an illustrative example of the fuzzy technique is presented.

Basic Operations With Ordinary Sets

Let U be a set of elements representing the universe of discourse and A and B subsets of U. Table 1 presents well-known operations and properties of these two subsets. One of these properties is the *intersection* $(A \cap B)$, which can be expressed in linguistic terms by the conjunction ***and***. Another property, the *union* between two sets $(A \cup B)$, can be expressed by the conjunction ***or***.

It is possible to establish a relationship between the properties *intersection* and *union* proposed in Table 1 and the classical Boolean arithmetic. *Intersection* can be expressed by a Boolean product, while the *union* is a Boolean sum, as shown in Eqs. (1) to (4), and Tables 2 and 3.

$$x \in (\boldsymbol{A} \cap \boldsymbol{B}) \quad \text{if} \quad x \in \boldsymbol{A} \text{ and } x \in \boldsymbol{B} \tag{1}$$

Table 1: Basic Operations and Properties

Properties of $\oslash$ and $\mathbf{U}$
$\boldsymbol{A} \cup \oslash = \boldsymbol{A} \quad \boldsymbol{A} \cap \oslash = \oslash \quad \boldsymbol{A} \cup \boldsymbol{U} = \boldsymbol{U} \quad \boldsymbol{A} \cap \boldsymbol{U} = \boldsymbol{A}$
Idempotent Properties
$\boldsymbol{A} \cup \boldsymbol{A} = \boldsymbol{A} \quad \boldsymbol{A} \cap \boldsymbol{A} = \boldsymbol{A}$
Commutative Properties
$\boldsymbol{A} \cup \boldsymbol{B} = \boldsymbol{B} \cup \boldsymbol{A} \quad \boldsymbol{A} \cap \boldsymbol{B} = \boldsymbol{B} \cap \boldsymbol{A}$
Associative Properties
$(\boldsymbol{A} \cup \boldsymbol{B}) \cup \boldsymbol{C} = \boldsymbol{A} \cup (\boldsymbol{B} \cup \boldsymbol{C}) \quad (\boldsymbol{A} \cap \boldsymbol{B}) \cap \boldsymbol{C} = \boldsymbol{A} \cap (\boldsymbol{B} \cap \boldsymbol{C})$
Distributive Properties
$(\boldsymbol{A} \cup \boldsymbol{B}) \cap \boldsymbol{C} = (\boldsymbol{A} \cap \boldsymbol{C}) \cup (\boldsymbol{B} \cap \boldsymbol{C}) \quad (\boldsymbol{A} \cap \boldsymbol{B}) \cup \boldsymbol{C} = (\boldsymbol{A} \cup \boldsymbol{C}) \cap (\boldsymbol{B} \cup \boldsymbol{C})$
Properties of Complement (Taking $\boldsymbol{U}$ as the Whole Universe)
$\neg\oslash = \boldsymbol{U} \quad \neg\boldsymbol{U} = \oslash \quad \boldsymbol{A} \cup \neg\boldsymbol{A} = \boldsymbol{U} \quad \boldsymbol{A} \cap \neg\boldsymbol{A} = \oslash$
De Morgan's Laws
$\neg(\boldsymbol{A} \cup \boldsymbol{B}) = \neg\boldsymbol{A} \cap \neg\boldsymbol{B} \quad \neg(\boldsymbol{A} \cap \boldsymbol{B}) = \neg\boldsymbol{A} \cup \neg\boldsymbol{B}$

$$\mu_{\boldsymbol{A}\cap\boldsymbol{B}}(x) = \mu_{\boldsymbol{A}}(x)\mu_{\boldsymbol{B}}(x) \tag{2}$$

$$x \in (\boldsymbol{A} \cup \boldsymbol{B}) \quad \text{if} \quad x \in \boldsymbol{A} \text{ or } x \in \boldsymbol{B} \tag{3}$$

$$\mu_{\boldsymbol{A}\cup\boldsymbol{B}}(x) = \mu_{\boldsymbol{A}}(x) + \mu_{\boldsymbol{B}}(x) \tag{4}$$

where $\mu_{\boldsymbol{A}}(x)$ represents the degree of the membership of x in a set $\boldsymbol{A}$, i.e., the value of $\mu_{\boldsymbol{A}}(x)$ (that can be 0 or 1) is equal to 0 if x is not member of $\boldsymbol{A}$ and 1 if x is member of $\boldsymbol{A}$. For example, let $\boldsymbol{U}$ and $\boldsymbol{A}$ be defined as follows:

$$\begin{aligned} \boldsymbol{U} &= \{x_1, x_2, x_3, x_4, x_5\} \\ \boldsymbol{A} &= \{x_1, x_3, x_4\} \end{aligned}$$

The values of $\mu_A(x_1)$, $\mu_A(x_3)$ and $\mu_{\boldsymbol{A}}(x_4)$ are equal to 1 if x_1, x_3; and x_4 are members of set A and 0 if not.

Table 2: Intersection Property as a Boolean Product

Intersection Minimum Value		$\mu_A(x)$ 0	1
$\mu_B(x)$	0	0	0
	1	0	1

Table 3: Intersection Property as a Boolean Product

Intersection Maximum Value		$\mu_A(x)$ 0	1
$\mu_B(x)$	0	0	1
	1	1	1

Fuzzy Sets

The notion of *fuzzy sets* was proposed by Lotfi A. Zadeh in 1965 (1). In conventional Boolean algebra, one manipulates the values of membership $\mu_{\boldsymbol{A}}(x)$ of x in $\boldsymbol{A}$, assuming just the values 0 (zero) or 1 (one). However, the fuzzy set theory, as proposed by Zadeh, assumes all values between 0 and 1 [$0 \leq \mu_{\boldsymbol{A}}(x) \leq 1$]. In other words, Zadeh's definition of fuzzy sets assumes that $\mu_{\mathbf{A}}(x)$ may assume any value, for example, from a set $\boldsymbol{M} = [0,1]$.

Using this concept, it is possible to introduce the idea of a *linguistic variable* or *fuzzy variable*. A fuzzy variable is a nondeterministic variable that assumes a certain fuzzy subset $\boldsymbol{A}$ defined by its membership function $\mu_{\mathbf{A}}(x)$ instead of numerical values. To describe the proposed concept in mathematical language, one may write

$$\{(x|\mu_{\boldsymbol{A}}(x))\}, \quad \text{for all } x \text{ belonging to } \boldsymbol{U} \tag{5}$$

where $\boldsymbol{U}$ is the universe of discourse. This mathematical expression means: for every x belonging to $\boldsymbol{U}$, it is possible to associate a value of membership $\mu_{\mathbf{A}}(x)$ of x in a defined subset $\boldsymbol{A}$ of $\boldsymbol{U}$.

Therefore, the value of a fuzzy variable is given by a subset of a certain universe of discourse, normally described by words used in day-to-day language. Therefore, the value assumed by a fuzzy variable can also be given by a word in a natural language, such as "small," "big," "tall," "old," "cold," or "hot." Each word is described by a specific subset defined by a particular membership function. Figure 1 shows a graphic description of the membership functions for a fuzzy variable x representing the size of a certain piece of hardware. This fuzzy variable can assume values of "small," "medium," and "big." One can say "the value of x is small," where "the value of x" is the fuzzy variable representing the size and "small" is the fuzzy value. Let the universe of discourse of the variable "size" of a certain piece of hardware be represented by $\boldsymbol{U} = \{0,1,\ldots,10\}$, then the

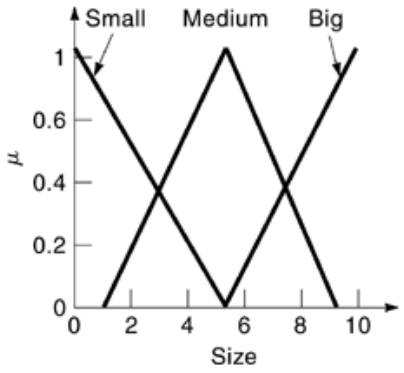


Fig. 1. Fuzzy subsets: small, medium, and big.

membership function of the values of "size" can, for example, be given by

$$\begin{aligned}
\text{small} &= \{(0|1), (1|0.8), (2|0.6), (3|0.4), (4|0.2), (5|0).(6|0), (7|0), (8|0), (9|0), (10|0)\} \\
\text{medium} &= \{0|0), (1|0), (2|0.25), (3|0.5), (4|0.75), (5|1), (6|0.75), (7|0.5), (8|0.25), (9|0), (10|0)\} \\
\text{big} &= \{(0|0), (1|0), (2|0), (3|0), (4|0), (5|0), (6|0.2), (7|0.4), (8|0.6), (9|0.8), (10|1)\}
\end{aligned}$$

To make these mathematical sentences more understandable, for example, one can take some values of x and, with respect to each membership function ("small," "medium," and "big"); they can be interpreted as follows:

$x = 0$	Belongs to "small" with a degree of membership equal to 1.0
$x = 2$	Belongs to "medium" with a degree of membership equal to 0 or does not belong to "medium"
$x = 7$	Belongs to "big" with a degree of membership equal to 0 or does not belong to "big"

In some cases, the grades of the membership $\mu_{\mathbf{A}}(x)$ can assume values from the set $\boldsymbol{M} = (-\infty, \infty)$ (or $[-1,1]$, for a normalized set). This generalization leads to a more general structure named L–fuzzy sets (2). The letter L comes from the word *lattice* (lattice theory).

As presented for ordinary sets, some properties and basic operations can also be defined for fuzzy sets. In fact, ordinary set theory is a subset of fuzzy set theory. The ordinary set theory uses a set $\boldsymbol{M}_1 = \{0,1\}$ and the fuzzy set theory uses, for example, a set $\boldsymbol{M}_2 = [0,1]$, where $\boldsymbol{M}_1$ is a subset of $\boldsymbol{M}_2$. Therefore, all operations and properties of ordinary set theory are valid for fuzzy set theory. In this way, according to this statement, if $\boldsymbol{A}$ and $\boldsymbol{B}$ are fuzzy subsets, all properties of Table 1 can be used in fuzzy set theory.

In addition, some operations can be redefined. Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be fuzzy subsets of a universe of discourse $\boldsymbol{U}$ and x an element of $\boldsymbol{U}$. Table 5 presents some of these basic operations.

For example, using the fuzzy subsets "small" and "medium," as defined above (shown in Fig. 1), to represent the values assumed by a certain fuzzy variable, some computations can be made using these subsets, according to the basic operations presented in Table 5. Table 6 shows these computations.

Basic Concepts of Fuzzy Statements

Data modeling can be defined as an attempt to represent, in a clear way, the available information from a set of data. The data modeling procedure is important because merging data in algebraic expressions allows the

Table 4. Basic Operations

Inclusion

$A \subset B$: $\mu_A(x) \leq \mu_B(x)$ for all x belonging to U

Meaning: A is an enclosure of B or B is an envelope of A

Equality

$A = B$ $\mu_A(x) = \mu_B(x)$ for all x belonging to U

Meaning: A is equal to B

Complement (with Values of $\mu_A(x)$ Given by the Set $M = [0, 1]$)

$\neg A$: $\mu_{\neg A}(x) = 1 - \mu_A(x)$ for all x belonging to U and the values of $\mu_A(x)$ given by the set $M = [0, 1]$

Meaning: $\neg \boldsymbol{A}$ is not $\boldsymbol{A}$

Intersection

$\boldsymbol{A} \cap \boldsymbol{B}$ $\mu_{\boldsymbol{A} \cap \boldsymbol{B}}(x) = \min(\mu_A(x), \mu_B(x))$ for all x belonging to U

Meaning: $\boldsymbol{A}$ and $\boldsymbol{B}$

Union

$\mathbf{A} \cup \mathbf{B}$: $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$ for all x belonging to U

Meaning: $\boldsymbol{A}$ or $\boldsymbol{B}$ (meaning, or $\boldsymbol{A}$ or $\boldsymbol{B}$ or both)

Disjunctive Sum

$\boldsymbol{A} \oplus \boldsymbol{B} = (\boldsymbol{A} \cap \neg \boldsymbol{B}) \cup (\neg \boldsymbol{A} \cap \boldsymbol{B})$

Meaning: $\boldsymbol{A}$ or (exclusive) $\boldsymbol{B}$ (meaning, or $\boldsymbol{A}$ or $\boldsymbol{B}$ but not both)

Difference

$\boldsymbol{A} - \boldsymbol{B} = \boldsymbol{A} \cap \neg \boldsymbol{B}$

Meaning: $\boldsymbol{A}$ minus $\boldsymbol{B}$

Algebraic Product

$\boldsymbol{A} \bullet \boldsymbol{B}$: $\mu_{\boldsymbol{A} \bullet \boldsymbol{B}}(x) = \mu_{\boldsymbol{A}}(x) \cdot \mu_{\boldsymbol{B}}(x)$ for all x belonging to $\boldsymbol{U}$

Meaning: $\boldsymbol{A}$ and probabilistic $\boldsymbol{B}$

Algebraic Sum

$\boldsymbol{A} \otimes \boldsymbol{B}$: $\mu_{A \otimes B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$ for all x belonging to U

Meaning: A or-probabilistic B

Table 5: Numerical Examples for Basic Operations

Complement

$$\text{not small} = \neg\text{small} : \mu_{\neg\text{small}}(X) = 1 - \mu_{\text{small}}(X)$$

not small = {(0|0), (1|0.2), (2|0.4), (3|0.6), (4|0.8), (5|1), (6|1), (7|1), (8|1), (9|1), (10|1)}

Intersection

$$\text{small and medium} = \text{small} \cap \text{medium} : \mu_{\text{small}\cap\text{medium}}(X) = MIN(\mu_{\text{small}}(X), \mu_{\text{medium}}(X))$$

small and medium = {(0|0), (1|0), (2|0.25), (3|0.4), (4|0.2), (5|0), (6|0), (7|0), (8|0), (9|0), (10|0)}

Union

$$\text{small or medium} = \text{small} \cup \text{medium} : \mu_{\text{small}\cup\text{medium}}(X) = MAX(\mu_{\text{small}}(X), \mu_{\text{medium}}(X))$$

small or medium = {(0|1), (1|0.8), (2|0.6), (3|0.5), (4|0.75), (5|1), (6|0.75), (7|0.5), (8|0.25), (9|0), (10|0)}

Disjunctive Sum

$$\text{small or (exclusive) medium} = \text{small} \oplus \text{medium} = (\text{small} \cup \neg\text{medium}) \cup (\neg\text{small} \cap \text{medium})$$

$$\text{small} \cap \neg\text{medium} = \{(0|1), (1|0.8), (2|0.6), (3|0.4), (4|0.2), (5|0), (6|0), (7|0), (8|0), (9|0), (10|0)\}$$

$$\neg\text{small} \cap \text{medium} = \{(0|0), (1|0), (2|0.25), (3|0.5), (4|0.75), (5|1), (6|0.75), (7|0.5), (8|0.25), (9|0), (10|0)\}$$

small or (exclusive) medium = {(0|1), (1|0.8), (2|0.6), (3|0.5), (4|0.75), (5|1), (6|0.75), (7|0.5), (8|0.25), (9|0), (10|0)}

Difference

$$\text{small} - \text{medium} = \text{small} \cap \text{not medium}$$

small − medium = {(0|1), (1|0.8), (2|0.6), (3|0.4), (4|0.2), (5|0), (6|0), (7|0), (8|0), (9|0), (10|0)}

Algebraic Product

$$\text{small} \bullet \text{medium} : \mu_{\text{small}\bullet\text{medium}}(X) = \mu_{\text{small}}(X) \cdot \text{medium}(X)$$

small • medium = {(0|0), (1|0), (2|0.15), (3|0.2), (4|0.15), (5|0), (6|0), (7|0), (8|0), (9|0), (10|0)}

Algebraic Sum

$$\text{small} \otimes \text{medium} : \mu_{\text{small}\otimes\text{medium}}(X) = \mu_{\text{small}}(X) + \mu_{\text{medium}}(X) - \mu_{\text{small}}(X) \cdot \mu\text{medium}(X)$$

small ⊗ medium = {(0|1), (1|0.8), (2|0.7), (3|0.7), (4|0.8), (5|1), (6|0.75), (7|0.5), (8|0.25), (9|0), (10|0)}

user to better visualize, understand, and interpret its structure. Moreover, algebraic expressions can be easily handled and incorporated into computer programs.

Fuzzy Statements. As seen before, the degree of membership of an element x of a fuzzy subset $\boldsymbol{A}$ can be denoted by $\mu_A(x)$. To express specific knowledge, input variables can be combined through linguistic conjunctions such as AND and OR, as shown in Table 5. Each linguistic conjunction has a meaning in the fuzzy logic theory and represents a specific operation (AND $\leftrightarrow$ minimum and OR $\leftrightarrow$ maximum). Also, in some cases, the complement of a membership can represent its negation; for example, the complement of $\boldsymbol{A}$ is $\not\boldsymbol{A}$, which means "not $\boldsymbol{A}$." In addition, these concepts allow the use of adverbs to modify (increase or decrease) the sharpness of a linguistic value, such as "very," "quite," and "about." Specific mathematical operations can be related to each adverb according to the desired effect in the shape of the fuzzy subset or linguistic value of the fuzzy variable.

According to these considerations, a fuzzy statement can be defined as an attribution of a fuzzy value to a fuzzy variable. This fuzzy value can be a single value (with or without adverbs) or a composed value (with two or more values that are combined by conjunctions). The general form of a fuzzy statement can be written as

$$X_i \text{ is } \boldsymbol{A}_i \tag{6}$$

where A_i represents the fuzzy value of the fuzzy variable x_i.

A fuzzy statement is a concept possible to be identified in some very simple examples taken from day-to-day life. For example, fuzzy statements can be "Mary is small," "John is tall," "the temperature is hot," and "the value of x is big or not very small," where "Mary," "John," "temperature," and "value of x" are the fuzzy variables of the statements, and "small," "tall," "hot," and "big or not very small" are their fuzzy values, respectively.

Fuzzy Conditional Statements. A mathematical equation represents a mapping between the input and output variable (or variables), and can be represented as a conditional statement in *if—then* form. Several kinds of mapping, such as artificial neural network techniques or linear equations, are described in the literature; they represent a way to manipulate the relations between input and output variable with advantages and drawbacks. Here will be proposed a mapping using fuzzy conditional statements in *if—then* format.

To build a fuzzy conditional statement, an *if—then* rule must have its premise and/or consequence represented by fuzzy statements. A structure based in conditional statement can be interpreted as a decision system: *if* the premise happens *then* the consequence will happen. A typical structure for a decision system has, normally, multiple input and multiple output (*MIMO*) variables. Let us consider a system with p input variables x, in the premise of the rules, and m output variables y, in the consequence. Thus, the general form of a fuzzy conditional statement is as follows:

$$\text{If}\{x_1 \text{ is } \boldsymbol{A}_1, \ldots, x_p \text{ is } \boldsymbol{A}_j\} \textit{ then } \{y_1 \text{ is } \boldsymbol{B}_1, \ldots, y_m \text{ is } \boldsymbol{B}_k\} \tag{7}$$

where x_1 to x_p are the input fuzzy variables, y_1 to y_m are the output fuzzy variables, and $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are the fuzzy values represented by the fuzzy subsets. The premise and the consequence of the fuzzy conditional statements are combined using a comma to denote the conjunction AND.

If the decision system has just one output [multiple input–single output (*MISO*)], the structure for fuzzy conditional statements has the general form given by

$$\text{If}\{x_1 \text{ is } \boldsymbol{A}_1, \ldots, x_p \text{ is } \boldsymbol{A}_j\} \textit{ then } \{y_1 \text{ is } \boldsymbol{B}_1\} \tag{8}$$

Ordinary and Fuzzy Relations

An ordinary relation can be defined as a set of degree of membership of each n-tuple from a set $\boldsymbol{U}_1 \times \boldsymbol{U}_2 \times \cdots \times \boldsymbol{U}_n$ (Cartesian product). For example, let $\boldsymbol{U}_1 = \{a,b,c\}$ and $\boldsymbol{U}_2 = \{d,e,f,g\}$, and $\boldsymbol{M} = [0,1]$. Figure 2 presents a sample of a relation $\Re$, while the following equation shows the numerical values:

$$\begin{aligned} \Re(\boldsymbol{U}_1 \times \boldsymbol{U}_2) \quad = \quad & \{((a,d)|1), ((a,e)|0), ((a,f)|1), ((a,g)|0), ((b,d)|1), \\ & ((b,e)|1), ((b,f)|1), ((b,g)|0), ((c,d)|0), ((c,e)|0), \\ & ((c,f)|1), ((c,g)|1)\} \end{aligned} \tag{9}$$

For a fuzzy relation, the definition follows the same structure of ordinary relations and the fuzzy set concepts. With two sets $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ and with x being an element of $\boldsymbol{U}_1$ and y an element of $\boldsymbol{U}_2$, for each element of the set of the ordered pairs (x,y), defined by the Cartesian product $\boldsymbol{U}_1 \times \boldsymbol{U}_2$, there is an associated degree of membership taken in a set $\boldsymbol{M} = [0,1]$. For example, Fig. 3 presents a sample of a relation $\Re$, while Eq. (10) shows the numerical values.

$$\begin{aligned} \Re(\boldsymbol{U}_1 \times \boldsymbol{U}_2) \quad = \quad & \{((a,d)|1), ((a,e)|0), ((a,f)|0.7), ((a,g)|0.2), \\ & ((b,d)|1), ((b,e)|0.8), ((b,f)|1), ((b,g)|0), \\ & ((c,d)|0.3), ((c,e)|0), ((c,f)|1), ((c,g)|0.8)\} \end{aligned} \tag{10}$$

In addition, some operations can be redefined. Let $\Re$ and $\Im$ be fuzzy relations defined in $\boldsymbol{U}_1 \times \boldsymbol{U}_2$, and (x,y) an ordered pair of $\boldsymbol{U}_1 \times \boldsymbol{U}_2$. Table 7 presents some of these basic operations using fuzzy relations.

Composition of Two Relations

MinMax Composition between two Relations. Consider two fuzzy (or ordinary) relations $\Re$ and $\Im$ defined in the following Cartesian products $\boldsymbol{X} \times \boldsymbol{Y}$ and $\boldsymbol{Y} \times \boldsymbol{Z}$, respectively. There are many ways to compute another relation $\aleph$, representing the Cartesian product $\boldsymbol{X} \times \boldsymbol{Z}$ based on $\Re$ and $\Im$. Minmax composition (or maxmin composition) is one of these ways.

Using the relations $\Re$ and $\Im$ defined above, the value of each element of $\aleph$ can be computed with

$$\aleph = \Re \circ \Im: \quad \mu_{\mathcal{R}\circ\mathcal{T}}(x,z) = \max\left(\min(\mu_{\Re}(x,y), \mu_{\Im}(y,z))\right) \tag{11}$$

for all (x,y) belonging to $\boldsymbol{X} \times \boldsymbol{Y}$ and for all (y,z) belonging to $\boldsymbol{Y} \times \boldsymbol{Z}$

Table 6. Basic Operations

Inclusion

$\mathfrak{R} \subset \mathfrak{S}$: $\mu_{\mathfrak{R}}(x, y) \leq \mu_{\mathfrak{S}}(x, y)$ for all (x, y) belonging to $\boldsymbol{U}_1 \times \boldsymbol{U}_2$

Meaning: $\mathfrak{R}$ is an enclosure of $\mathfrak{S}$ or $\mathfrak{S}$ is an envelope of $\mathfrak{R}$

Equality

$\mathfrak{R} = \mathfrak{R}$: $\mu_{\mathfrak{R}}(x, y) = \mu_{\mathfrak{S}}(x, y)$ for all (x, y) belonging to $\boldsymbol{U}_1 \times \boldsymbol{U}_2$

Meaning: $\mathfrak{R}$ is equal to $\mathfrak{S}$

Complementation

$\neg\mathfrak{R}$: $\mu_{\neg}\mathfrak{R}(x, y) = 1 - \mu_{\mathfrak{R}}(x, y)$ for all (x, y) belonging to $\boldsymbol{U}_1 \times \boldsymbol{U}_2$ and $\boldsymbol{M} = [0, 1]$

Meaning: $\neg\mathfrak{R}$ is not $\mathfrak{R}$

Intersection

$\mathfrak{R} \cap \mathfrak{S}$: $\mu_{\mathfrak{R}\cap\mathfrak{S}}(x, y) = \min(\mu_{\mathfrak{R}}(x, y), \mu_{\mathfrak{S}}(x, y))$ for all (x, y) belonging to $\boldsymbol{U}_1 \times \boldsymbol{U}_2$,

Meaning: $\mathfrak{R}$ and $\mathfrak{S}$

Union

$\mathfrak{R} \cup \mathfrak{S}$: $\mu_{\mathfrak{R}\cup\mathfrak{S}}(x, y) = \max(\mu_{\mathfrak{R}}(x, y), \mu_{\mathfrak{S}}(x, y))$ for all (x, y) belonging to $\boldsymbol{U}_1 \times \boldsymbol{U}_2$,

Meaning: $\mathfrak{R}$ or $\mathfrak{S}$

Disjunctive Sum

$\mathfrak{R} \oplus \mathfrak{S} = (\mathfrak{R} \cap \neg\mathfrak{S}) \cup (\neg\mathfrak{R} \cap \mathfrak{S})$

Meaning: $\mathfrak{R}$ or (exclusive) $\mathfrak{S}$

Difference

$\mathfrak{R} - \mathfrak{S} = \mathfrak{R} \cap \neg\mathfrak{S}$

Meaning: $\mathfrak{R}$ minus $\mathfrak{S}$

Algebraic product

$\mathfrak{R} \bullet \mathfrak{S}$: $\mu_{\mathfrak{R}\bullet\mathfrak{S}}(x, y) = \mu_{\mathfrak{R}}(x, y) \cdot \mu(x, y)$ for all x belonging to $\boldsymbol{U}_1 \times \boldsymbol{U}_2$,

Meaning: $\mathfrak{R}$ and probabilistic $\mathfrak{S}$

Algebraic sum

$\mathfrak{R} \otimes \mathfrak{S}$: $\mu_{\mathfrak{R}\otimes\mathfrak{S}}(x, y) = \mu_{\mathfrak{R}}(x, y) + \mu_{\mathfrak{S}}(x, y) - \mu_{\mathfrak{R}}(x, y) \cdot \mu_{\mathfrak{S}}(x, y)$ for all (x, y) belonging to $\boldsymbol{U}_1 \times \boldsymbol{U}_2$,

Meaning: $\mathfrak{R}$ or-probabilistic $\mathfrak{S}$

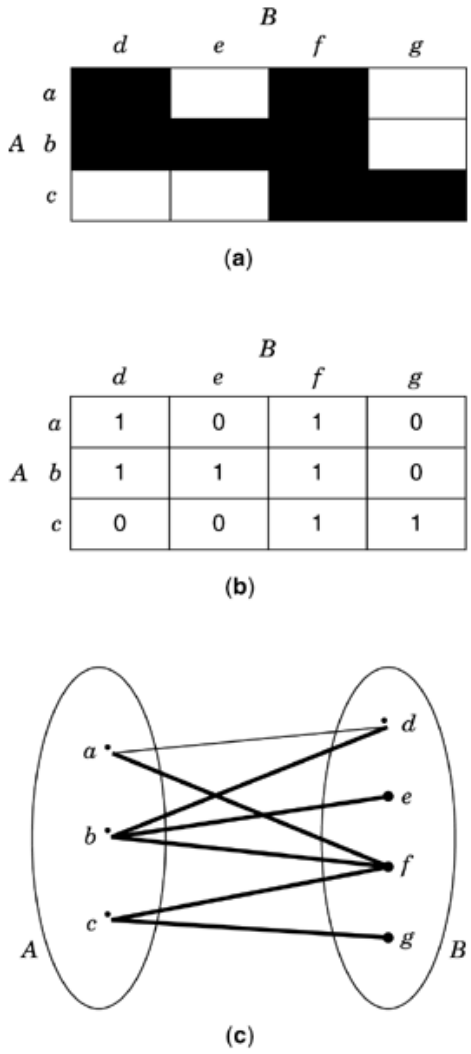


Fig. 2. Relation representations: (a) table, (b) numerical, and (c) graphs.

Numerical example. Let $\Re$ and $\Im$ be the fuzzy relations defined below.

$\Re$	y_1	y_2	y_3	y_4	y_5
x_1	0.1	0.3	0.5	0.7	1
x_2	0.5	0.8	1	1	0.8
x_3	1	0.6	0.2	0	0.2

$\Im$	z_1	z_2	z_3	z_4
y_1	0.2	0.4	0	1
y_2	0.7	0.9	1	0.9
y_3	1	0.4	0.2	0.5
y_4	0.3	0.1	0.3	0.3
y_5	0	0.1	0.2	0.1

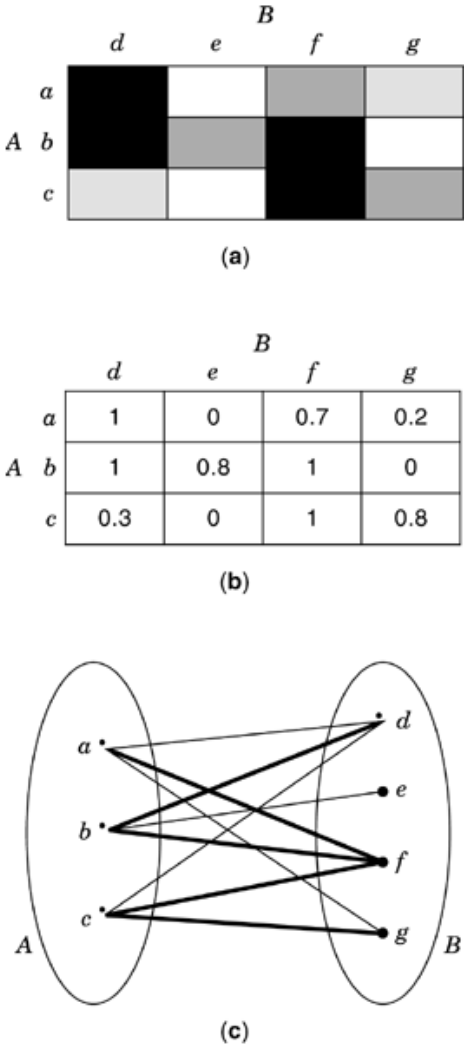


Fig. 3. Relation representations: (a) table, (b) numerical, and (c) graphs.

Computing the value of $\aleph(x_1,z_1)$:

$$\begin{aligned}
\aleph(x_1, z_1) &= \max\{\min(\mu_{\Re}(x_1, y_1), \mu_{\Im}(y_1, z_1)), \min(\mu_{\Re}(x_1, y_2), \mu_{\Im}(y_2, z_1)), \\
&\quad \min(\mu_{\Re}(x_1, y_3), \mu_{\Im}(y_3, z_1)), \min(\mu_{\Re}(x_1, y_4), \mu_{\Im}(y_4, z_1))\} \\
&= \max\{\min(0.1, 0.2), \min(0.3, 0.7), \min(0.5, 1), \\
&\quad \min(0.7, 0.3), \min(1, 0)\} \\
&= \max\{0.1, 0.3, 0.5, 0.3, 0\} = 0.5
\end{aligned}$$

Continuing this computation for other elements of $\aleph$, the relation obtained is

$\aleph$	z_1	z_2	z_3	z_4
x_1	0.5	0.4	0.3	0.5
x_2	1	0.4	0.8	0.8
x_3	0.6	0.6	0.6	1

Operations with MinMax Composition and Other Compositions. The minmax composition operation is associative and distributive with respect to union but not with respect to intersection.

$$(\Re \circ \Im) \circ \aleph = \Re \circ (\Im \circ \aleph) \tag{12}$$

$$\Re \circ (\Im \cup \aleph) = (\Re \circ \Im) \cup (\Re \circ \aleph) \tag{13}$$

$$\Re \circ (\Im \cap \aleph) \neq (\Re \circ \Im) \cap (\Re \circ \aleph) \tag{14}$$

There are many other possible compositions like max-product, max-times, and min-product used in some specific conditions. The most natural composition is the minmax, because it is very similar to the matrix product. In this product, there is an algebraic product of each pair and then an algebraic sum of the results. Observing the minmax composition, there is a min composition (in ordinary relations, equal to a Boolean product) of each pair and then a max composition (in ordinary relations, equal to a Boolean sum) of the results.

Relation as Mapping. A relation can also be used as mapping between two worlds, for example $\boldsymbol{X}$ and $\boldsymbol{Y}$. In other words, a fuzzy set $\boldsymbol{A}$ in the first world $\boldsymbol{X}$ has an image (a fuzzy set) $\boldsymbol{B}$ in the world $\boldsymbol{Y}$. Also, there are many ways to compute this image. MinMax is the most frequently used composition.

This mapping can be expressed by the following equation, in which there is a relation (mapping) $\Re$, of a fuzzy set $\boldsymbol{A}$, in $\boldsymbol{X}$ (domain), and a fuzzy set $\boldsymbol{B}$, in $\boldsymbol{Y}$ (range).

$$\mu_{\mathrm{B}}(y) = \max\left(\min(\mu_{\Re}(x, y), \mu_{\mathrm{A}}(x)\right) \quad \begin{array}{l}\text{for all}(x, y)\text{belonging to } \boldsymbol{X} \times \boldsymbol{Y}, \\ \text{for all}(x)\text{belonging to } \boldsymbol{X} \\ \text{for all}(y)\text{belonging to } \boldsymbol{Y}\end{array} \tag{15}$$

An example of this computation is carried out below, using the relation $\Re$ defined above and the fuzzy set $\boldsymbol{A}$.

$$\boldsymbol{A} = \{(x_1|1), (x_2|0.8), (x_3|0.6)\}$$

$\Re$	y_1	y_2	y_3	y_4	y_5
x_1	0.1	0.3	0.5	0.7	1
x_2	0.5	0.8	1	1	0.8
x_3	1	0.6	0.2	0	0.2

Computing the value of $\boldsymbol{B}(y_1)$:

$$\begin{aligned}\boldsymbol{B}(y_1) &= \max\{\min(\mu_{\Re}(x_1, y_1), \mu_{\mathrm{A}}(x_1)), \min(\mu_{\Re}(x_2, y_1), \\ &\quad \mu_{\Im}(x_2)), \min(\mu_{\Re}(x_3, y_1), \mu_{\Im}(x_3))\} \\ &= \max\{\min(1, 0.1), \min(0.8, 0.5), \min(0.6, 1)\} \\ &= \max\{0.1, 0.3, 0.6\} = 0.6\end{aligned}$$

Continuing this computation for other elements of $\boldsymbol{B}$, the fuzzy set obtained is

$$\boldsymbol{B} = \{(y_1|0.6), (y_2|0.8), (y_3|0.8), (y_4|0.8)), (y_5|1)\}$$

An Illustrative Example

The Problem. In order to have a more accurate view of the application of MinMax in the fuzzy technique, let us propose a simple and comprehensive control problem. The problem refers to the action of stepping on the brakes of a car in order to stop the car when the driver sees a red light. It is easy to notice that the inputs evaluated by the driver are the speed of the car ($\boldsymbol{v}$) and the distance between the car and the traffic light ($\boldsymbol{d}$).

The idea behind the application of the fuzzy technique is to replace the driver in this action. The human strategy is replaced by a decision-making algorithm described in terms of a set of fuzzy conditional statements.

The analysis of the driver algorithm. When the driver sees the red light, he or she evaluates (measures) the car's speed and the distance to the stop point, and according to the obtained information, he or she decides on the amount of force that has to be exerted on the brake pedal. To translate into words the strategy used by the driver, let us imagine two common situations. The first is when the driver sees the red light and the stopping distance ($\boldsymbol{d}$) is short and the speed of the car ($\boldsymbol{v}$) is high. In this situation, the force applied to the brakes has to be high to avoid disaster. The second situation is when the stopping distance is very short and the speed of the car is very close to zero. Now, the force applied to the brakes can be very small.

The control system takes the input variables, which are numerical values measured directly from the process, and manipulates them using a control algorithm (in this case, the fuzzy decision-making algorithm). The control algorithm generates the output that allows the control process to achieve the desired behavior. These values are not fuzzy, but crisp values. They are presented to the system as continuous values read and manipulated (computed); thus, the algorithm, representing the driver, continuously generates output values to guarantee the best performance of the entire process.

To use a fuzzy technique scheme, the crisp values obtained from the process have to be transformed into fuzzy values; this step is called the *fuzzification* process. The *fuzzification* process is the first step before running the set of conditional fuzzy statements or set of fuzzy rules, which is done after reading the numerical values of variables. The objective of *fuzzification* is to transform the "actual values" of the variables into "fuzzy

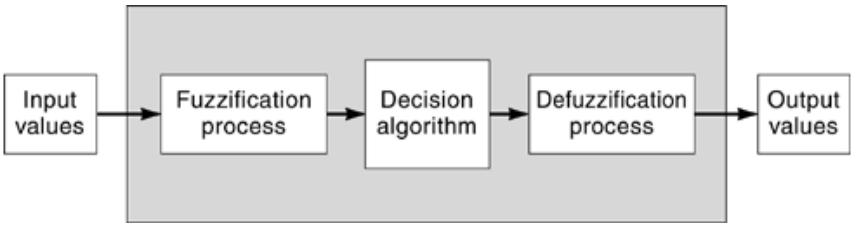


Fig. 4. Scheme for application of the fuzzy technique.

values" (linguistic values), which are then manipulated by the decision algorithm. After running the decision algorithm, the calculated output value is also a fuzzy value that cannot be used as an input for a real process. The *defuzzification* process is the last step of the fuzzy technique in order to transform fuzzy values back to actual values.

Figure 4 presents the general scheme of a decision process using fuzzy technique. Let us define the values that the input fuzzy variables will assume. The fuzzy variable distance from the car to the stopping point ($\boldsymbol{d}$) will assume three fuzzy values: small, medium, and big. The other input fuzzy variable, the speed of the car ($\boldsymbol{v}$), will assume also three fuzzy values: low, medium, and high. And finally, for the output fuzzy variable, force applied to the brake pedal ($\boldsymbol{f}$), also three values will be assigned: low, medium, and high.

Figure 5 illustrates graphically the values assumed for each variable of the process, described by their fuzzy subsets. It is important to note that the fuzzy value medium or high for one variable does not necessarily have to coincide with other variable; in general they are really different subsets.

An example of the fuzzification process is shown in Figure 5. Considering only the first rule depicted in Figure 5, the input $\boldsymbol{d} = 10$ m generates a value of memberships for the subset "small" depending on the premise of the fuzzy rule equal to 0.6 [$\mu_{\text{small}}(\boldsymbol{d}) = \mu_{\text{small}}(10) = 0.6$], and the input $\boldsymbol{v} = 8$ m/s generates a value of membership for the subset "medium," equal to 0.5 [$\mu_{\text{medium}}(\boldsymbol{v}) = \mu_{\text{medium}}(8) = 0.5$]. An example of the *defuzzification* process is presented next.

Fuzzy Inference Process. A fuzzy inference process is the way the set of fuzzy conditional statements (or fuzzy rules) are executed to obtain a meaningful inference in the output. As mentioned before, the complete decision algorithm is composed of a set of fuzzy rules. The algorithm first transforms the input variables into fuzzy statements, and then computes the output value. The execution of each rule is done using *modus ponens*, which means that the premise of each rule produces a degree of membership for a certain value of input variables. This degree of membership is a function of the *fuzzified* values (*fuzzification* procedure) of the input variables and of the conjunctions used among them for each fuzzy rule. Let's consider the example presented in Fig. 5, where the two arbitrary fuzzy conditional statements or fuzzy rules are shown:

Rule i *If* $\boldsymbol{d}$ is small **and** $\boldsymbol{v}$ is medium, *then* $\boldsymbol{f}$ is medium.
Rule j *If* $\boldsymbol{d}$ is small **and** $\boldsymbol{v}$ is high, *then* $\boldsymbol{f}$ is high

Notice that during the *fuzzification* process for rule i, the values obtained for the membership functions are 0.6 and 0.5, as mentioned above. Since the liaison element used between the fuzzy statements is the conjunction "and" (which represents the "minimum" operator), the conclusion of the rule (value "medium") has its value of membership limited by the minimum value of the premise, that is, $\min(0.6, 0.5) = 0.5$. This concept is depicted as a shaded area (S_i) in Fig. 5.

Each rule produces a limited area (as illustrated by the shaded area in Fig. 5) according to the value produced by the premise and the output fuzzy value of the conclusion represented by its membership function.

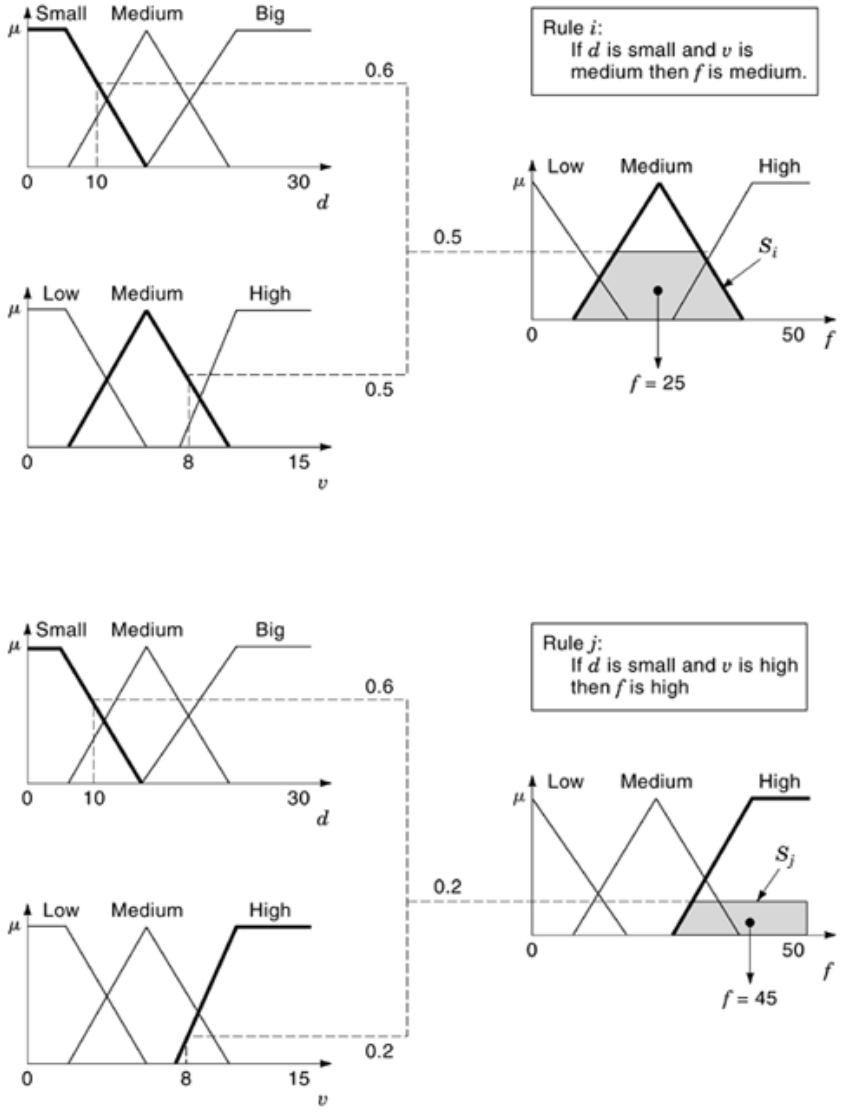


Fig. 5. Fuzzification, inference and defuzzification processes.

In the same way, for rule j, the limitation of the fuzzy value "high" is 0.2. If the value of membership obtained by the premise is zero, it means the rule has no influence on the computation of the final output value.

After the execution of all the rules, the *defuzzification* process begins. The shaded area for each rule is computed; then the maximum operation is applied to define the largest one. In Fig. 5, the shaded area S_i is bigger than S_j, that is, $\max(S_i, S_j) = S_i$. So the area S_i is taken to calculate the final actual output value. This value is computed using the center of gravity method (centroid) for the geometrical figure represented by the shaded area S_i. The value of the abscisa found is the actual value of the output variable. Figure 5 shows an example of the process, where the centroid method produces a value of $\boldsymbol{f}$ equal to 25 for the largest shaded area

obtained (rule S_i). The following describes the centroid method used:

$$f_{(\text{actual output})} = \frac{\sum \mu_{\text{Medium}}(f)\, f}{\sum \mu_{\text{Medium}}(f)} \tag{16}$$

BIBLIOGRAPHY

1. L. A. Zadeh Fuzzy sets, *Information and Control*, **8**: 338–353, 1965.
2. A. Kaufmann *Introduction to the Theory of Fuzzy Subsets*, Vol. I, London: Academic 1957.

GERMANO LAMBERT-TORRES
Escola Federal de Engenharia de Itajubá
JOÃO ONOFRE PEREIRA PINTO
The University of Tennessee at Knoxville
LUIZ EDUARDO BORGES DA SILVA
Escola Federal de Engenharia de Itajubá

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Nomograms
Standard Article
Michael D. Harpen[1]
[1]University of South Alabama, Mobile, AL,
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2440
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (129K)

Abstract

The sections in this article are

Examples of Simple Nomograms

General Theory of Nomograms

Design Strategy

Keywords: nomograms; nomographs; alignment charts; quality control; resistance; reactive; power factor; determinant

| | |

Copyright © 1999-2008 All Rights Reserved.

NOMOGRAMS

A nomogram, nomograph, or alignment chart is a graphical display of the relationship between (usually) three variables which allows the value of one of the variables to be determined given the values of the other two variables.

EXAMPLES OF SIMPLE NOMOGRAMS

The underlying concepts of nomograms and the meanings of some fundamental terms can be illustrated by considering some of the basic simple nomogram types and the functional relations treated by them.

The Parallel Nomogram

A simple nomogram for performing the sum $\omega = \mu + u$ is given in Fig. 1. The calibrated axes represent values of the variables μ, u, and ω. A straight line, the *isopleth,* connects the point on the μ-axis corresponding to the value $\mu = 0.4$ with the point on the u-axis corresponding to the value $u =$

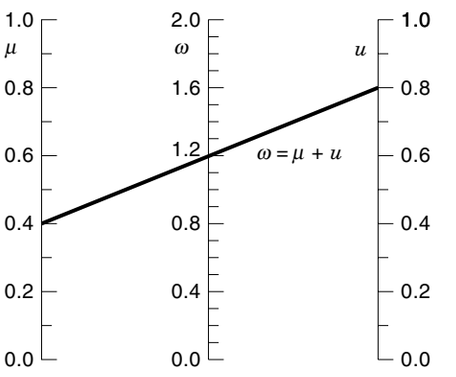


Figure 1. A simple parallel nomogram with linear axes representing the relation $\omega = \mu + u$.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

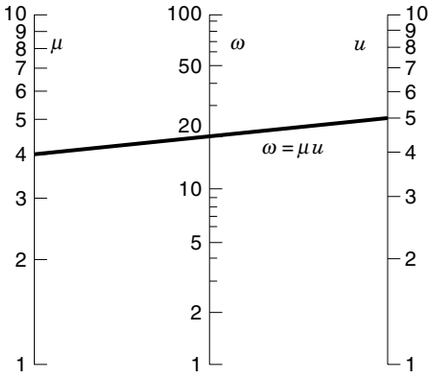


Figure 2. A parallel nomogram with logarithmic axes representing the relation $\omega = \mu u$.

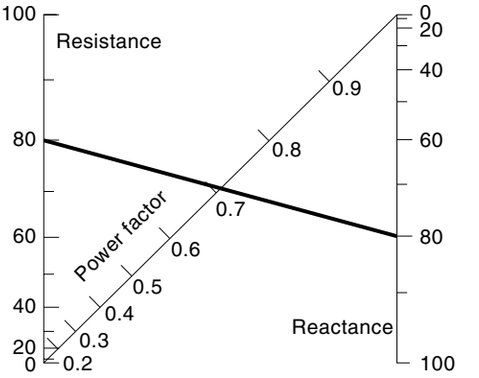


Figure 4. A Z nomogram with parabolic scales which can be used to calculate the power factor from the resistive and reactive components of load impedance.

0.8 and intercepts the ω-axis at the value $\omega = 1.2$, the sum of 0.4 and 0.8. In this manner the sum of any pair of μ and u up to a maximum value of 2 can be determined with the corresponding isopleth. The three axes are linear; i.e., changes in the values of a variable correspond to proportional changes in position along the respective axis, the constant of proportionality being the *scale* of the axis. The scale of the ω-axis in this case is half that of the μ- and u-axes. If we were to replace the ω-axis with one of equal scale to that of the μ- and u-axes the nomogram would yield the average of μ and u. By appropriately modifying the scales of the μ- and u-axes or the separation between the parallel axes a nomogram for calculating weighted averages of μ and u can be constructed. By replacing the linear scales in Fig. 1 with appropriate logarithmic scales we can obtain a nomogram which adds logarithms, i.e. which multiplies powers of μ and u (1). A simple nomogram with logarithmic scales for calculating $\omega = \mu u$ is illustrated in Fig. 2.

The Z Nomogram

A Z nomogram with linear scales is illustrated in Fig. 3. In this example the percent of students passing an examination (% pass) is calculated from the total number of students passing and the total number of students failing the examination. For example, 20 passing and 5 failing produces an 80% pass rate. This nomogram expresses the relation

$$\%\ \text{Passing} = 100\frac{\text{Passing}}{\text{Passing} + \text{Failing}} \tag{1}$$

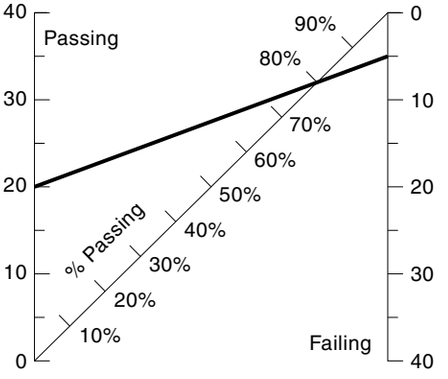


Figure 3. A Z nomogram with linear scales which can be used to calculate the percent pass rate from the total number passing and total number failing.

In general a Z nomogram with linear scales expresses a relation of the form

$$\omega = \frac{\mu}{\mu + u} \tag{2}$$

or any relation which can be cast into this form with a suitable transformation of variables and corresponding nonlinear scales. For example, if a load impedance has a resistive component of R and a reactive component of X, then the power factor can be expressed as

$$(\text{Powerfactor})^2 = \frac{R^2}{R^2 + X^2} \tag{3}$$

Comparing Eq. (3) with Eq. (2) we have the correspondence

$$\begin{aligned} (\text{Powerfactor})^2 &= \omega \\ R^2 &= \mu \\ X^2 &= u \end{aligned} \tag{4}$$

which defines the appropriate nonlinear scales. A nomogram constructed using Eq. (4) is given in Fig. 4.

The W Nomogram

A W chart nomogram with linear scales is illustrated in Fig. 5. In this example the nomogram determines the electrical resistance R resulting from two resistances R_1 and R_2 in parallel. The nomogram performs the calculation

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} \tag{5}$$

which illustrates the basic functional form treated by this type of nomogram.

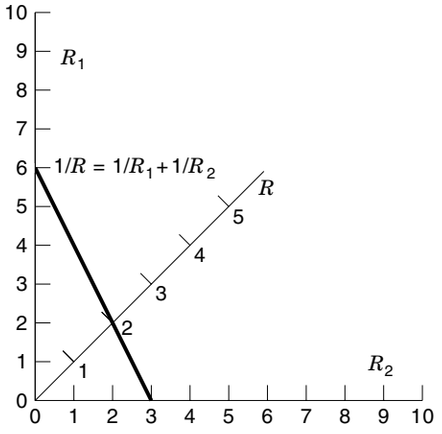


Figure 5. A W nomogram with linear axes for calculating the resultant resistance R of two resistances R_1 and R_2 in parallel.

The Circular Nomogram

A basic circular nomogram is illustrated in Fig. 6. The upper and lower semicircles are two axes with linear scales for the variables α and β, respectively, which for simplicity are also the angles between the respective radii and the x-axis. In this case the radius of the circle is one so that x runs from -1 to $+1$. The values of α, β, and x related by the isopleth satisfy

$$\frac{1-x}{1+x} = \tan\frac{\alpha}{2}\tan\frac{\beta}{2} \tag{6}$$

which illustrates the basic functional form treated by this type of nomogram. This nomogram is typically employed in quality assurance (2). Suppose a small sample of n light bulbs is selected at random from an assembly line and tested until the first failure is recorded at H hours of continuous use. The probability (W) that in a large sample of bulbs no more than $K\%$ of failures will be observed after H hours is given by

$$W = 1 - (1 - 0.01K\%)^n \tag{7}$$

This formula may be rewritten as

$$-\frac{\ln(1-W)}{3} = \frac{n}{30}(-10\ln(1-0.01K\%)) \tag{8}$$

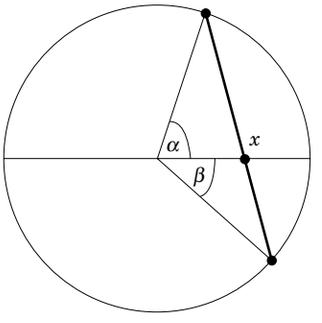


Figure 6. A circular nomogram which shows relation between angles (α, β) subtended by points on upper and lower semicircular axes to center, the x-intercept, and the isopleth.

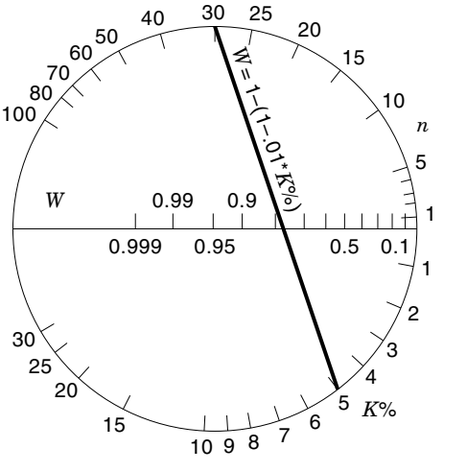


Figure 7. If in a sample of 30 light bulbs the first failure is observed after H hours, then there is a 0.79 probability that in a large sample of bulbs less than 5% will fail in H hours.

Comparing Eq. (8) with Eq. (6) we can make the correspondence

$$\begin{aligned} -\frac{\ln(1-W)}{3} &= \frac{1-x}{1+x} \\ \frac{n}{30} &= \tan\frac{\alpha}{2} \\ -10\ln(1-0.01K\%) &= \tan\frac{\beta}{2} \end{aligned} \tag{9}$$

Inverting these expressions we have

$$\begin{aligned} x &= \frac{2}{1-\dfrac{\ln(1-W)}{3}} - 1 \\ \alpha &= 2\tan^{-1}\frac{n}{30} \\ \beta &= 2\tan^{-1}(-10\ln(1-0.01K\%)) \end{aligned} \tag{10}$$

which can be used to calibrate the α-, β-, and x-axis in terms of n, $K\%$, and W respectively. The resulting nomogram is given in Fig. 7.

GENERAL THEORY OF NOMOGRAMS

We now present the detailed theory of the nomogram treating explicitly nomograms consisting of parallel scales as in Figs. 1 and 2 for expository convenience. Nomograms of this type represent functional relationships of the form

$$f_1(\mu) + f_2(u) = f_3(\omega) \tag{11}$$

We refer the axes of the nomograms to a Cartesian coordinate systems where the μ-, u-, and ω-axes are parallel to the y-axis at x values 0, b, and a, respectively, so that a given value of μ corresponds to the coordinates

$$(X_1, Y_1) = (0, m_\mu f_1(\mu)) \tag{12}$$

where m_μ is a scale factor which associates the value of the function $f_1(\mu)$ with a position on the μ-axis. Likewise, given values of u and ω will correspond to coordinates

$$(X_2, Y_2) = (b, m_u f_2(u)) \tag{13}$$

and

$$(X_3, Y_3) = (a, m_\omega f_3(\omega)) \tag{14}$$

respectively. When these three points are collinear, i.e. fall along an isopleth, their coordinates satisfy the homogeneous determinant relation

$$\begin{vmatrix} X_1 & Y_1 & 1 \\ X_2 & Y_2 & 1 \\ X_3 & Y_3 & 1 \end{vmatrix} = 0 \tag{15}$$

The Construction Determinant

Substituting Eqs. (12), (13), and (14) into Eq. (15) we have

$$\begin{vmatrix} 0 & m_\mu f_1(\mu) & 1 \\ b & m_u f_2(u) & 1 \\ a & m_\omega f_3(\omega) & 1 \end{vmatrix} = 0 \tag{16}$$

Eq. (16) is referred to as the construction determinant for the nomogram (3). Expanding the determinant and rearranging terms we have

$$\left(1 - \frac{a}{b}\right) \frac{m_\mu}{m_\omega} f_1(\mu) + \frac{a}{b} \frac{m_u}{m_\omega} f_2(u) = f_3(\omega) \tag{17}$$

Comparing Eq. (17) with Eq. (11) we have the correspondence

$$\left(1 - \frac{a}{b}\right) \frac{m_\mu}{m_\omega} = 1 \tag{18}$$

and

$$\frac{a}{b} \frac{m_u}{m_\omega} = 1 \tag{19}$$

which are the constraints between the scale factors of the axes and the distances between the axes. Suppose we wish to construct a nomogram to add two variables in quadrature, e.g.

$$\mu^2 + u^2 = \omega^2 \tag{20}$$

where μ runs from 0 to 4 and u from 0 to 5 and that the size of the nomogram is to be 5 centimeters high by 3 centimeters wide. The scale factors for the μ- and u-axes are determined by requiring the range of values to fit the 5-cm height of the nomogram:

$$m_\mu = \frac{5}{4^2} = \frac{5}{16}, \quad m_u = \frac{5}{5^2} = \frac{1}{5}$$

and the separation between the μ- and u-axes by the width of the nomogram:

$$b = 3$$

Dividing Eqs. (18) and (19) and substituting the values m_μ, m_u, and b, allow a to be determined as

$$a = \frac{b}{\dfrac{m_u}{m_\mu} + 1} = \frac{75}{41}$$

Finally, m_ω may be calculated as

$$m_\omega = \frac{a}{b} m_u = \frac{5}{41}$$

Thus, the calibrated scales of the nomogram are calculated as

$$\begin{aligned} X_1 &= 0 & Y_1 &= \frac{5}{16}\mu^2 \\ X_2 &= 3 & Y_2 &= \frac{1}{5}u^2 \\ X_3 &= \frac{75}{41} & Y_3 &= \frac{5}{41}\omega^2 \end{aligned} \tag{21}$$

The nomogram is given in Fig. 8.

Shifting Axis Origins

In this nomogram the origin of the axes correspond to the 0 values of the respective variables. In many situations it is preferable to have the axis origins corresponding to non-0 initial values; e.g., μ_0, u_0, and ω_0 which satisfy Eq. (11). In this case the construction determinant is modified as

$$\begin{vmatrix} 0 & m_\mu[f_1(\mu) - f_1(\mu_0)] & 1 \\ b & m_u[f_2(u) - f_2(u_0)] & 1 \\ a & m_\omega[f_3(\omega) - f_3(\omega_0)] & 1 \end{vmatrix} = 0 \tag{22}$$

and Eq. (18) and Eq. (19) are still valid. If for example $\mu_0 = 1$ and $u_0 = 2$ the scale factors and design parameters become

$$m_\mu = \frac{5}{5^2 - 1^2} \qquad m_u = \frac{5}{6^2 - 2^2} \qquad a = \frac{12}{7}$$

and

$$m_\omega = \frac{5}{56}$$

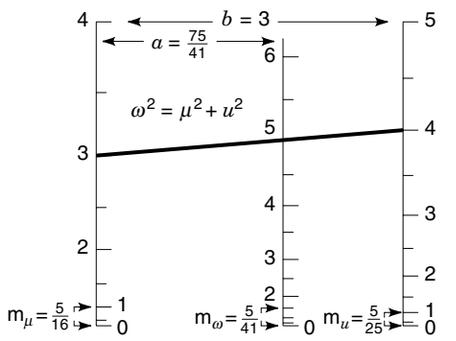


Figure 8. Parallel nomogram for adding variables in quadrature. The nomogram is 5 units high by 3 units wide and illustrates design parameters a, b, and m_μ, m_u, and m_ω.

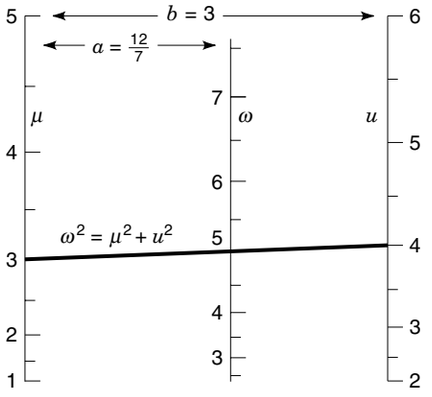


Figure 9. Nomogram as in Fig. 8 with shifted origins.

The calibrated scales for the nomogram are calculated as

$$\begin{aligned} X_1 &= 0 \qquad Y_1 = \frac{5}{24}(\mu^2 - 1) \\ X_2 &= 3 \qquad Y_2 = \frac{5}{32}(u^2 - 4) \\ X_3 &= \frac{12}{7} \quad Y_3 = \frac{5}{56}(\omega^2 - 5) \end{aligned} \tag{23}$$

This nomogram is given in Fig. 9.

Other Common Nomograms

We now apply the construction determinant methodology to the other nomogram types treated thus far, justifying the functional forms consistent with three nomograms. As stated earlier the Z chart nomogram can be used to represent the functional relationship from

$$f_3(\omega) = \frac{f_1(\mu)}{f_1(\mu) + f_2(u)} \tag{24}$$

Referring to Fig. 10 we can assign the following coordinates to the three points on the isopleth

$$\begin{aligned} X_1 &= 0 & Y_1 &= m_\mu f_1(\mu) \\ X_2 &= a & Y_2 &= Y_{\max} - m_u f_2(u) \\ X_3 &= \frac{a}{[a^2 + Y_{\max}^2]^{1/2}} m_\omega f_3(\omega) & Y_3 &= \frac{Y_{\max}}{[a^2 + Y_{\max}^2]^{1/2}} m_\omega f_3(\omega) \end{aligned} \tag{25}$$

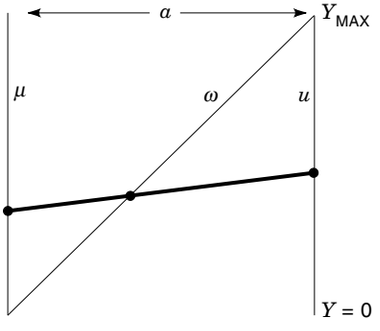


Figure 10. Prototype of Z nomogram illustrating parameters a and Y_{max}.

Where a is the separation between the parallel axes which are both of length Y_{max}. Evaluating the construction determinant with the substitutions of Eqs. (25) leads to

$$f_3(\omega) = [a^2 + Y_{\max}^2]^{1/2} \frac{m_\mu}{m_\omega} \frac{f_1(\mu)}{m_\mu f_1(\mu) + m_u f_2(u)} \tag{26}$$

which is equivalent to Eq. (24) when

$$\frac{m_\omega}{[a^2 + Y_{\max}^2]^{1/2}} = 1 \tag{27}$$

and

$$m_\mu = m_u \tag{28}$$

Equation (27) allows the scale factor m_ω to be determined from the dimensions of the nomogram. Scale factors m_μ and m_u are equal and determined by the length of the nomogram and the range of values of μ and u as in the parallel nomogram treated previously.

The W chart nomogram can be used to represent functional relationship of the form

$$\frac{1}{f_3(\omega)} = \frac{1}{f_1(\mu)} + \frac{1}{f_2(u)} \tag{29}$$

Referring to Fig. 11 we can make the following assignments for the coordinates of the three points on the isopleth

$$\begin{aligned} X_1 &= m_\mu (f_1(\mu) & Y_1 &= 0 \\ X_2 &= \cos\beta\, m_u f_2(u) & Y_2 &= \sin\beta\, m_u f_2(u) \\ X_3 &= \cos\alpha\, m_\omega f_3(\omega) & Y_3 &= \sin\alpha\, m_\omega f_3(\omega) \end{aligned} \tag{30}$$

Evaluating the construction determinant with these substitutions leads to

$$\frac{1}{f_3(\omega)} = \frac{m_\omega}{m_\mu} \frac{\sin(\beta - \alpha)}{\sin\beta} \frac{1}{f_1(\mu)} + \frac{m_\omega}{m_u} \frac{\sin\alpha}{\sin\beta} \frac{1}{f_2(u)} \tag{31}$$

Comparing Eq. (31) with Eq. (29) we have

$$\frac{m_\omega}{m_\mu} \frac{\sin(\beta - \alpha)}{\sin\beta} = 1 \tag{32}$$

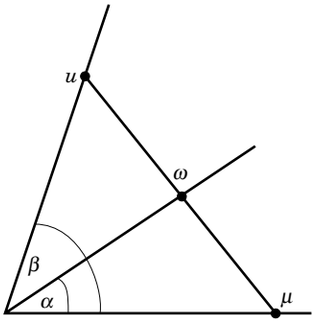


Figure 11. Prototype of W nomogram illustrating parameters α and β.

and

$$\frac{m_\omega}{m_u}\frac{\sin\alpha}{\sin\beta} = 1 \tag{33}$$

The scale factors and angles are determined by the range of the variables, the size of the nomogram, and the constraint of Eq. (32) and Eq. (33). For example, in Fig. 5 the R_1- and R_2-axes (i.e., the μ- and u-axes) have equal scale (e.g. $m_\mu = m_u = 1$) and $\beta = 90°$. Dividing Eq. (32) by Eq. (33) gives

$$\frac{\sin(\beta-\alpha)}{\sin(\alpha)} = 1 \tag{34}$$

which requires $\alpha = 45°$. Substituting into Eq. (33) we have $m_\mu = \sqrt{2}$.

In the simple circular nomogram given in Fig. 6 the X and Y coordinates of the three points of the isopleth are

$$\begin{aligned} X_1 &= \cos\alpha \quad & Y_1 &= \sin\alpha \\ X_2 &= \cos\beta \quad & Y_2 &= -\sin\beta \\ X_3 &= x \quad & Y_3 &= 0 \end{aligned} \tag{35}$$

Substituting these into Eq. (15) and rearranging terms we arrive at

$$\frac{1-x}{1+x} = \tan\frac{\alpha}{2}\tan\frac{\beta}{2} \tag{36}$$

Functional relationships of the form

$$f_3(\omega) = f_1(\mu)f_2(u) \tag{37}$$

may be represented by circular nomograms using the scale transformation

$$\begin{aligned} \alpha &= 2\tan^{-1}(m_\mu f_1(\mu)) \\ \beta &= 2\tan^{-1}(m_u f_2(u)) \\ x &= \frac{2}{m_\mu m_u f_3(\omega)+1} - 1 \end{aligned} \tag{38}$$

DESIGN STRATEGY

Up to this point the construction determinant has been used to justify the functional relation consistent with a given nomogram topology (i.e. parallel, Z, W, circular). We now illustrate the strategy for using the construction determinant to determine the nomogram topology given a nonstandard functional relation, for example

$$f_1(\mu) + f_2(u)f_3(\omega) - f_3^2(\omega) = 0 \tag{39}$$

The first step is to find a 3×3 matrix whose determinant is the left hand side of Eq. (39). This can be done by introducing the dummy variables

$$x = f_1 \qquad y = f_2 \qquad z = 1$$

so that Eq. (39) can be thought of as the result of eliminating these variables from the three equations

$$\begin{aligned} x - f_1 z &= 0 \\ y - f_2 z &= 0 \\ x + f_3 y - f_3^2 z &= 0 \end{aligned} \tag{40}$$

Equation (40) can be written in matrix form as

$$\begin{bmatrix} 1 & 0-f_1 \\ 0 & 1-f_2 \\ 1 & f_3 - f_3^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0 \tag{41}$$

The consistency of Eq. (40) requires

$$\det \begin{bmatrix} 1 & 0-f_1 \\ 0 & 1-f_2 \\ 1 & f_3 - f_3^2 \end{bmatrix} = 0 \tag{42}$$

which leads directly to Eq. (39). The second step is to use standard determinant identities to transform the determinant of Eq. (42) to the form of the construction determinant, Eq. (15). We proceed by replacing column one with the sum of columns one and two resulting in

$$\begin{vmatrix} 1 & 0 & f_1 \\ 1 & 1 & f_2 \\ 1+f_3 & f_3 & f_3^2 \end{vmatrix} = 0 \tag{43}$$

where we have also multiplied column 3 by -1. Dividing row 3 by $1 + f_3$ and rearranging columns we obtain

$$\begin{vmatrix} 0 & f_1 & 1 \\ 1 & f_2 & 1 \\ \dfrac{f_3}{1+f_3} & \dfrac{f_3^2}{1+f_3} & 1 \end{vmatrix} = 0 \tag{44}$$

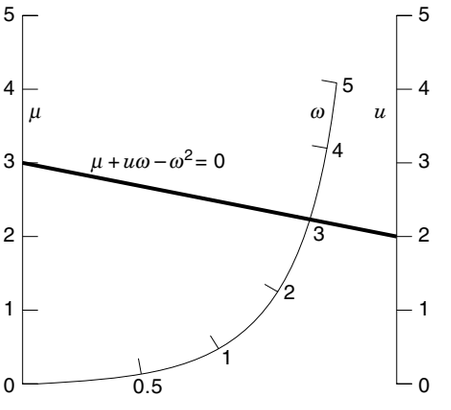


Figure 12. Nonstandard nomogram, with curved axis, generated using the construction determinant.

which is in the form of a construction determinant. The coordinates of the required axes are therefore:

$$\begin{aligned} X_1 &= 0 & Y_1 &= f_1 \\ X_2 &= 1 & Y_2 &= f_2 \\ X_3 &= \frac{f_3}{1+f_3} & Y_3 &= \frac{f_3^2}{1+f_3} \end{aligned} \tag{45}$$

In the simple case where

$$f_1 = \mu, \quad f_2 = u, \quad f_3 = \omega \tag{46}$$

we have the nomogram given in Fig. 12.

BIBLIOGRAPHY

1. M. D. Harpen, A mathematical spreadsheet application for production of entrance skin exposure nomograms, *Med. Phys.,* **23** (2): 241–242, 1996.
2. L. B. Harris, On a limiting case for the distribution of exceedances, with an application to life-testing, *Awards Math. Stat.,* **23** (2): 103–107, 1952.
3. A. S. Levens, *Nomography,* 2nd ed., New York: Wiley, 1965.

MICHAEL D. HARPEN
University of South Alabama

NOMOGRAPH. See NOMOGRAMS.
NONCONTACT THERMOMETERS. See PYROMETERS.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Nonlinear Equations
Standard Article

E. Y. Deeba[1] and S. A. Khuri[2]
[1]University of Houston–Downtown, Houston, TX
[2]University of Houston–Downtown, Houston, TX

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2441
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (169K)

Abstract

The sections in this article are

Decomposition Method

Chandrasekhar H-Equation

Korteweg–DE Vries

Conservative Hyperbolic Systems

Klein–Gordon Equation

| | |

Copyright © 1999-2008 All Rights Reserved.

NONLINEAR EQUATIONS

In this article a decomposition method is presented for solving nonlinear equations arising in the study of radiative transfer such as the Chandrasekhar H-equation, conservative hyperbolic systems and nonlinear waves including the Korteweg–de Vries (KdV), and the Klein–Gordon equations. An essential feature of this numerical technique is its rapid convergence and the high degree of accuracy by which it approximates a solution using only a few terms of its iterative scheme. Other techniques including perturbation methods, finite element, finite difference and Galerkin approximation have been used in the literature to solve nonlinear equations (21–33).

This article presents a decomposition technique for solving nonlinear equations. The original method was first introduced by Adomain (1). Numerous articles have been written using this method to solve partial, ordinary and delay differential equations, nonlinear algebraic equations, and boundary-value problems (1–17). The scheme assumes an infinite solution

$$u = \sum_{n=0}^{\infty} u_n$$

where the terms u_n are recursively determined. A common feature to note of problems solved by the decomposition method is that the solution of the underlying equations obtained by this method approximates the exact solution with a high degree of accuracy using only a few terms of the iterative scheme. A modified version of the technique will be presented to handle some of the nonlinear equations we will be dealing with. Four basic equations that are of importance in mathematical physics and engineering will be considered. In particular, we will present the well-known *H-equation* due to Chandrasekhar (18) which arises in the study of radiative transfer; two nonlinear wave equations: the KdV equation, that arises in the modelling of shallow water waves and the Klein–Gordon equation which is an important model in quantum mechanics. Finally, the method will be implemented for solving a hyperbolic conservative system that models shocks.

In the sections that follow we will present the decomposition method along with the results on convergence, the H-equation, the KdV equation, the hyperbolic conservative system, as well as the Klein–Gordon equation.

DECOMPOSITION METHOD

Recently, there has been a great deal of interest (1–12) in applying the Adomian decomposition technique for solving a wide class of nonlinear equations including algebraic, differential, partial-differential, differential-delay and integro-differential equations. The main thrust of this technique is that the solution which is expressed as an infinite series converges very fast to exact solutions. In (19,20) a proof of convergence of the method has been given by employing fixed point theorems. Most recently, Cherruault et al. (19) presented new proofs of convergence with less stringent hypotheses that are more adaptable to dealing with physical problems. A theoretical analysis for the method has been discussed in (15).

In general, we seek a solution to the following nonlinear equation

$$u = L(u) + N(u) + g \tag{1}$$

where L is a linear operator, N is a nonlinear operator and g is a known function in the underlying function space which is normally a Hilbert space. The decomposition technique consists of representing the solution as an infinite series, namely,

$$u = \sum_{n=0}^{\infty} u_n \tag{2}$$

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

where the terms u_n are to be recursively computed. Also the nonlinear operator N is decomposed as follows:

$$N(u) = \sum_{n=0}^{\infty} A_n \tag{3}$$

where $A_n = A_n(u_0, u_1, u_2, \ldots, u_n)$ are the so-called *Adomian polynomials.* Substituting Eqs. (2) and (3) into Eq. (1) yields

$$\sum_{n=0}^{\infty} u_n = \sum_{n=0}^{\infty} L(u_n) + \sum_{n=0}^{\infty} A_n(u_0, u_1, \ldots, u_n) + g \tag{4}$$

Assuming convergence of the series in Eq. (4), both sides of Eq. (4) will match by setting

$$\begin{cases} u_0 = g \\ u_1 = L(u_0) + A_0(u_0) \\ u_2 = L(u_1) + A_1(u_0, u_1) \\ \vdots \\ u_{n+1} = L(u_n) + A_n(u_0, u_1, \ldots, u_n) \\ \vdots \end{cases} \tag{5}$$

Thus, from Eq. (5) the $u_n's$ given in Eq. (2) can be obtained in a recurrent manner and hence u is determined.

There are important questions to raise now:

1. How are the Adomian polynomials A_n determined?
2. Do the series in Eqs. (2) and (3) always converge? If so, to which function do they converge?

Before we proceed, we give a heuristic argument for determining A_n's when $N(u) = f(u)$ and $f(u)$ is a scalar function.

The Taylor expansion of $f(u)$ around u_0 is:

$$f(u) = f(u_0) + f^{(1)}(u_0)(u - u_0) + \frac{1}{2!} f^{(2)}(u_0)(u - u_0)^2 + \ldots \tag{6}$$

If u is given as an infinite sum

$$u = \sum_{n=0}^{\infty} u_n = u_0 + u_1 + u_2 + \ldots \tag{7}$$

then upon substituting the difference $u - u_0$ from Eq. (7) into Eq. (6), we get

$$\begin{aligned} f(u) = f(u_0) &+ f^{(1)}(u_0)(u_1 + u_2 + \ldots) \\ &+ \frac{1}{2!} f^{(2)}(u_0)(u_1 + u_2 + \ldots)^2 + \ldots \end{aligned} \tag{8}$$

and when simplified this results in:

$$\begin{aligned} f(u) = f(u_0) &+ f^{(1)}(u_0)(u_1 + u_2 + u_3 + \ldots) \\ &+ \frac{1}{2!} f^{(2)}(u_0)(u_1^2 + 2u_1u_2 + 2u_1u_3 + u_2^2 + 2u_2u_3 + u_3^2 + \ldots) \\ &+ \frac{1}{3!} f^{(3)}(u_0)(u_1^3 + 3u_1^2 + 3u_1^2u_2 + 3u_1^2u_3 + 3u_1u_2^2 + \ldots) \\ &+ \ldots \end{aligned} \tag{9}$$

Adomian polynomials are obtained by a reordering and rearranging of the terms given in Eq. (9). Indeed, to determine the Adomian polynomials, one needs to determine the order of each term in Eq. (9) which actually depends on both the subscripts and the exponents of the u_n's. For example, u_1 is of order 1; u_1^2 is of order 2; u_2^3 is of order 6; and so on. If a particular term involves the multiplication of u_n's, its order is determined by the sum of the orders of the u_n's in the term. For example, $u_2^3u_1^2$ is of order 8 since (3)(2) + (2)(1) = 8. Therefore, rearranging the terms in the expansion Eq. (9) according to the order and assuming that $N(u)$ is as given in Eq. (3) will give the A_n as

$$\begin{cases} A_0 = f(u_0) \\ A_1 = u_1 f^{(1)}(u_0) \\ A_2 = u_2 f^{(1)}(u_0) + \dfrac{1}{2!} u_1^2 f^{(2)}(u_0) \\ A_3 = u_3 f^{(1)}(u_0) + u_1u_2 f^{(2)}(u_0) + \dfrac{1}{3!} u_1^3 f^{(3)}(u_0) \\ \vdots \end{cases} \tag{10}$$

We will now briefly present the general method and refer the reader to (15) for a more detailed study. As was pointed out earlier the Adomian algorithm assumes a series solution for u given by Eq. (2) and that the nonlinear operator $N(u)$ can be decomposed into:

$$N(u) = \sum_{n=0}^{\infty} A_n \tag{11}$$

The Adomian polynomials A_n's are given by the general formula

$$A_n = \frac{1}{n!} \frac{d^n}{d\lambda^n} \left[N\left(\sum \lambda^i u_i \right) \right]_{\lambda=0} \qquad n = 0, 1, \ldots \tag{12}$$

Once the A_n are determined by Eq. (12), one can recurrently determine the terms u_n of the series and hence the solution u of Eq. (1). The convergence of the series solution has been established (15,19). The two hypotheses that are necessary for proving convergence of Adomian technique are given in (19) by:

1. The nonlinear functional Eq. (1) has a series solution $\sum_{n=0}^{\infty} u_n$ such that $\sum_{n=0}^{\infty} (1 + \epsilon)^n |u_n| < \infty$ where $\epsilon > 0$ may be very small.
2. The nonlinear operator $N(u)$ is analytic and can be developed in series according to u: $N(u) = \sum_{n=0}^{\infty} \alpha_n u^n$

These conditions are generally satisfied in the modeling of many physical problems.

To illustrate the scheme, let $N(u)$ be a nonlinear function of u, say $f(u)$, where

$$u = u_0 + \lambda u_1 + \lambda^2 u_2 + \ldots$$

then the first four Adomian's polynomials A_n are given by

$$\begin{cases} A_0 = f(u(\lambda))|_{\lambda=0} = f(u_0) \\ A_1 = (df/du)(du/d\lambda)|_{\lambda=0} \\ A_2 = \dfrac{1}{2}[(d^2g/du^2)(du/d\lambda)^2 + (df/du)(d^2u/d\lambda^2)]|_{\lambda=0} \\ A_3 = \dfrac{1}{6}[(d^3f/du^3)(du/d\lambda)^3 + 2(d^2f/du^2)(du/d\lambda)(d^2u/d\lambda^2) \\ \qquad + (d^2f/du^2)(d^2u/d\lambda^2)(du/d\lambda) + (df/du)(d^3u/d\lambda^3)]|_{\lambda=0} \\ \vdots \end{cases} \tag{13}$$

The $A_n's$ can finally be written in the following convenient way

$$A_n = \sum_{v=1}^{n} c(v,n) f^{(v)}(u_0) \tag{14}$$

This results in the polynomials given in Eq. (10).

In the next sections, this method and a modified version of it will be used for solving several interesting nonlinear equations which are of physical importance. We will begin with the Chandrasekhar equation (18).

CHANDRASEKHAR *H*-EQUATION

In this section, the decomposition method is applied to the Chandrasekhar H-equation given by:

$$H(x) = 1 + H(x) \int_0^1 \frac{x}{x+t} \Psi(t) H(t) dt \tag{15}$$

where the H-function, $H(x)$, measures the emergent radiation and $\Psi(t)$ is referred to as the characteristic function and is a measure of phase. This equation arises in the formulation of problems in the theory of radiative transfer in semi-infinite atmospheres. *Radiative transfer* is the angular distribution of the emergent radiations which results from scattering. For standard problems in isotropic scattering, these angular distributions of the emergent radiations are directly expressed in terms of the *H-functions.*

In Eq. (15), the function $\Psi(t)$ is usually a nonnegative even polynomial in t satisfying

$$\int_0^1 \Psi(t) dt \leq \frac{1}{2}$$

It is well known that a positive and continuous solution of Eq. (15) exists (18).

A case to consider, is when the law of diffuse reflection of scattering is given in terms of the phase function $\Psi_0(1 + x \cos \theta)$. This can be expressed in terms of the H-equation corresponding to the following particular choice of the *characteristic function,* $\Psi(t)$ (18)

$$\Psi(t) = \frac{1}{2} \Psi_0 [1 + x(1 - \Psi_0) t^2] \tag{16}$$

where Ψ_0 is a constant.

If we set

$$z(x) = \frac{1}{H(x)} \tag{17}$$

then Eq. (15) can be written as

$$z(x) = 1 - x \int_0^1 \frac{\Psi(t)}{x+t} \frac{1}{z(t)} dt \tag{18}$$

The nonlinear term in Eq. (18) is,

$$N(z) = f(z) = \frac{1}{z} \tag{19}$$

Thus, upon writing $z(x) = \sum_{n=0}^{\infty} z_n(x)$, and $N(z(x)) = 1/z(x) = \sum_{n=0}^{\infty} A_n(z(x))$ in terms of the Adomian polynomials, Eq. (18) becomes,

$$\sum_{n=0}^{\infty} z_n(x) = 1 - x \sum_{n=0}^{\infty} \int_0^1 \frac{\Psi(t)}{x+t} A_n(t) dt \tag{20}$$

Applying the decomposition method to Eq. (20), the various iterates are given by

$$z_0 = 1 \tag{21}$$

and

$$z_{n+1}(x) = -x \int_0^1 \frac{\Psi(t)}{x+t} A_n(t) dt, \quad n \geq 1 \tag{22}$$

where, upon using Eq. (10), the Adomian polynomials for the nonlinear operator given in Eq. (19) are:

$$\begin{cases} A_0 = \dfrac{1}{z_0} \\ A_1 = -\dfrac{1}{z_0^2} z_1 \\ A_2 = -\dfrac{1}{z_0^2} z_2 + \dfrac{1}{z_0^3} z_1^2 \\ A_3 = -\dfrac{1}{z_0^2} z_3 + \dfrac{2}{z_0^3} z_1 z_2 - \dfrac{1}{z_0^4} z_1^3 \\ \vdots \end{cases} \tag{23}$$

Substituting Eq. (21) into Eq. (23) gives:

$$\begin{cases} A_0 = 1 \\ A_1 = -z_1 \\ A_2 = z_1^2 - z_2 \\ A_3 = -z_1^3 + 2z_1 z_2 - z_3 \end{cases} \tag{24}$$

From Eqs. (24), (21), and (22) the first few iterates are:

$$\begin{cases} z_0(x) = 1 \\ z_1(x) = -x \displaystyle\int_0^1 \frac{\Psi(t)}{x+t} dt \\ z_2(x) = x \displaystyle\int_0^1 \frac{\Psi(t)}{x+t} z_1(t) dt \\ z_3(x) = -x \displaystyle\int_0^1 \frac{\Psi(t)}{x+t} (z_1^2(t) - z_2(t)) dt \\ \vdots \end{cases} \tag{25}$$

Evaluating the integrals in Eq. (25) (using the computer algebra system maple) yields

$$\begin{cases} z_0(x) = 1 \\ z_1(x) = -\dfrac{1}{4}x\Psi_0[(\Psi_0 - 1)(x - 2x^2) \\ \qquad - 2(1 + x^3 - \Psi_0 x^3)(\ln x + 1 - \ln x)] \\ z_2(x) = -\dfrac{1}{16}x^2\Psi_0^2[2(1 - \Psi_0)x^2 - (1 - \Psi_0)x \\ \qquad - 2(1 + x^3 - 2\Psi_0 x^3)(\ln x + 1 - \ln x)]^2 \\ \vdots \end{cases} \tag{26}$$

The solution of the H-equation Eq. (15) is therefore

$$H(x) = \frac{1}{z(x)} = \frac{1}{\sum_{n=0}^{\infty} z_n} \tag{27}$$

where the $z_n's$ are given in Eq. (26).

The H-functions given by Eq. (27) for various values of Ψ_0 are given in Table 1 using the decomposition method. The values in Table 2 were obtained by Chandrasekhar and Breen (18) by a process of iteration, where the solution started with the fourth approximation for $H(x)$ in terms of the Gaussian division and characteristic roots. The iterates were evaluated at some points and the intermediate values were predicted by interpolating among the differences between the successive iterates. Upon comparing Tables 1 and 2 we note that the decomposition technique with only *three iterations* yields approximately the same values as those in Table 2, derived by Chandrasekhar and Breen, with error less than 1%.

Another satisfactory check for the accuracy of the decomposition method is provided by evaluating the integral

$$\int_0^1 \Psi(t)H(t)dt \tag{28}$$

Table 1. The H-Functions Defined in Terms of the Characteristic Function $\Psi(x) = \frac{1}{2}\Psi_0[1 + x(1 - \Psi_0)t^2]$ Are Evaluated Using Adomian's Method with Three Iterations

x	$\Psi_0 = 0.1$	$\Psi_0 = 0.2$	$\Psi_0 = 0.3$	$\Psi_0 = 0.4$	$\Psi_0 = 0.5$
0.00	1.0000	1.0000	1.0000	1.0000	1.0000
0.05	1.0078	1.0158	1.0241	1.0326	1.0415
0.10	1.0125	1.0256	1.0393	1.0538	1.0691
0.15	1.0162	1.0334	1.0517	1.0713	1.0923
0.20	1.0193	1.0400	1.0624	1.0866	1.1129
0.25	1.0220	1.0459	1.0719	1.0004	1.1316
0.30	1.0245	1.0512	1.0806	1.1130	1.1490
0.35	1.0267	1.0561	1.0886	1.1248	1.1653
0.40	1.0288	1.0606	1.0961	1.1359	1.1809
0.45	1.0307	1.0649	1.1032	1.1465	1.1957
0.50	1.0326	1.0690	1.1100	1.1566	1.2100
0.55	1.0343	1.0728	1.1165	1.1663	1.2237
0.60	1.0360	1.0765	1.1227	1.1758	1.2372
0.65	1.0376	1.0801	1.1288	1.1850	1.2503
0.70	1.0391	1.0836	1.1347	1.1939	1.2631
0.75	1.0406	1.0870	1.1404	1.2026	1.2758
0.80	1.0421	1.0902	1.1460	1.2113	1.2882
0.85	1.0435	1.0934	1.1515	1.2198	1.3005
0.90	1.0449	1.0966	1.1570	1.2281	1.3126
0.95	1.0462	1.0997	1.1623	1.2364	1.3248
1.00	1.0475	1.1027	1.1676	1.2446	1.3367

Table 2. The H-Functions Defined in Terms of the Characteristic Function $\Psi(x) = \frac{1}{2}\Psi_0[1 + x(1 - \Psi_0)t^2]$ As Obtained by Chandrasekhar and Breen

x	$\Psi_0 = 0.1$	$\Psi_0 = 0.2$	$\Psi_0 = 0.3$	$\Psi_0 = 0.4$	$\Psi_0 = 0.5$
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.05	1.0089	1.0183	1.0280	1.0383	1.0492
0.10	1.0145	1.0297	1.0459	1.0632	1.0817
0.15	1.0188	1.0388	1.0602	1.0832	1.1084
0.20	1.0224	1.0463	1.0722	1.1003	1.1311
0.25	1.0254	1.0528	1.0825	1.1151	1.1511
0.30	1.0280	1.0584	1.0916	1.1281	1.1689
0.35	1.0303	1.0634	1.0996	1.1398	1.1850
0.40	1.0324	1.0678	1.1069	1.1504	1.1996
0.45	1.0343	1.0719	1.1135	1.1600	1.2129
0.50	1.0359	1.0755	1.1194	1.1688	1.2252
0.55	1.0375	1.0788	1.1249	1.1769	1.2365
0.60	1.0389	1.0819	1.1300	1.1844	1.2470
0.65	1.0401	1.0847	1.1346	1.1913	1.2568
0.70	1.0413	1.0873	1.1389	1.1978	1.2659
0.75	1.0424	1.0897	1.1429	1.2038	1.2745
0.80	1.0434	1.0919	1.1467	1.2094	1.2825
0.85	1.0444	1.0940	1.1502	1.2047	1.2900
0.90	1.0453	1.0960	1.1535	1.2196	1.2972
0.95	1.0461	1.0978	1.1566	1.2243	1.3039
1.00	1.0469	1.0995	1.1595	1.2287	1.3103

numerically, where $H(t)$ is the solution obtained from the decomposition technique and then comparing it with its exact value (18) which is given by

$$\int_0^1 \Psi(t)H(t)dt = 1 - \left[1 - 2\int_0^1 \Psi(t)dt\right]^{1/2} \tag{29}$$

For the particular $\Psi(t)$ given in Eq. (16) and upon substituting it into Eq. (29), we get the exact value of the integral

$$\int_0^1 \Psi(t)H(t)dt = 1 - \left[1 - \frac{1}{3}\Psi_0(1 + x - \Psi_0)\right]^{1/2} \tag{30}$$

Table 3 shows that for $x = 1$ and different values of Ψ_0, the error between the exact value in Eq. (30) and the numerical values of the integral in Eq. (28) is less than 1% with H-function being approximated using only three terms of the decomposition method.

KORTEWEG–DE VRIES

In this section a modified decomposition algorithm is presented for solving the Korteweg–de Vries (KdV) equation that

Table 3. Comparison of the Integrals $\int_0^1 \Psi(x)H(x)\,dx$, Where $H(x)$ Is the H-Function Obtained Using Decomposition Method with Three Iterations, with Their Exact Values $1 - [1 - \frac{1}{3}\Psi_0(2 - \Psi_0)^{1/2}]$

Ψ_0	Decomposition	Exact	Ψ_0	Decomposition	Exact
0.10	0.06469	0.06726	0.35	0.23252	0.24226
0.15	0.09742	0.10139	0.40	0.26767	0.27889
0.20	0.13051	0.13590	0.45	0.30351	0.31626
0.25	0.16401	0.17084	0.50	0.34040	0.35450
0.30	0.19798	0.20627	0.55	0.37802	0.39378

arises in the study of nonlinear waves. Versions of this equation have been extensively studied both analytically and numerically (21–26). The general KdV equation is given by

$$u_t + f(u)u_x + \beta u_{xxx} = 0 \tag{31}$$

The following choice of the KdV equation which arises in shallow water theory is considered:

$$u_t + (\alpha + \epsilon u)u_x + \beta u_{xxx} = 0 \tag{32}$$

with initial condition

$$u(x, 0) = g(x) \tag{33}$$

where α, β, and ϵ are constants.

Define the linear operators

$$L_t = \frac{\partial}{\partial t}, \quad L_1 = \frac{\partial}{\partial x}, \quad L_2 = \frac{\partial^3}{\partial x^3} \tag{34}$$

The inverse operators are the indefinite integrals given by

$$L_t^{-1} = \int_0^t dt, \quad L_1^{-1} = \int_0^x dx, \quad L_2^{-1} = \int_0^x dx \int_0^x dx \int_0^x dx \tag{35}$$

The conditions under which the decomposition algorithm converges imply the existence of L_t^{-1}, L_1^{-1} and L_2^{-1}. Equation (32) can be written in the following operator form:

$$L_t u = -f(u)L_1 u - \beta L_2 \boldsymbol{u} \tag{36}$$

where

$$f(u) = \alpha + \epsilon u \tag{37}$$

Applying the inverse operator L_t^{-1}, to both sides of Eq. (36) gives

$$L_t^{-1}L_t u = -L_t^{-1}(f(u)L_1 u) - \beta L_t^{-1}L_2 u \tag{38}$$

or

$$u(x, t) = u(x, 0) - L_t^{-1}(f(u)L_1 u) - \beta L_t^{-1}L_2 u \tag{39}$$

Following the decomposition method, the term u_0 is determined as

$$u_0(x, t) = u(x, 0) \tag{40}$$

The other iterations are obtained via:

$$u_{n+1} = -L_t^{-1}[f(u_n)L_1 u_n] - \beta L_t^{-1}L_2 u_n, \quad n \geq 0 \tag{41}$$

Consider now the argument $f(u)L_1 u$ of the first term on the right-hand side of Eq. (38). Since

$$u = \sum_{n=0}^{\infty} u_n$$

we have

$$L_1 u = L_1 u_0 + L_1 u_1 + L_1 u_2 + \ldots + L_1 u_n + \ldots \tag{42}$$

Expressing $N(u) = f(u)$ in terms of Adomian polynomials

$$f(u) = A_0 + A_1 + A_2 + \ldots + A_n + \ldots \tag{43}$$

then the product $f(u)L_1 u$ can be expanded, after rearranging terms, as follows:

$$\begin{aligned} f(u)L_1 u = {} & [A_0 L_1 u_0] + [A_0 L_1 u_1 + A_1 L_1 u_0] \\ & + [A_0 L_1 u_2 + A_2 L_1 u_0 + A_1 L_1 u_1] \\ & + [A_3 L_1 u_0 + A_2 L_1 u_1 + A_1 L_1 u_2 + A_0 L_1 u_3] \\ & + \ldots \end{aligned} \tag{44}$$

From Eq. (44), the first three *modified Adomian polynomials* $B_i's$ for the nonlinear operator $f(u)u_x = f(u)L_1 u = \sum_{n=0}^{\infty} B_n$ are:

$$\begin{cases} B_0 = A_0 L_1 u_0 \\ B_1 = A_0 L_1 u_1 + A_1 L_1 u_0 \\ B_2 = A_0 L_1 u_2 + A_2 L_1 u_0 + A_1 L_1 u_1 \end{cases} \tag{45}$$

Substituting Eq. (44) into Eq. (41) and then replacing the A_n by their values as given in Eq. (10), we obtain the following first three iterates of the *modified Adomian algorithm*

$$\begin{cases} u_1 = -L_t^{-1}[f(u_0)L_1 u_0] - \beta L_t^{-1}L_2 u_0 \\ u_2 = -L_t^{-1}[f(u_0)L_1 u_1 + u_1 f'(u_0)L_1 u_0] - \beta L_t^{-1}L_2 u_1 \\ u_3 = -L_t^{-1}\left[f(u_0)L_1 u_2 + u_1 f'(u_0)L_1 u_1 \right. \\ \qquad \left. + \left(u_2 f'(u_0) + \frac{u_1^2}{2!} f''(u_0) \right) L_1 u_0 \right] - \beta L_t^{-1}L_2 u_2 \\ \vdots \end{cases} \tag{46}$$

Implementing the modified algorithm to Eqs. (32) and (33), for the special case where $f(u)$ is linear in u and is given in Eq. (37), one obtains

$$u(x, t) = g(x) - L_t^{-1}[(\alpha + \epsilon u)L_1 u] - \beta L_t^{-1}L_2 u \tag{47}$$

It then follows, by using Eqs. (46) and (47), that the iterates of the KdV equation are given by:

$$\begin{cases} u_0 = g(x) \\ u_1 = -L_t^{-1}[(\alpha + \epsilon u_0)L_1 u_0] - \beta L_t^{-1}L_2 u_0 \\ u_2 = -L_t^{-1}[(\alpha + \epsilon u_0)L_1 u_1 + \epsilon u_1 L_1 u_0] - \beta L_t^{-1}L_2 u_1 \\ u_3 = -L_t^{-1}[(\alpha + \epsilon u_0)L_1 u_2 + \epsilon u_1 L_1 u_1 \\ \qquad + (\epsilon u_2)L_1 u_0] - \beta L_t^{-1}L_2 u_2 \\ \vdots \end{cases} \tag{48}$$

We now pick a special function

$$g(t) = A \operatorname{sech}^2\left(\sqrt{\frac{\epsilon A}{12\beta}} t \right)$$

to compare the exact solution of Eqs. (32) and (33) with the numerical solution obtained by the decomposition method. From Eq. (48) the first two iterates are given by

$$u_0 = A \operatorname{sech}^2\left(\sqrt{\frac{\epsilon A}{12\beta}} t \right) \tag{49}$$

and

$$u_1 = \frac{1}{9}\frac{1}{\cosh^3\left(\frac{1}{2}\sqrt{\frac{\epsilon A}{3\beta}}t\right)}A(3\alpha+\epsilon A)\sqrt{\frac{3\epsilon A}{\beta}}t\sinh\left(\frac{1}{2}\sqrt{\frac{\epsilon A}{3\beta}}t\right) \tag{50}$$

Many other iterates were generated using MAPLE. Table 4 shows the errors obtained upon solving the KdV equation after normalizing the constants (i.e., we set $\alpha = \beta = A = 1$ and $\epsilon = 0.02$) and using only five iterations of the decomposition method. It is to be noted that only five iterates were needed to obtain an error of less than $10^{-5}\%$. The overall errors can be made even much smaller by adding new terms of the decomposition.

CONSERVATIVE HYPERBOLIC SYSTEMS

In this section, we will consider the nonlinear partial differential equation:

$$u_t + \frac{\partial}{\partial x}f(u) = 0 \tag{51}$$

$$u(x,0) = h(x) \tag{52}$$

which arises in the formulation of conservative hyperbolic systems. A *hyperbolic system* is one for which the wave speeds coalesce along certain curves in the state space. Such systems occur, for example, in the modeling of oil recovery problems. Many studies have dealt with the numerical diffusion, resolution and shock fronts and spurious oscillations which arise in approximating the solution of Eqs. (51) and (52), (27–29).

Following the decomposition analysis, define the linear operators

$$L_t = \frac{\partial}{\partial t} \tag{53}$$

and

$$L_1 = \frac{\partial}{\partial x} \tag{54}$$

Consequently, Eq. (51) can be written in terms of these operators as

$$L_t u + f'(u)L_1 u = 0 \tag{55}$$

Applying the inverse operator of L_t, namely L_t^{-1}, defined by

$$[L_t^{-1}g](t) := \int_0^t g(y)dy \tag{56}$$

Table 4. Error Obtained Using Decomposition Method with Five Iterations for the KdV Equation

x	$t = 0.2$	$t = 0.4$	$t = 0.6$	$t = 0.8$
0.2	3.1×10^{-9}	4.85×10^{-8}	2.45×10^{-7}	7.79×10^{-7}
0.4	2.8×10^{-9}	4.85×10^{-8}	2.46×10^{-7}	7.78×10^{-7}
0.6	2.9×10^{-9}	4.89×10^{-8}	2.46×10^{-7}	7.76×10^{-7}
0.8	3.1×10^{-9}	4.86×10^{-8}	2.45×10^{-7}	7.74×10^{-7}

to both sides of Eq. (55) yields

$$u(x,t) = u(x,0) - L_t^{-1}[f'(u)L_1 u] \tag{57}$$

from which it follows, upon using the initial condition Eq. (52),

$$u(x,t) = h(x) - L_t^{-1}[f'(u)L_1 u] \tag{58}$$

If we set

$$N(x) = f(u) = \sum_{n=0}^{\infty} A_n \quad \text{and} \quad u = \sum_{n=0}^{\infty} u_n$$

then the term $f'(u)L_1 u$ in Eq. (58) can be expanded in terms of the modified Adomian polynomials B_n's, where

$$f'(u)L_1 u = \sum_{n=0}^{\infty} B_n$$

Hence, we have

$$u(x,t) = h(x) - L_t^{-1}\left[\sum_{n=0}^{\infty} B_n\right] \tag{59}$$

To derive the first few B_n's we have

$$\sum_{n=0}^{\infty} B_n = f'(u)L_1 u = \left(\sum_{n=0}^{\infty} A_n'\right)\left(\sum_{n=0}^{\infty} L_1 u_n\right) \tag{60}$$

and the A_n' are the Adomian polynomials of $f'(u)$. Upon collecting terms in Eq. (60), the first few B_n's are:

$$\begin{aligned} B_0 &= A_0' L_1 u_0 \\ B_1 &= A_0' L_1 u_1 + A_1' L_1 u_0 \\ B_1 &= A_0' L_1 u_2 + A_1' L_1 u_1 + A_2' L_1 u_0 \end{aligned} \tag{61}$$

Applying the decomposition algorithm to Eq. (59), the iterates are given by

$$u_0 = h(x) \tag{62}$$

and

$$u_{n+1} = -L_t^{-1}[B_n], \quad n \geq 1 \tag{63}$$

where the B_n's are given in Eq. (61). Consider the following two special cases:

Case 1. $f(u) = -u^2$

$$u_t - \frac{\partial}{\partial x}u^2 = 0 \tag{64}$$

with initial condition

$$u(x,0) = x \tag{65}$$

Since, for this case,

$$f'(u) = -2u = \sum_{n=0}^{\infty} A_n'$$

hence using Eq. (10) the Adomian polynomials A'_n of $f'(u)$ are given by

$$A'_n = -2u_n, \quad n \geq 0 \tag{66}$$

Using Eqs. (61)–(63) and Eq. (66) the first few iterates are:

$$u_0 = x \tag{67}$$

$$u_1 = 2L_t^{-1}[u_0 L_1 u_0] = 2L_t^{-1}[(x)(1)] = 2xt \tag{68}$$

$$\begin{aligned} u_2 &= 2L_t^{-1}[u_0 L_1 u_1 + u_1 L_1 u_0] \\ &= 2L_t^{-1}[(x)(2t) + (2xt)(1)] = 2L_t^{-1}[4xt] = 4xt^2 \end{aligned} \tag{69}$$

Similarly,

$$\begin{aligned} u_3 &= 2L_t^{-1}[u_0 L_1 u_2 + u_1 L_1 u_1 + u_2 L_1 u_0] \\ &= 2L_t^{-1}[(x)(4t^2) + (2xt)(2t) + (4xt^2)(1)] = 24L_t^{-1}[xt^2] = 8xt^3 \end{aligned} \tag{70}$$

Summing these iterates yields

$$u = x + 2xt + 4xt^2 + 8xt^3 + 16xt^4 + \ldots \tag{71}$$

If $-\frac{1}{2} < t < \frac{1}{2}$, then Eq. (71) can be written in closed form as

$$u(x,t) = \frac{x}{1-2t} \tag{72}$$

which is the exact solution of Eqs. (64) and (65).

Case 2. $f(u) = \cos u$

$$u_t + \frac{\partial}{\partial x}\cos u = 0 \tag{73}$$

with initial condition

$$u(x,0) = x \tag{74}$$

For this case

$$f'(u) = -\sin u = \sum_{n=0}^{\infty} A'_n$$

hence using Eq. (10) the Adomian polynomials A'_n of $f'(u)$ are given by

$$\begin{aligned} A'_0 &= -\sin u_0 \\ A'_1 &= -u_1 \cos u_0 \\ A'_2 &= -u_2 \cos u_0 + \frac{1}{2!}u_1^2 \sin u_0 \end{aligned} \tag{75}$$

Using Eq. (61), Eq. (63), and (75) gives the following first few iterates

$$u_0 = x \tag{76}$$

$$u_1 = L_t^{-1}[\sin u_0 L_1 u_0] = L_t^{-1}[(\sin x)(1)] = t\sin x \tag{77}$$

$$\begin{aligned} u_2 &= L_t^{-1}[u_1 \cos u_0 L_1 u_0 + \sin u_0 L_1 u_1] \\ &= L_t^{-1}[(t\sin x)(\cos x)(1) + (\sin x)(t\cos x)] \\ &= L_t^{-1}[2t\sin x \cos x] = \frac{1}{2}t^2 \sin 2x \end{aligned} \tag{78}$$

Similarly,

$$\begin{aligned} u_3 &= L_t^{-1}\left[\sin u_0 L_1 u_2 + u_1 \cos u_0 L_1 u_1 \right. \\ &\quad \left. + \left(u_2 \cos u_0 - \frac{1}{2}u_1^2 \sin u_0\right) L_1 u_0\right] \\ &= \frac{1}{3}t^3\left(\sin 3x - \frac{1}{2}\sin^3 x\right) \end{aligned} \tag{79}$$

Upon summing these iterates we get

$$u = x + \sin xt + \frac{1}{2}\sin 2xt^2 + \frac{1}{3}\left(\sin 3x - \frac{1}{2}\sin^3\right)t^3 + \ldots \tag{80}$$

KLEIN–GORDON EQUATION

In this section, we will focus on the nonlinear Klein–Gordon equation given generally by:

$$\frac{\partial^2 u(x,t)}{\partial t^2} - \Delta u(x,t) + ku(x,t) + f(u(x,t)) = g(x,t) \tag{81}$$

$$u(x,0) = b_0(x), \quad \frac{\partial u}{\partial t}(x,0) = b_1(x) \tag{82}$$

with

$$x = (x_1, x_2, \ldots, x_m) \in R^m, \quad t \in (0,T]$$

where

$$\Delta = \sum_{j=0}^{m} \frac{\partial^2}{\partial x_j^2} \tag{83}$$

and k is real, f is a given nonlinear function, and h is a known function.

The Klein–Gordon equation is an important mathematical model in quantum mechanics and also occurs in relativistic physics as a model of dispersive phenomena [see (25,26,28–33)].

Following the decomposition scheme, define

$$L_t = \frac{\partial^2}{\partial t^2}, \quad L_{x_i} = \frac{\partial^2}{\partial x_i^2}, \quad i = 1, 2, \ldots, m \tag{84}$$

Consequently, Eq. (81) can be written in the following operator form

$$L_t u = \sum_{i=1}^{m} L_{x_i} u - ku - f(u) + g \tag{85}$$

It was shown in (24) that Eq. (81) with the conditions of Eq. (82) possesses a unique solution. Thus the inverse operator of L_t, namely L_t^{-1}, exists and is the two-fold indefinite integral, that is,

$$[L_t^{-1}h](t) := \int_0^t du \int_0^u dv h(v) \tag{86}$$

Applying L_t^{-1}, to both sides of Eq. (84) yields

$$L_t^{-1}L_t u = \sum_{i=0}^{m} L_t^{-1}L_{x_i}u - kL_t^{-1}u - L_t^{-1}(f(u)) + L_t^{-1}g \tag{87}$$

and upon using the initial conditions of Eq. (82) it follows that

$$\begin{aligned} u(x,t) = b_0(x) + b_1(x)t + \sum_{i=0}^{m} L_t^{-1}L_{x_i}u - kL_t^{-1}u \\ - L_t^{-1}(f(u)) + L_t^{-1}g \end{aligned} \tag{88}$$

Following the decomposition technique the first term u_0 is determined as

$$u_0(x,t) = b_0(x) + b_1(x)t + L_t^{-1}(g(x,t)) \tag{89}$$

Setting $N(u) = f(u) = \sum_{n=0}^{\infty} A_n$, then the next iterates are determined as

$$u_{n+1} = \sum_{i=0}^{m} L_t^{-1}L_{x_i}u_n - kL_t^{-1}u_n - L_t^{-1}A_n, \quad n \geq 0 \tag{90}$$

Replacing the A_n in Eq. (90) by their values as given in Eq. (10), then the first three iterates are given by

$$\begin{cases} u_1 = \displaystyle\sum_{i=0}^{m} L_t^{-1}L_{x_i}u_0 - kL_t^{-1}u_0 - L_t^{-1}(f(u_0)) \\ u_2 = \displaystyle\sum_{i=0}^{m} L_t^{-1}L_{x_i}u_1 - kL_t^{-1}u_1 - L_t^{-1}(u_1 f(u_0)) \\ u_3 = \displaystyle\sum_{i=0}^{m} L_t^{-1}L_{x_i}u_2 - kL_t^{-1}u_2 \\ \qquad - L_t^{-1}\left(u_2\dfrac{d}{du_0}f(u_0) + \dfrac{u_1^2}{2!}\dfrac{d^2}{du_0^2}f(u_0)\right) \end{cases} \tag{91}$$

We will show through two examples that the number of terms required to obtain an accurate computable solution is very small. The outcome of the decomposition method will be compared with the known solution to the underlying Klein–Gordon equation. The solutions obtained are generated by using MAPLE.

Example 1. Consider the Klein–Gordon equation of the form

$$\begin{aligned} u_{tt} - u_{xx} + \frac{\pi^2}{4}u + u^3 = \frac{x^3}{4}\cos\frac{3\pi}{2}t + \frac{3x^3}{4}\cos\frac{\pi}{2}t, \\ u(x,0) = x, u_t(x,0) = 0 \end{aligned} \tag{92}$$

Equation (88) implies that

$$\begin{aligned} u(x,t) = x + L_t^{-1}L_x u - \frac{\pi^2}{4}L_t^{-1}u - L_t^{-1}u^3 \\ + L_t^{-1}\left(\frac{x^3}{4}\cos\frac{3\pi}{2}t + \frac{3x^3}{4}\cos\frac{\pi}{2}t\right) \end{aligned} \tag{93}$$

Table 5. Error Obtained Using Decomposition Method with Three Iterations for the Klein–Gordon Eq. (92)

x	$t = 0.1$	$t = 0.3$	$t = 0.5$
0.1	8.1×10^{-9}	5.8×10^{-6}	1.1×10^{-4}
0.3	2.1×10^{-8}	1.5×10^{-5}	3.1×10^{-4}
0.5	2.2×10^{-8}	1.6×10^{-5}	3.4×10^{-4}

Equations (89) and (91) imply that the various iterates are given by:

$$\begin{aligned} u_0 &= x + L_t^{-1}\left(\frac{x^3}{4}\cos\frac{3}{2}\pi t + \frac{3x^3}{4}\cos\frac{\pi}{2}t\right) \\ &= x + \frac{1}{9\pi^2}x^3\left(28 - \cos\frac{3}{2}\pi t + 27\cos\frac{1}{2}\pi t\right) \end{aligned} \tag{94}$$

and

$$u_1 = L_t^{-1}L_x u_0 - \frac{\pi^2}{4}L_t^{-1}u_0 - L_t^{-1}u_0^3 \tag{95}$$

In a like manner,

$$u_2 = L_t^{-1}L_x u_1 - \frac{\pi^2}{4}L_t^{-1}u_1 - 3L_t^{-1}u_0^2 u_1 \tag{96}$$

Table 5 shows the error obtained by comparing the decomposition method with three iterations and the exact solution which is $u = x\cos\pi/2t$.

Example 2. Consider the Klein–Gordon equation of the form

$$\begin{aligned} u_{tt} - u_{xx} + \pi^2 u + u^2 = x^4\cos^2\pi t - 2\cos\pi t, \\ u(x,0) = x^2, u_t(x,0) = 0 \end{aligned} \tag{97}$$

Equation (88) implies that

$$\begin{aligned} u(x,t) = x^2 + L_t^{-1}L_x u - \pi^2 L_t^{-1}u - L_t^{-1}u^2 \\ + L_t^{-1}(x^4\cos^2\pi t - 2\cos\pi t) \end{aligned} \tag{98}$$

Equations (89) and (91) imply that the various iterates are given by:

$$\begin{aligned} u_0 &= x^2 + L_t^{-1}(x^4\cos^2\pi t - \cos\pi t) \\ &= -\frac{2}{\pi^2} + x^2 + \frac{1}{4\pi^2}(\pi^2 x^4 t^2 + 8\cos\pi t + x^4\sin^2\pi t) \end{aligned} \tag{99}$$

and

$$u_1 = L_t^{-1}L_x u_0 - \pi^2 L_t^{-1}u_0 - L_t^{-1}u_0^2 \tag{100}$$

In a like manner,

$$u_2 = L_t^{-1}L_x u_1 - \pi^2 L_t^{-1}u_1 - 2L_t^{-1}u_0 u_1 \tag{101}$$

Table 6. Error Obtained Using Decomposition Method with Three Iterations for the Klein–Gordon Eq. (97)

x	$t = 0.1$	$t = 0.3$	$t = 0.5$
0.1	7.3×10^{-7}	5.2×10^{-4}	1.1×10^{-2}
0.3	7.0×10^{-7}	5.0×10^{-4}	1.0×10^{-2}
0.5	6.0×10^{-7}	4.2×10^{-4}	8.7×10^{-3}

Table 6 shows the error obtained by comparing the decomposition method with three iterations and the exact solution which is $u = x^2 \cos \pi t$.

Many other interesting physical problems whose mathematical formulation lead to nonlinear equations can be handled by the decomposition method. Some open problems related to this decomposition method include an extension of the method to solve nonlinear equations with specified behavior at ∞ and nonlinear equations with fractional exponents.

BIBLIOGRAPHY

1. G. Adomian, Convergent series solution of nonlinear equations, *Comput. and App. Math.,* **11** (2): 225–230, 1984.
2. G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method,* Kluwer, 1994.
3. G. Adomian and R. Rach, Analytic solution of nonlinear boundary-value problems in several dimensions by decomposition, *J. Math. Anal. Appl.,* **174**: 118–137, 1993.
4. G. Adomian and R. Rach, Noise terms in decomposition solution series, *Computers Math. Applic.,* **24** (11): 61–64, 1992.
5. G. Adomian, A review of the decomposition method and some recent results for nonlinear equations, *Computers Math. Applic.,* **21** (5): 101–127, 1991.
6. G. Adomian, The Sine–Gordon, Klein–Gordon, and Korteweg–De Vries equations. *Computers Math. Applic.,* **21** (5): 133–136, 1991.
7. G. Adomian and R. Rach, Equality of partial solutions in the decomposition method for linear or nonlinear partial differential equations, *Computers Math. Applic.,* **10** (12): 9–12, 1990.
8. G. Adomian, *Nonlinear Stochastic Systems Theory and Applications to Physics,* Kluwer, 1989.
9. G. Adomian, Stochastic water reservoir modeling, *J. Math. Anal. Appl.,* **115**: 233–234, 1986.
10. G. Adomian, A new approach to the heat equation—An application of the decomposition method, *J. Math. Anal. Appl.,* **113**: 202–209, 1986.
11. G. Adomian, A new approach to nonlinear partial differential equations, *J. Math. Anal. Appl.,* **102**: 420–434, 1984.
12. R. E. Bellman and G. Adomian, *Partial Differential Equations—Methods for their Treatment and Solutions,* Reidel, Dordrecht, 1984.
13. B. K. Datta, A technique for approximate solutions to Schrödinger-Like equations, *Computers Math. Applic.,* **20** (1): 61–65, 1990.
14. B. K. Datta, A new approach to the wave equation, an application of the decomposition method, *J. Math. Anal. Appl.,* **142**: 6–12, 1989.
15. L. Gabet, The theoretical foundation of the Adomian method, *Computers Math. Applic.,* **27** (12): 41–52, 1994.
16. E. Y. Deeba and S. A. Khuri, The decomposition method applied to Chandrasekhar H-equation, *Applied Mathematics and Computation,* **77**: 67–78, 1996.
17. E. Y. Deeba and S. A. Khuri, A decomposition method for solving the nonlinear Klein–Gordon equation, *Journal of Computational Physics,* **124**: 442–448, 1996.
18. S. Chandrasekhar, *Radiative Transfer,* New York: Dover, 1960.
19. Y. Cherruault, G. Saccomandi, and B. Some, New results for convergence of Adomian's method applied to integral equations, *Mathl. Comput. Modeling,* **16** (2): 85–93, 1992.
20. Y. Cherruault, Convergence of Adomian's method, *Kybernetes,* **18** (2): 31–38, 1989.
21. R. Grimshaw and N. Joshi, Weakly nonlocal solitary waves in a singularly perturbed Korteweg–de Vries equation, *SIAM J. Math. Anal.,* **5** (1): 124–135, 1995.
22. S. Kichenassamy and P. J. Olver, Existence and nonexistence of solitary waves solitons to higher order model evolution equations, *SIAM J. Math. Anal.,* **23** (5): 1141–1166, 1992.
23. Y. Pomeau, A. Ramani, and B. Grammaticos, Structural stability of the Korteweg–de Vries solitons under a singular perturbation, *Physica D,* **31**: 127–134, 1988.
24. S. W. Schoombie, Spline Petrov–Galerkin methods for the numerical solution of the Korteweg–de Vries equation, *IMA J. Numerical Analysis,* **2**: 95–109, 1982.
25. G. B. Whitham, *Linear and Nonlinear Waves,* New York: Wiley, 1974.
26. E. Zauderer, *Partial Differential Equations of Applied Mathematics,* New York: Wiley, 1983.
27. C. Coray and J. Koebbe, Accuracy optimized methods for constrained numerical solutions of hyperbolic conservation laws, *J. of Computational Physics,* **109**: 115–132, 1993.
28. P. D. Lax, *Hyperbolic Systems of Conservative Laws and the Mathematical Theory of Shock Waves,* SIAM, Philadelphia, 1973.
29. R. J. LeVeque, *Numerical Methods for Conservative Laws,* Basel: Birkhauser, 1990.
30. P. Kuo and L. Vazquez, *J. Appl. Sci.,* **1**: 25, 1983.
31. J. L. Lions, *Quelques Méthods de Résolution des Problèmes aux Limites Non Linéaires,* Paris: Dunod, 1969.
32. Wei-Ming and Ben-Yu Guo, Fourier Collocation method for solving nonlinear Klein–Gordon equation. *J. Computational Physics,* **108**: 296–305, 1993.
33. W. Strauss and L. Vazquez, Numerical solutions of a nonlinear Klein–Gordon equation, *J. Computational Physics,* **28**: 271–278, 1978.

E. Y. DEEBA
S. A. KHURI
University of Houston–Downtown

NONLINEAR EQUATIONS. See NONLINEAR EQUATIONS.

NONLINEAR FILTERING. See FILTERING AND ESTIMATION, NONLINEAR.

Number Theory
Standard Article
Wayne Patterson[1]
[1]Howard University and the National Science Foundation, Washington, DC
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2442
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (974K)

Abstract

The sections in this article are

Divisibility

Multiplicative Functions

Congruence

Quadratic Residues

Sums of Squares

Continued Fractions

Algebraic and Transcendental Numbers

Partitions

Prime Numbers

Diophantine Equations

Elliptic Curves

Applications

| | |

Copyright © 1999-2008 All Rights Reserved.

NUMBER THEORY

In the contemporary study of mathematics, number theory stands out as a peculiar branch for many reasons. Most of the development of mathematical thought is concerned with the identification of certain structures and relations in these structures. For example, the study of algebra is concerned with different types of operators on objects, such as the addition and multiplication of numbers or the permutation of objects or the transformation of geometric objects—and the study of algebra is concerned with the classification of the many such types of operators.

Similarly, the study of analysis is concerned with the properties of operators that satisfy conditions of continuity.

Number theory, however, is the study of the properties of those few systems that arise naturally, beginning with the natural numbers (which we shall usually denote $\mathbb{N}$), progressing to the integers ($\mathbb{Z}$), the rationals ($\mathbb{Q}$), the reals ($\mathbb{R}$), and the complex numbers ($\mathbb{C}$). Rather than identifying very general principles, number theory is concerned with very specific questions about these few systems.

For that reason, for many centuries, mathematicians thought of number theory as the purest form of inquiry. After all, it wasn't inspired by physics or astronomy or chemistry or other "applied" aspects of the physical universe. Consequently, mathematicians could indulge in number theoretic pursuits while being concerned only with the mathematics itself.

But number theory is a field with great paradoxes. This purest of mathematical disciplines, as we shall see, has served as the source for arguably the most important set of applications of mathematics in many years!

Another curious aspect of number theory is that it is possible for a very novice student of the subject to pose questions that can baffle the greatest minds. One example is the following: It is an interesting observation that there are many triples of natural numbers (in fact, an infinite number) that satisfy the equation $x^2 + y^2 = z^2$. For example, $3^2 + 4^2 = 5^2$; $5^2 + 12^2 = 13^2$; $7^2 + 24^2 = 25^2$; and so on.

However, one might easily be led to the question, can we find (nonzero) natural numbers x, y, and z such that $x^3 + y^3 = z^3$? Or, indeed, such that $x^n + y^n = z^n$, for any natural number $n > 2$ and nonzero integers x, y, and z?

The answer to this simple question was announced by the famous mathematician, Pierre Auguste de Fermat, in his last written work in 1637. Unfortunately, Fermat did not provide a proof but only wrote the announcement in the margins of his writing. Subsequently, this simply stated problem became known as Fermat's Last Theorem, and the answer eluded the mathematics community for 356 years, until 1993, when it was finally solved by Andrew Wiles.

The full proof runs to over one thousand pages of text and involves mathematical techniques drawn from a wide variety of disciplines within mathematics. It is thought to be highly unlikely that Fermat, despite his brilliance, could have understood the true complexity of his Last Theorem. (Lest the reader leave for want of the answer, what Wiles proved is that there are no possible nonzero x, y, z, and $n > 2$ that satisfy the equation.)

Other questions that arise immediately in number theory are even more problematic than Fermat's Last Theorem. For example, a major concern in number theory is the study of prime numbers—those natural numbers that are evenly divisible only by themselves and 1. For example, 2, 3, 5, 7, 11, 13 are prime numbers. Nine, 15, and any even number except 2 are not. (1, by convention, is not considered a prime.)

One can easily observe that small even numbers can be described as the sum of two primes: $2 + 2 = 4$; $3 + 3 = 6$; $3 + 5 = 8$; $3 + 7 = 5 + 5 = 10$; $5 + 7 = 12$; $7 + 7 = 14$; and so on. One could ask, can all even numbers be expressed as the sum of two primes? Unfortunately, no one knows the answer to this question. It is known as the *Goldbach Conjecture,* and fame and fortune await the person who successfully answers the question.

In this article, we will describe some of the principal areas of interest in number theory and then indicate what current research has shown to be an extraordinary application of this purest form of mathematics to several very current and very important applications.

Although this will in no way encompass all the areas of development in number theory, we will introduce:

Divisibility At the heart of number theory is the study of the multiplicative structure of the integers under multiplication. What numbers divide (i.e., are factors of) other numbers? What are all the factors of a given number? Which numbers are prime?

Multiplicative Functions In analyzing the structure of numbers and their factors, we are led to the consideration of functions that are *multiplicative:* In other words, a function is multiplicative if $f(a \times b) = f(a) \times f(b)$ for all a and b.

Congruence Two integers a and b are said to be congruent modulo n (where n is also an integer), and written $a \equiv b \pmod{n}$, if their difference is a multiple of n; alternatively, that a and b yield the same remainder when divided (integer division) by n. The study of the integers under congruence yields many interesting properties and is fundamental to number theory. The modular systems so developed are called modulo n arithmetic and are denoted either $\mathbb{Z}/n\mathbb{Z}$ or $\mathbb{Z}_n$.

Residues In $\mathbb{Z}_n$ systems, solutions of equations (technically, *congruences*) of the form $x^2 \equiv a \pmod{n}$ are often studied. In this instance, if there is a solution for x, a is called a quadratic residue of n. Otherwise, it is called a quadratic nonresidue of n.

Sums of Squares Beyond congruences of the form $x^2 \equiv a \pmod{n}$, we also consider sums of squares, that is equations of the form $x^2 + y^2 = n$.

Partitions Although, as we shall see, there is an essentially unique way of describing a natural number in terms of the ways by which other numbers can multiply together to form the number, the same is not true for addition. The study of partitions is the study of the number of ways a natural number can be reached by summing other natural numbers. This study also introduces some concepts from an-

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 2007 John Wiley & Sons, Inc.

other very vital branch, combinatorial mathematics.

Prime Numbers The prime numbers, with their special property that they have no positive divisors other than themselves and 1, have been of continuing interest to number theorists. In this section we will see, among other things, an estimate of how many prime numbers there are less than some fixed number n.

Continued Fractions The study of the division operations in $\mathbb{Q}$ has led to a branch of number theory studying repeated divisions and their relation to converging and diverging sequences. In general, a continued fraction is a finite or infinite sequence of the form

$$a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{a_3 + \cdots}}}$$

Algebraic and Transcendental Numbers More generally, number theory defines a class of real numbers that are obtainable by the solution of certain forms of polynomial equations. Numbers that can be so obtained (such as $\sqrt{2}$, a solution to $x^2 - 2 = 0$) are called *algebraic numbers*. Numbers that cannot be so determined are called *transcendental numbers*. It has proved to be enormously difficult to prove that certain common numbers, such as π and e, the base of natural logarithms, are transcendental. It is not yet known whether $\pi + e$ is transcendental!

Diophantine Equations The term *Diophantine equation* is used to apply to a family of algebraic equations in a number system such as $\mathbb{Z}$ or $\mathbb{Q}$. A good deal of research on this subject has been directed at polynomial equations with integer or rational coefficients, the most famous of which is the class of equations $x^n + y^n = z^n$, the subject of Fermat's Last Theorem.

Elliptic Curves A final area of discussion in number theory will be the theory of elliptic curves. Although generally beyond the scope of this article, this theory has been so important in contemporary number theory that some discussion of the topic is in order. An elliptic curve represents the set of points in some appropriate number system that are the solutions to an equation of the form $y^2z = x^3 + mxz^2 + nz^3$ when $m, n \in \mathbb{Z}$.

Applications The final section of this article will address several important applications in business, economics, engineering, and computing of number theory. It is remarkable that this, the purest form of mathematics, has found such important applications, often of theory that is hundreds of years old, to very current problems in these fields.

It should perhaps be noted here that many of the results indicated here are given without proof. Indeed, because of space limitations, proofs will be given only when they are especially instructive. A number of references will be given later in which proofs of all the results cited can be found.

DIVISIBILITY

Many of the questions arising in number theory have as their basis the study of the divisibility of the integers. An integer n is *divisible* by k if there exists another integer m, such that $k \times m = n$. We sometimes indicate divisibility of n by k by writing $k|n$, or $k\nmid n$ if n is not divisible by k.

A fundamental result is the *division algorithm.* Given $m, n \in \mathbb{Z}$, with $n > 0$, there exist unique integers c and d such that $m = c \times n + d$ and $0 \leq d < n$.

Equally as fundamental is the Euclidean algorithm.

Theorem 1 Let $m, n \in \mathbb{Z}$, both $m, n \neq 0$. There exists a unique integer c satisfying $c > 0$, $c|m$, $c|n$; and if $d|m$ and $d|n$, then $d|c$.

Proof Consider $\{d|d = am + bn, \forall a, b \in \mathrm{Z}\}$. Let c^* be the smallest natural number in this set. Then c^* satisfies the given conditions.

Clearly $c^* > 0$. $c^*|a$ because, by the division algorithm, there exists s and t such that $a = cs + t$ with $0 \leq t < u$. Thus $a = c^*s + t = ams + bns + t$, thus $a(1 - ms) + b(-ns) = t$. Because $t < c^*$, this implies that $t = 0$, and thus $a = c^*s$ or $c^*|a$. Similarly $c^*|b$. c^* is unique because if c' also meets the conditions then $c^*|c'$ and $c'|c^*$, so $c' = c^*$.

The *greatest common divisor* of two integers m and n is the largest positive integer [denoted GCD(m, n)] such that GCD$(m, n)|m$ and GCD$(m, n)|n$.

Theorem 2 (Greatest Common Divisor) The equation $am + bn = r$ has integer solutions $a, b \Leftrightarrow$ GCD$(m, n)|r$.

Proof Let $r \neq 0$. It is not possible that GCD$(m, n)|r$ since GCD$(m, n)|m$ and GCD$(m, n)|n$, thus GCD$(m, n)|(am + bn) = r$. By Theorem 1 this means that there exists a', b' such that $a'm + b'n =$ GCD(m, n).

Thus $a'' = ra'/\mathrm{GCD}(m, n)$, $b'' = rb'/\mathrm{GCD}(m, n)$ represents an integer solution to $am + bn = r$.

A related concept to the GCD is the LCM, or *least common multiple.*It can be defined as the smallest positive integer that m and n both divide. It is also worth noting that GCD$(m, n) \times$ LCM$(m, n) = m \times n$.

Primes

An integer $p > 1$ is called *prime* if it is divisible only by itself and 1. An integer greater than 1 and not prime is called *composite.* Two numbers m and n with the property that GCD$(m, n) = 1$ are said to *be relatively prime* (or sometimes *coprime*).

Here are two subtle results.

Theorem 3 Every integer greater than 1 is a prime or a product of primes.

Proof Suppose otherwise. Let n be the least integer that is neither; thus n is composite. Thus $n = ab$, and $a, b < n$. Thus a and b are either primes or products of primes, and

thus so is their product n.

Theorem 4 There are infinitely many primes.

Proof (Euclid) If not, let $p_1, \ldots, p_n$ be a list of all the primes, ordered from smallest to largest. Then consider $q = (p_1 p_2 \cdots p_n) + 1$. By the previous statement,

$$q = p'_1 p'_2 \cdots p'_k \tag{1}$$

p'_1 must be one of the $p_1, \ldots, p_n$, say p_j (because these were all the primes); but $p_j | q$, because it has a remainder of 1. This contradiction proves the theorem.

Theorem 5 (Unique Factorization) Every number can be expressed as a product of prime numbers in a unique manner.

Proof Let

$$\begin{aligned} a &= p_1^{\alpha_1} \cdots p_k^{\alpha_k} = P \\ &= q_1^{\beta_1} \cdots q_m^{\beta_m} = Q \end{aligned} \tag{2}$$

For each p_i, $p_i | Q \Rightarrow p_i | q^{\beta}{}_s$ for some s, $1 \leq s \leq m$.

Because p_i and q_s are both primes, $p_i = q_s$. Thus $k = m$, and $Q = p_1^{\beta_1} \cdots p_k^{\beta} k$. We need to show only that the α_i and β_i are equal. Divide both decompositions P and Q by $p_i^{\alpha_i}$. If $\alpha_i \neq \beta_i$, on the one hand one decomposition $a / p_i^{\alpha_i}$ will contain p_i, and the other will not. This contradiction proves the theorem.

What has occupied many number theorists was a quest for a formula that would generate all the infinitely many prime numbers.

For example, Marin Mersenne (1644) examined numbers of the form $M_p = 2^p - 1$, where p is a prime. He discovered that some of these numbers were, in fact, primet. Generally, the numbers he studied are known as Mersenne numbers, and those that are prime are called Mersenne primes. For example, $M_2 = 2^2 - 1 = 3$ is prime; as are $M_3 = 2^3 - 1 = 7$; $M_5 = 2^5 - 1 = 31$; $M_7 = 2^7 - 1 = 127$. Alas, $M_{11} = 2^{11} - 1 = 2047 = 23 \times 89$ is not. Any natural number > 2 ending in an even digit 0, 2, 4, 6, 8 is divisible by 2. Any number > 5 ending in 5 is divisible by 5. There are also convenient tests for divisibility by 3 and 9—if the sum of the digits of a number is divisible by 3 or 9, then so is the number; and by 11—if the sum of the digits in the even decimal places of a number, minus the sum of the digits in the odd decimal places, is divisible by 11, then so is the number.

Several other Mersenne numbers have also been determined to be prime. At the current writing, the list includes 44 numbers:

M_n, where n = 2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203, 2281, 3217, 4253, 4423, 9689, 9941, 11213, 19937, 21701, 23209, 44497, 86243, 110503, 132049, 216091, 756839, 859433, 1257787, 1398269, 2976221, 3021377, 6972593, 13466917, 20996011, 24036583, 25964951, 30402457, 32582657. With current technology, particularly mathematical software packages such as *Mathematica* or *Maple,* many of these computations can be done rather easily. For example, a one-line program in Mathematica, 5.2, executing for 27.2 minutes on a Pentium 4 PC, with 1 GB of RAM and running at 2.8 GHz, was capable of verifying the primality of all M_n up to M_{10000}, and thus determining the first 22 Mersenne primes (up to M_{9941}).

It is not known whether an infinite number of the Mersenne numbers are prime.

It is known that $m^c - 1$ is composite if m ≥ 2 or if c is composite; for if $c = de$ we have

$$(m^e - 1)(m^{e(d-1)} + m^{e(d-1)} + \cdots + m^{e(d-1)} + 1)$$

We can also show that $a^c + 1$ is composite if m is odd, or if c has an odd factor. Certainly if m is odd, m^c is odd, and $m^c + 1$ is even and thus composite. If $c = d(2k + 1)$, then

$$m^c + 1 = (m^d + 1)(m^{2de} - m^{d(2e-1)} + m^{d(2e-2)} - \cdots + 1) \tag{4}$$

and $m^d + 1 > 1$.

Another set of numbers with interesting primality properties are the *Fermat numbers,* $F_n = 2^{2^n} + 1$. Fermat's conjecture was that they were primes. He was able to verify this for $F_1 = 5$, $F_2 = 17$, $F_3 = 257$, and $F_4 = 2^{16} + 1 = 65537$. But then, Euler showed that

Theorem 6 $641 | F_5$.

Proof $641 = 2^4 + 5^4 = 5 \times 2^7 + 1$; thus $2^4 = 641 - 5^4$. Because $2^{32} = 2^4 \times 2^{28} = 641 \times 2^{28} - (5 \times 2^7)^4 = 641 \times 2^{28} - (641 - 1)^4 = 641k - 1$.

To this date, no other Fermat numbers F_n with $n > 4$ have been shown to be prime. It has been determined that the other Fermat numbers through F_{20} are composite.

MULTIPLICATIVE FUNCTIONS

Functions that preserve the multiplicative structure of the number systems studied in number theory are of particular interest, not only intrinsically, but also for their use in determining other relationships among numbers.

A function f defined on the integers, which takes values in a set closed under multiplication, is called a *number-theoretic function.* If the function preserves multiplication for numbers that are relatively prime ($f(m) \times f(n) = f(m \times n)$), it is called a *multiplicative function;* it is called *completely multiplicative* if the restriction GCD(m, n) = 1 can be lifted.

Consider any number theoretic function f, and define a new function F by the sum of the values of f taken over the divisors of n,

$$F(n) = \sum_{d|n} f(d) = \sum_{d|n} f(n/d) \tag{5}$$

the latter being the former sum in reverse order.

Theorem 7 If f is a multiplicative function, then so is F.

Proof Let GCD$(m, n) = 1$; then $d|mn$ can be uniquely expressed as gh where $g|m$, $h|n$, with GCD$(g, h) = 1$.

$$\begin{aligned} F(mn) &= \sum_{d|mn} f(d) = \sum_{g|m}\sum_{h|n} f(gh) = \sum_{g|m}\sum_{h|n} f(g)f(h) \\ &= \sum_{g|m} f(g)f(h_1) + \cdots + \sum_{g|m} f(g)f(h_k) \\ &= F(m)\left(\sum_{h|n} f(h)\right) = F(m)F(n) \end{aligned} \tag{6}$$

Two multiplicative functions of note are the divisor function $\tau(n)$, defined as the number of positive divisors of n; and the function $\sigma(n)$, defined as the sum of the positive divisors of n.

Theorem 8 τ and σ are multiplicative.

Proof Both τ and σ are the "uppercase" functions for the obviously multiplicative functions $1(n) = 1$ for all n, and $i(n) = n$ for all n. In other words,

$$\tau(n) = \sum_{d|n} 1(d) \qquad \text{and} \qquad \sigma(n) = \sum_{d|n} i(d) \tag{7}$$

In order to compute τ and σ, for a prime p note that

$$\tau(p^n) = 1 + n$$

(because the divisors of p^n are $1, p, p^2, \ldots, p^n$); (8)

$$\sigma(p^n) = 1 + p + p^2 + \cdots + p^n = (p^{n+1} - 1)/(p - 1) \tag{9}$$

Thus, for any $n = p_1 a^{n_1} p_2 a^{n_2} \ldots p_k a^{n_k}$,

$$\tau(n) = \prod_{i=1}^{k}(1 + n_i) \qquad \text{and} \quad \sigma(n) = \prod_{i=1}^{k}(p_i^{n_i+1} - 1)/(p_i - 1) \tag{10}$$

In very ancient times, numbers were sometimes considered to have mystical properties. Indeed, the Greeks identified numbers that they called *perfect:* numbers that were exactly the sums of all their proper divisors, in other words, that $\sigma(n) = 2n$. A few examples are $6 = 1 + 2 + 3$; $28 = 1 + 2 + 4 + 7 + 14$; and $496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248$. It is not known whether there are any odd perfect numbers or if an infinite number of perfect numbers exist.

Theorem 9 n is even and perfect $\Leftrightarrow n = 2^{p-1}(2^p - 1)$ where both p and $2^p - 1$ are primes.

In other words, there is one perfect number for each Mersenne prime.

Another multiplicative function of considerable importance is the Möbius function μ.

$\mu(1) = 1$; $\mu(n) = 0$ if n has a square factor; $\mu(p_1 p_2 \cdots p_k) = (-1)^k$ if $p_1, \ldots, p_k$ are distinct primes.

Theorem 10 μ is multiplicative, and the "uppercase" function is 0 unless $n = 1$, when it takes the value 1.

Theorem 11 (Möbius Inversion Formula) If f is a number theoretic function and $F(n) = \Sigma_{d|n} f(d)$, then

$$f(n) = \sum_{d|n} F(d)\mu(n/d) = \sum_{d|n} F(n/d)\mu(n) \tag{11}$$

Proof

$$\sum_{d|n} \mu(d)F(n/d)$$

$$= \sum_{d_1 d_2 = n} \mu(d_1)F(n/d_2) \quad \text{(taking pairs } d_1 d_2 = n\text{)} \tag{12}$$

$$= \sum_{d_1 d_2 = n} \left[\mu(d_1)\sum_{d|d_2} f(d)\right] \quad \text{(definition of } F\text{)} \tag{13}$$

$$= \sum_{d_1 d|n} \mu(d_1)f(d) \quad \text{(multiplying the terms in brackets)} \tag{14}$$

$$= \sum_{d|n} f(d) \sum_{d_1|(n/d)} \mu(d_1) \quad \text{[collecting multiples of } f(d)\text{]}$$

$$= f(n) \tag{15}$$

Theorem 12 If F is a multiplicative function, then so is f.

A final multiplicative function of note is the *Euler function,* $\phi(n)$. It is defined for n to be the number of numbers less than n and relatively prime to n. For primes p, $\phi(p) = p - 1$.

Theorem 13 ϕ is multiplicative.

CONGRUENCE

The study of congruence leads to the definition of new algebraic systems derived from the integers. These systems, called *residue systems,* are interesting in and of themselves, but they also have properties that allow for important applications to be discussed later.

Consider integers a, b, n with $n > 0$. Note that a and b could be positive or negative. We will say that a *is congruent to* b, *modulo* n [written $a \equiv b \pmod{n}$] $\Leftrightarrow n|(a - b)$. Alternatively, a and b yield the same remainder when divided by n. In such a congruence, n is called the modulus, and b is called a residue of a.

An algebraic system can be defined by considering classses of all numbers satisfying a congruence with fixed modulus. It is observed that congruence is an equivalence relation and that the definition of addition and multiplication of integers can be extended to the equivalence classes. Thus, for example, in the system with modulus 5 (also called mod 5 arithmetic), the equivalence classes are $\{\ldots, -10, -5, 0, 5, 10, \ldots\}$, $\{\ldots, -9, -4, 1, 6, 11, \ldots\}$, $\{\ldots, -8, -3, 2, 7, 12, \ldots\}$, $\{\ldots, -7, -2, 3, 8, 13, \ldots\}$, and $\{\ldots, -6, -1, 4, 9, 14, \ldots\}$. It is customary to denote the class by the (unique) representative of the class between 0 and $n - 1$. Thus the five classes in mod 5 arithmetic are denoted 0, 1, 2, 3, 4. Formally, the mod n system can be defined as the algebraic quotient of the integers $\mathbb{Z}$ and the subring defined

by the multiples of $n(n\mathbb{Z})$. Thus the mod n system is often written $\mathbb{Z}/n\mathbb{Z}$. An alternative, and more compact notation, is $\mathbb{Z}_n$.

As mentioned earlier, addition and multiplication are defined naturally in $\mathbb{Z}_n$. Under addition, every $\mathbb{Z}_n$ forms an Abelian group [that is, the addition operation is closed, associative, and commutative; 0 is an identity; and each element has an additive inverse—for any a, $b = n - a$ always yields $a + b \equiv 0 \pmod{n}$].

In the multiplicative structure, however, only the closure, associativity, commutativity, and identity (1) are ensured. It is not necessarily the case that each element will have an inverse. In other words, the congruence $ax \equiv 1 \pmod{n}$ will not always have a solution.

Technically, an algebraic system with the properties described previously is called a *commutative ring with identity.* If it is also known that each (nonzero) element of $\mathbb{Z}_n$ has an inverse, the system would be called a *field.*

Theorem 14 Let a, b, n be integers with $n > 0$. Then $ax \equiv 1 \pmod{n}$ has a solution $\Leftrightarrow \mathrm{GCD}(a, n) = 1$. If x_0 is a solution, then there are exactly $\mathrm{GCD}(a, n)$ solutions given by $\{x_0, x_0 + n/\mathrm{GCD}(a, n), x_0 + 2n/\mathrm{GCD}(a, n), \ldots, x_0 + (\mathrm{GCD}(a, n) - 1)n/\mathrm{GCD}(a, n)\}$.

Proof This theorem is a restatement of Theorem 2.

In order for an element a in $\mathbb{Z}_n$ to have an inverse, alternatively to be a unit, by Theorem 14, it is necessary and sufficient for a and n to be relatively prime [i.e., $\mathrm{GCD}(a, n) = 1$]. Thus, by the earlier definition of the Euler function, the number of units in $\mathbb{Z}_n$ is $\phi(n)$.

The set of units in $\mathbb{Z}_n$ is denoted $(\mathbb{Z}_n)^*$, and it is easily verified that this set forms an Abelian group under multiplication. As an example, consider $\mathbb{Z}_{12}$ or $\mathbb{Z}/12\mathbb{Z}$. Note that $(\mathbb{Z}_{12})^* = \{1, 5, 7, 11\}$, and that each element is its own inverse: $1 \times 1 \equiv 5 \times 5 \equiv 7 \times 7 \equiv 11 \times 11 \equiv 1 \pmod{12}$. Furthermore, closure is observed because $5 \times 7 \equiv 11$, $5 \times 11 \equiv 7$, and $7 \times 11 \equiv 5 \pmod{12}$.

Theorem 15 If p is a prime number, then $\mathbb{Z}_p$ is a field with p elements. If n is composite, $\mathbb{Z}_n$ is not a field.

Proof If p is a prime number, every element $a \in (\mathbb{Z}_p)^*$ is relatively prime to p, that is $\mathrm{GCD}(a, p) = 1$. Thus $ax \equiv 1 \pmod{p}$ always has a solution. Because every element in $(\mathbb{Z}_p)^*$ has an inverse, $(\mathbb{Z}_p)^*$ is a field. If n is composite, there are integers $1 < k$, $1 < n$ such that $k1 = n$. Thus $k1 \equiv 0 \pmod{n}$, and so it is impossible that k could have an inverse. Otherwise, $\mathrm{l} \equiv (k^{-1}k) \times l \equiv k^{-1}(\mathrm{k} \times l) \equiv k^{-1} \times 0 \equiv 0 \pmod{n}$, which contradicts the assumption that $1 < n$.

One of the most important results of elementary number theory is the so-called *Chinese Remainder Theorem.* It is given this name because a version was originally derived by the Chinese mathematician Sun Tzu in the third century. The Chinese Remainder Theorem establishes a method of solving simultaneously a system of linear congruences in several modular systems.

Although there is a long history of the use of this term, it is perhaps a disservice to the discoverer not to coll it the Sun Tzu Theorem. After all, we do not call Fermat's Last Theorem the "French Last Theorem."

Theorem 16 (Chinese Remainder) Given a system $x \equiv a_i \pmod{n_i}$, $\mathrm{i} = 1, 2, \ldots, m$. Suppose that for all $i \neq j$, $\mathrm{GCD}(n_i, n_j) = 1$. Then there is a unique common solution modulo $n = n_1 n_2 \cdots n_m$.

Proof (by construction) Let $n'_i = n/n_i$, $i = 1, \ldots, m$. Note that $\mathrm{GCD}(n_i, n'_i) = 1$. Thus there exists an integer n''_i such that $n'_i n''_i \equiv 1 \pmod{n_i}$. Then

$$x \equiv a_1 n'_1 n''_1 + a_2 n'_2 n''_2 + \cdots + a_m n'_m n''_m \pmod{n} \qquad (16)$$

is the solution. Because $n_i | n'_j$ if $i \neq j$, $x \equiv a_i n'_i n''_i \equiv a_i \pmod{n_i}$. The solution is also unique. If both x and y are common solutions, then $x - y \equiv 0 \pmod{n}$.

An interesting consequence of this theorem is that there is a 1–1 correspondence, preserved by addition and multiplication, of integers modulo n, and m-tuples of integers modulo n_i. Consider $\{n_1, n_2, n_3, n_4\} = \{7, 11, 13, 17\}$, and $n = 17017$. Then

$$\begin{aligned} 95 &\rightarrow (a_1, a_2, a_3, a_4) \\ &= (95 \bmod 7, 95 \bmod 11, 95 \bmod 13, 95 \bmod 17) \\ &= (4, 7, 4, 10) \end{aligned}$$

also

$$162 \rightarrow (1, 8, 6, 9)$$

Performing addition and multiplication tuple-wise:

$$\begin{aligned} &(4, 7, 4, 10) + (1, 8, 6, 9) \\ &\quad = (5 \bmod 7, 15 \bmod 11, 10 \bmod 13, 19 \bmod 17) \\ &\quad = (5, 4, 10, 2) \\ &(4, 7, 4, 10) \times (1, 8, 6, 9) \\ &\quad = (4 \bmod 7, 56 \bmod 11, 24 \bmod 13, 19 \bmod 17) \\ &\quad = (4, 1, 11, 5) \end{aligned}$$

Now verify that $95 + 162 = 257$ and $95 \times 162 = 15390$ are represented by $(5, 4, 10, 2)$ and $(4, 1, 11, 5)$ by reducing each number mod $n_1, \ldots, n_4$.

Another series of important results involving the products of elements in modular systems are the theorems of Euler, Fermat, and Wilson. Fermat's Theorem, although extremely important, is very easily proved—thus it is sometimes called the "Little Fermat Theorem" in contrast to the famous Fermat's Last Theorem described earlier.

Theorem 17 (Euler) If $\mathrm{GCD}(a, n) = 1$, then $a^{\phi(n)} \equiv 1 \pmod{n}$.

Theorem 18 (Fermat) If p is prime, then $a^p \equiv a \pmod{p}$.

Proof (of Euler's Theorem) Suppose that $A = \{a_1, \ldots, a_{\phi(n)}\}$ is a list of the set of units in $\mathbb{Z}_n$. By definition, each of the a_i has an inverse, $a^{-1}{}_i$. Now consider the product, $b = a_1 a_2 \cdots a_{\phi(n)}$. It also has an inverse, in particular b^{-1}

$= a^{-1}{}_1 a^{-1}{}_2 \cdots a^{-1}{}_{\phi(n)}$. Choose any of the units—suppose it is a. Now consider the set $A' = \{aa_1, aa_2, aa_{\phi(n)}\}$. We need to show that as a set, $A' = A$. It is sufficient to show that the aa_i are all distinct. Because there are $\phi(n)$ of them, and they are all units, they represent all of the elements of A.

Suppose that $aa_i \equiv aa_j$ for some $i \neq j$. Then, because a is a unit, we can multiply by a^{-1}, yielding $(a^{-1}a)a_i \equiv (a^{-1}a)a_j$ or $a_i \equiv a_j \pmod{n}$, which is a contradiction. Thus the aa_i are all distinct, and $A = A'$.

Now compute

$$\prod_{i=1}^{\phi(n)} (aa_i) \equiv b \pmod{n} \qquad \text{because } A = A' \tag{17}$$

$$\begin{aligned} a^{\phi(n)} b &\equiv b \pmod{n} \\ a^{\phi(n)} b b^{-1} &\equiv b b^{-1} \pmod{n} \quad \text{multiplying by } b^{-1} \\ a^{\phi(n)} &\equiv 1 \pmod{n} \end{aligned} \tag{18}$$

Although the Chinese Remainder Theorem gives a solution for linear congruences, we would also like to consider nonlinear or higher degree polynomial congruences. In the case of polynomials in one variable, in the most general case,

$$f(x) \equiv 0 \pmod{n} \tag{19}$$

If $n = p_1^{a_1} \cdots p_k^{a_k}$ is the factorization of n, then the Chinese Remainder Theorem ensures solutions of Eq. (24) $\Leftrightarrow$ each of

$$f(x) \equiv 0 \pmod{p_i^{a_i}} \qquad i = 1, 2, \ldots, k \tag{20}$$

has a solution.

Since $\phi(p) = p - 1$ for p a prime, Fermat's Theorem is a direct consequence of Euler's Theorem.

Solutions to Eq. (25) can be found by using a procedure to find solutions to $f(x) \equiv 0 \pmod{p^k}$.

Suppose that x_0 is a solution to $f(x) \equiv 0 \pmod{p^k}$. Then compute a Taylor expansion (using formal derivatives):

$$f(x_0 + tp^k) = f(x_0) + tp^k f'(x_0) + \cdots + 1/n![(tp^k)n] f^{(n)}(x_0) \tag{21}$$

where n is the degree of f and t is an integer. In mod $p^{k+1}(\mathbb{Z}_{p^{k+1}})$, all the terms after the second vanish. Thus $x_0 + tp^k$ is a solution if and only if

$$tf'(x_0) \equiv -f(x_0)/p^k \pmod{p} \tag{22}$$

By Theorem 14, if $p+f'_-(x_0)$, Eq. (27) has a unique solution, and so the solution x_0 of Eq. (27) gives rise to a unique solution $x_0 + tp^k$ of Eq. (26). If $p|f'(x_0)$, then $f(x_0 + tp^k) \equiv f(x_0) \pmod{p^{k+1}}$; hence, either x_0 is also a solution of Eq. (26) in which case so is $x_0 + tp^k$ for all t, or x_0 is not a solution in which case Eq. (26) has no solution satisfying $x \equiv x_0 \pmod{p^{k+1}}$.

Example Consider the congruence $f(x) = x^5 + 100x^4 + 112x^3 + 31x^2 + 67x + 64 \equiv 0 \pmod{125}$. Note that $f'(x) = 5x^4 + 400x^3 + 336x^2 + 62x + 67$. By the method described in the previous paragraph, we first examine the case $p = 5$.

Note that $f(1) \equiv 0$; $f(2) \equiv 0$; and $f(4) \equiv 0 \pmod{5}$. Thus we consider these for the case $p^2 = 25$.

- $x = 1$: $f(1) \equiv 0 \pmod{25}$, and $f'(1) \equiv 20$; so 1 is a solution mod 25. Therefore, so are $1 + 5n \pmod{25}$, that is, 6, 11, 16, 21.
- $x = 2$: $f(2) \equiv 0$, and $f'(1) \equiv 15$; so 2 is a solution mod 25. Therefore, so are $2 + 5n \pmod{25}$, that is, 7, 12, 17, 22.
- $x = 4$: $f(4) \equiv 20 \not\equiv 0 \pmod{25}$, and $f'(4) \equiv 21$; thus compute $21t \equiv -20/5 \pmod{25} \Rightarrow t = 1$. Therefore $4 + tp = 4 + 5 = 9$ is a solution,

Now finally consider $p^3 = 125$. The results are summarized in Table 1. Thus there are 21 roots of the congruence in all.

There are other important results in the theory of polynomial congruences.

Theorem 19 (Lagrange) If $f(x)$ is a nonzero polynomial of degree n, whose coefficients are elements of $\mathbb{Z}_p$ for a prime p, then $f(x)$ cannot have more than n roots.

Theorem 20 (Chevalley) If $f(x_1, \ldots, x_n)$ is a polynomial with degree less than n, and if the congruence

$$f(x_1, x_2, \ldots, x_n) \equiv 0 \pmod{p}$$

has either zero or at least two solutions.

The Lagrange Theorem can be used to demonstrate the result of Wilson noted earlier.

Theorem 21 (Wilson) If p is a prime then $(p - 1)! \equiv -1 \pmod{p}$.

Proof If $p = 2$, the result is obvious. For p an odd prime, let

$$f(x) = x^{p-1} - (x - 1)(x - 2) \cdots (x - p + 1) - 1$$

Consider any number $1 \leq k \leq p - 1$. Substituting k for x causes the term $(x - 1)(x - 2) \cdots (x - p + 1)$ to vanish; also, by Fermat's theorem, $k^{p-1} \equiv 1 \pmod{p}$. Thus, $f(k) \equiv 0 \pmod{p}$. But k has degree less than $p - 1$; and so by the Lagrange Theorem, $f(x)$ must be identically zero, which means that all the coefficients must be divisible by p. The constant coefficient is

$$-1 - (p - 1)! \pmod{p}$$

and thus

$$(p - 1)! \equiv -1 \pmod{p}$$

QUADRATIC RESIDUES

Having considered general polynomial congruences, now we restrict consideration to quadratics. The study of quadratic residues leads to some useful techniques; additionally, they have important and perhaps surprising results.

Table 1

$f(x)$	Root?	$f'(x)$	More Roots?	Associated Roots ($x - 25n$)
1	Yes	$\not\equiv 0$	Yes	1, 26, 51, 76, 101
2	No			
6	No			
7	Yes	$\not\equiv 0$	Yes	7, 32, 57, 82, 107
9	No	$\not\equiv 0$	Yes	solve $121t = -100/25 \pmod 5$, giving 34
11	No			
12	No			
16	Yes	$\not\equiv 0$	Yes	16, 41, 66, 91, 116
17	Yes	$\not\equiv 0$	Yes	17, 42, 67, 92, 117
21	No			
22	No			

The most general quadratic congruence (in one variable) is of the form $ax^2 + bx + c \equiv 0 \pmod m$. Such a congruence can always be reduced to a simpler form. For example, as indicated in the previous section, by the Chinese Remainder Theorem, we can assume that the modulus is a prime power. Also because in the case $p = 2$ we can easily enumerate the solutions, we will henceforth consider only odd primes. Finally, we can use the technique of "completing the square" from elementary algebra to transform the general quadratic to one of the form $x^2 \equiv a \pmod p$.

If $p+a$, then if $x^2 \equiv a \pmod p$ is soluble, a is called a *quadratic residue mod p;* if not, a is called a *quadratic nonresidue mod p.*

Theorem 22 Exactly half of the integers a, $1 \le a \le p - 1$, are quadratic residues mod p.

Proof Consider the set $\mathrm{QR} = \{1^2, 2^2, \ldots, ((p-1)/2)^2\}$. Each of these is a quadratic residue, and no two are congruent mod p. Because $t^2 - u^2 \equiv 0 \pmod p \Rightarrow t + u \equiv 0 \pmod p$ or $t - u \equiv 0 \pmod p$ and t and u are distinct, the second case is not possible; and since t and u must be both $< p - 1$, neither is the first case. Thus there are at least $(p-1)/2$ quadratic residues.

If x_0 solves $x^2 \equiv a \pmod p$, then so does $p - x_0 = p^2 - 2px_0 + x^2{}_0 \equiv x^2{}_0 \equiv a \pmod p$, and $p - x_0 \not\equiv x_0 \pmod p$. Thus we have found $p - 1$ elements of $\mathbb{Z}_p$ that square to elements of QR. Thus, there can be no more quadratic residues outside of QR, so the result is proved.

An important number theoretic function to evaluate quadratic residues is the *Legendre symbol.* For p an odd prime, the Legendre symbol for a

$$\left(\text{written alternatively } (a/p) \text{ or } \left(\frac{a}{p}\right)\right)$$

is

$$\left(\frac{a}{p}\right) = \begin{cases} +1 & \text{if } a \text{ is a quadratic residue mod } p \\ 0 & \text{if } p \mid a \\ -1 & \text{if } a \text{ is a quadratic nonresidue mod } p \end{cases}$$

One method of evaluating the Legendre symbol uses *Euler's criterion.* If p is an odd prime and $\mathrm{GCD}(a, p) = 1$, then

$$\left(\frac{a}{p}\right) = 1 \Leftrightarrow a^{(p-1)/2} \equiv 1 \pmod p$$

Equivalently,

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod p$$

Here are some other charcteristics of the Legendre symbol.

Theorem 23

i. $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$:

ii. if $a \equiv b \pmod p$, then $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$:

iii. $\left(\frac{a^2}{p}\right) = 1, \left(\frac{1}{p}\right) = 1$:

iv. $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}$.

Suppose that we want to solve $x^2 \equiv 518 \pmod{17}$. Then, compute

$$\left(\frac{518}{17}\right) = \left(\frac{8}{17}\right) = \left(\frac{2}{17}\right)$$

But

$$\left(\frac{2}{17}\right) = 1$$

because $6^2 = 36 \equiv 2 \pmod{17}$. Thus, $x^2 \equiv 518 \pmod{17}$ is soluble.

Computation of the Legendre symbol is aided by the following results. First, define an absolute least residue modulo p as the representation of the equivalence class of a mod p, which has the smallest absolute value.

Theorem 24 (Gauss' Lemma) Let $\mathrm{GCD}(a, p) = 1$. If d is the number of elements of $\{a, 2a, \ldots, (p-1)a\}$ whose

absolute least residues modulo p are negative, then

$$\left(\frac{a}{p}\right) = (-1)^d$$

Theorem 25 2 is a quadratic residue (respectively, quadratic nonresidue) of primes of the form $8k \pm 1$ (respectively, $8k \pm 3$). That is,

$$\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8}$$

Theorem 26

1. If $k > 1, p = 4k + 3$, and p is prime, then $2p + 1$ is also prime $\Leftrightarrow 2^p \equiv 1 \pmod{2p+1}$.
2. If $2p + 1$ is prime, then $2p + 1 | M_p$, the pth Mersenne number, and M_p is composite.

A concluding result for the computation of the Legendre symbol is one that, by itself, is one of the most famous—and surprising—results in all of mathematics. It is called *Gauss' Law of Quadratic Reciprocity.* What makes it so astounding is that it manages to relate prime numbers and their residues that seemingly bear no relationship to one another.

Suppose that we have two odd primes p and q. Then, the Law of Quadratic Reciprocity relates the computation of their Legendre symbols; that is, it determines the quadratic residue status of each prime with respect to the other.

The proof, although derived from elementary principles, is long and would not be possible to reproduce here. Several sources for the proof follow.

Theorem 27 (Gauss' Law of Quadratic Reciprocity)

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{(p-1)(q-1)/4}$$

A consequence of the Law of Quadratic Reciprocity follows.

Theorem 28 Let p and q be distinct odd primes, and $a \geq 1$. If $p \equiv \pm q \pmod{4a}$ then

$$\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right)$$

An extension of the Legendre symbol is the *Jacobi symbol.* The Legendre symbol is defined only for primes p. By a natural extension, the Jacobi symbol, also denoted

$$\left(\frac{a}{n}\right)$$

is defined for any $n > 0$, assuming that the prime factorization of n is $p_1 \ldots, p_k$, by

$$\left(\frac{a}{n}\right) \text{[Jacobi symbol]} = \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right)\cdots\left(\frac{a}{p_k}\right) \text{[Legendre symbols]}$$

SUMS OF SQUARES

A next step in the investigation of the properties of numbers is to consider sums of squares. In particular, we will consider a quadratic form $f(x_1, \ldots, x_n) = a$, and look for solutions of this equation.

One technique that can often be employed here is the method of infinite descent. Infinite descent (although in fact a finite process) is a type of mathematical induction.

Given the form $f(x_1, \ldots, x_n) = a$ stated earlier, we seek a solution to

$$f(x_1, \ldots, x_n) = ka \tag{23}$$

for some positive integer k. We would seek this using congruence methods as described earlier. Then, given the solution to Eq. (23), we look for another solution $f(x_1, \ldots, x_n) = k'm$, with $0 < k' < k$. Then, by repeating the process, we may eventually find a solution to Eq. (23).

One of the first results of this technique is the result of Fermat.

Theorem 29 (Fermat) For a prime p, $x^2 + y^2 = p$ has a solution for x and $y \Leftrightarrow p = 2$ or $p \equiv 1 \pmod 4$.

Proof In the case $p = 2$, we have $1^2 + 1^2 = p$.

If $p \equiv 3 \pmod 4$, there can be no solution, because for $q \in \mathbb{Z}_4$, $q^2 \equiv 0$ or $1 \pmod 4$. Thus, we need to verify only the case $p \equiv 1 \pmod 4$. Using the method of infinite descent, by the Legendre symbol there exists an x such that $0 < x < p/2$ and $x^2 + 1 \equiv 0 \pmod p$. Hence, $x^2 + 1 = kp$ for some $k < p$.

Some related results follow.

Theorem 30 The equation $x^2 + y^2 = m$ has an integer solution $\Leftrightarrow$ each prime factor of m congruent to 3 modulo 4 occurs to an even power in the prime factorization of m.

Theorem 31 Let p be a prime. Then

1. $x^2 + 3y^2 = p$
2. $x^2 - xy + y^2 = p$

are both soluble in the integers $\Leftrightarrow p \equiv 1 \pmod 3$ or $p = 3$.

Theorem 2 Every nonnegative integer can be expressed as a sum of four squares.

Proof First, because of the following identity, it will be sufficient to prove the result for primes, as the product of

two numbers expressible as the sum of four squares is also so expressible:

$$\begin{aligned}(x_1^2+x_2^2+x_3^2+x_4^2)(y_1^2+y_2^2+y_3^2+y_4^2)&\\ =(x_1y_1+x_2y_2+x_3y_3+x_4y_4)^2&\\ =(x_1y_2-x_2y_1+x_3y_4-x_4y_3)^2&\\ =(x_1y_3-x_3y_1+x_4y_2-x_2y_4)^2&\\ =(x_1y_4-x_4y_1+x_2y_3-x_3y_2)^2&\end{aligned} \quad (24)$$

Since the condition is satisfied for 1 and 2 ($1 = 1^2 + 0^2 + 0^2 + 0^2$; $2 = 1^2 + 1^2 + 0^2 + 0^2$), we need only prove it for odd primes. Consider $S = \{a^2 | a = 1, 2, \ldots, (p-1)/2\}$, and also $T = \{-1 - b^2 | b = 0, 1, \ldots, (p-1)/2\}$ for any odd prime p. No two elements of S are congruent mod p (by Theorem 22). The set $S \cup T$ has $p + 1$ elements. Thus there exists $a \in S$, $b \in T$ with $a, b < p/2$ and $a^2 \equiv -1 - b^2 \pmod{p}$. Thus $a^2 + b^2 + 1^2 + 0^2 \equiv 0 \pmod{p}$, and thus

$$x_1^2+x_2^2+x_3^2+x_4^2 = kp \text{ has a solution for some } k < p \quad (25)$$

Now there are two cases to consider. Either k is even or odd. If k is even, then $x_1, \ldots, x_4$ are either all even, all odd, or two even. Thus $(x_1 + x_2)/2$, $(x_1 - x_2)/2$, $(x_3 + x_4)/2$, $(x_3 - x_4)/2$, after relabeling, are all integers. Thus, $[(x_1 + x_2)/2]^2 + [(x_1 - x_2)/2]^2 + [(x_3 + x_4)/2]^2 + [(x_3 - x_4)/2]^2 = kp/2$. Thus, we have successfully applied the principle of infinite descent.

If k is odd, let y_i be the absolute least residue mod k of x_i, so that

$$y_1^2+y_2^2+y_3^2+y_4^2 \equiv x_1^2+x_2^2+x_3^2+x_4^2 \equiv 0 \pmod{k} \text{ [by Eq. (25)]}$$

Thus ${y^2}_1 + {y^2}_2 + {y^2}_3 + {y^2}_4 = k_1 k$, where $k_1 < k$. Let z_1, z_2, z_3, z_4 represent each of the right-hand side terms in Eq. (46). Then,

$$\begin{aligned}z_1^2+z_2^2+z_3^2+z_4^2 &= (x_1^2+x_2^2+x_3^2+x_4^2)(y_1^2+y_2^2+y_3^2+y_4^2)\\ &= k^2k_1p\end{aligned} \quad (26)$$

Also, each $z_i \equiv 0 \pmod{k}$. thus z_1/k, z_2/k, z_3/k, z_4/k are all integers. Thus we have the integer equation

$$(z_1/k)^2 + (z_2/k)^2 + (z_3/k)^2 + (z_4/k)^2 = k_1 p \quad (27)$$

and infinite descent can be applied in this case as well, proving the result.

CONTINUED FRACTIONS

Part of the interest in studying the classical number systems lies in their interrelationships. In particular, because all real numbers can be approximated by limits of sequences of rational numbers, it is natural to consider methods of approximating real numbers through various sequences of rationals.

One type of rational sequence is called a *continued fraction*. A continued fraction is an expression of the form:

$$a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{b_2}{a_3 + \cdots}}} \quad (28)$$

where all the a_i, b_i are integers. If, in particular, the b_is are all equal to 1, and the a_is are all greater than or equal to 1, the continued fraction is called simple. We may also consider such a continued fraction with finitely many entries, with the last being any real number greater than or equal to 1.

An infinite continued fraction

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}} \quad (29)$$

henceforth to be denoted $[a_0, a_1, a_2, a_3, \ldots]$ is said to converge when the sequence $[a_0]$, $[a_0, a_1]$, $[a_0, a_1, a_2]$, ... converges.

One fundamental result follows.

Theorem 33 Let $a_0, a_1, \ldots$ be a finite sequence of $n + 1$ positive integers, or an infinite sequence, except that a_0 can be zero; and let c_i and d_i be given by

$$\begin{aligned} c_0 &= a_0 & d_0 &= 1\\ c_1 &= a_0a_1 + 1 & d_1 &= a_1\\ &\vdots & &\vdots\\ c_{k+2} &= c_{k+1}c_{k+2} = c_k & d_{k+2} &= d_{k+1}a_{k+2} + d_k \end{aligned}$$

where $k \geq n$ if the sequence is finite. If $\alpha \in \mathbb{R}$ is greater than 1, then

i. $[a_0, a_1, \ldots, a_k, \alpha] = \dfrac{\alpha c_k + c_{k-1}}{\alpha d_k + d_{k-1}}$ if $k > 0$

ii. $[a_0, a_1, \ldots, a_k] = c_k/d_k$

Proof By induction. When $k = 1$,

$$a_0 + \frac{1}{a_1 + (1/\alpha)} = \frac{a_0a_1 + (a_0/\alpha) + 1}{a_1 + (1/\alpha)} = \frac{\alpha a_0a_1 + \alpha + a_0}{\alpha a_1 + 1} \quad (30)$$

and

$$\frac{\alpha c_1 + c_0}{\alpha d_1 + d_0} = \frac{\alpha(a_0a_1 + 1) + a_0}{\alpha a_1 + 1} = \frac{\alpha a_0a_1 + \alpha + a_0}{\alpha a_1 + 1} \quad (31)$$

For $k > 1$, $[a_0, \ldots, a_{k+1}, \alpha] = [a_0, \ldots, a_k, a_{k+1} + 1/\alpha]$

$$\begin{aligned}\frac{[a_{k+1} + (1/\alpha)]c_k + c_{k-1}}{[a_{k+1} + (1/\alpha)]d_k + d_{k-1}} &= \frac{a_{k+1}c_k + (1/\alpha)c_k + c_{k-1}}{a_{k+1}d_k + (1/\alpha)d_k + d_{k-1}}\\ &= \frac{\alpha(a_{k+1}c_k + c_{k-1}) + c_k}{\alpha(a_{k+1}d_k + d_{k-1}) + d_k} = \frac{\alpha c_{k+1} + c_k}{\alpha d_{k+1} + d_k}\end{aligned} \quad (32)$$

Theorem 34 For $k > 0$,

1. $c_k d_{k+1} - c_{k+1} d_k = (-1)^{k+1}$
2. $\mathrm{GCD}(c_k, d_k) = 1$
3. if $k > 0$ then $d_{k+1} > d_k$, so $d_k \geq k$
4. $c_0/d_0 < c_2/d_2 < c_{2k}/d_{2k} \cdots < c_{2k+1}/d_{2k+1} < \cdots < c_1/d_1$
5. all simple continued fractions converge.

Proof

1. Using the definition, $c_0d_1 - c_1d_0 = -1$; $c_kd_{k+1} - c_{k+1}d_k = -(c_{k-1}d_k - c_kd_{k-1})$. Use induction to complete the proof.
2. This follows from (i).
3. Again use the definition and induction, since $a_k \geq 1$, $\forall k$.
4. Substitute a_{k+1} for α in Theorem 33, to conclude that $a_{k+1} \geq 1$, c_{k+2}/d_{k+2} lies between c_k/d_k and c_{k+1}/d_{k+1}. But $c_0/d_0 < c_1/d_1$, so $c_0/d_0 < c_2/d_2 < c_1/d_1$. Prove the general result by induction.
5. By (i) and (iv), $\{c_k/d_k\}$ is a Cauchy sequence, and so converges.

Theorem 35 α is a rational number $\Leftrightarrow$ it has a finite continued fraction representation.

Proof Clearly if all the entries of a finite continued fraction are integers, then α is rational. On the other hand, if α is rational and can be expressed as s/t, we have $s/t = q_1 + 1/(t/r_1)$ if $r_1 > 0$, by the Euclidean algorithm and dividing by t. We can continue this process, next dividing (t/r_1), with the next remainder r_2, r_1. The process will eventually terminate, and the result is the desired continued fraction.

Example Consider $\alpha = 326/89$. Then,

$$\frac{326}{89} = 3 + \frac{79}{89} = 3 + \frac{1}{\frac{89}{79}} = 3 + \frac{1}{1 + \frac{1}{\frac{79}{10}}}$$

$$= \cdots = 3 + \cfrac{1}{1 + \cfrac{1}{7 + \cfrac{1}{1 + \frac{1}{9}}}}$$

Theorem 36 Let α be an irrational number. Then

1. $\lim_{n\to\infty} c_k/d_k = \alpha$.
2. $|\alpha - c_k/d_k| < 1/d_kd_{k+1} < 1/d^2{}_k$.

It can also be shown that continued fraction representations are unique.

For example, we can show that

$$e = [2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, 1, \ldots]: \sqrt{2}]$$
$$= [1, 2, 2, 2, 2, \ldots]: \text{and } (1 + \sqrt{5})/2 = [1, 1, 1, 1, \ldots]$$

Two other interesting results follow.

Theorem 37 The continued fraction representation of α is eventually periodic $\Leftrightarrow$ α is a quadratic number, that is it solves a quadratic polynomial equation with rational coefficients.

Theorem 38 (Khinchin) $[a_0, a_1, a_2, \ldots]$ converges $\Leftrightarrow$ $\Sigma^{\infty}{}_{i=0}\, a_i$ diverges.

ALGEBRAIC AND TRANSCENDENTAL NUMBERS

Continuing in this theme, the study of number theory has also been concerned with discovering real (and complex) numbers that satisfy polynomial equations with coefficients that are algebraic numbers.

A complex number $c \in \mathbb{C}$, which is the solution of an equation $q_nc^n + q_{n-1}c^{n-1} + \cdots + q_1c + q_0 = 0$, where $q_0, \ldots, q_n \in \mathbb{Q}$, is called an *algebraic number*. If all the $q_0, \ldots, q_n \in \mathbb{Z}$, then c is called an *algebraic integer*.

Then, numbers that do not satisfy any such polynomial equation are called *transcendental*. In some ways, the transcendental numbers are the most intractable.

There is quite a lengthy history of research in number theory just to establish that several well-known numbers are transcendental. For example, it is known that if $\alpha \neq 0$ or 1 and is an algebraic number, then the following are transcendental:

π, e^{α}, $\sin \alpha$, $\cos \alpha$, $\tan \alpha$, $\sinh \alpha$, $\cosh \alpha$, $\arcsin \alpha$, $\arccos \alpha$, $\ln \alpha$

However, it is unknown as to whether or not $e + \pi$, for example, is transcendental.

A major result in the theory of transcendental numbers is the Gelfond–Schneider theorem.

Theorem 39 (Gelfond–Schneider) If α and β are algebraic numbers, and $\alpha \neq 0$ or 1, and β is irrational, then α^{β} is transcendental.

PARTITIONS

We have been considering various questions arising from the properties of the ordinary arithmetic operators in the classical number systems. Within the natural numbers, another question of interest, for a natural number n, in how many ways can we find sums that will add to n?

Any such sum is called a *partition* of n, and the counting function that determines the number of such partitions is usually called $p(n)$. This function is one that, alas, does not admit to convenient algebraic properties. It is also one that grows very quickly. For example, although $p(5) = 7$, $p(10) = 42$, $p(15) = 176$, and $p(20) = 627$.

For S any subset of $\mathbb{N}$, $S \subseteq \mathbb{N}$, let S' be the set of all partitions with parts only in S; and let S'_m be the set of those partitions in which no part is used more than m times. Further, let N' be the set of all partitions, and N'_1 the set of all partitions with no number repeated. Also, let $p(S', n)$ be the number of partitions of n with all summands in S.

Also, for any infinite sequence $\{a_0, a_1, a_2, \ldots\}$, let the power series $f(q) = \Sigma^{\infty}{}_{i=0}\, a_iq^i$ be called the generating function for the sequence $\{a_0, a_1, a_2, \ldots\}$.

Theorem 40 Let $S \subseteq \mathbb{N}$, $m > 0$, and f and f_m be given by

$$f(q) = \sum_{i=0}^{\infty} p(S', i)q^i: \qquad f_m(q) = \sum_{i=0}^{\infty} p(S'_m, i)q^i$$

Then

$$f(q) = \prod (1 - q^i)^{-1}$$

$$f_m(q) = \prod_{i \in H}^{i \in S} (1 - q^{(m+1i)})/(1 - q^i)$$

Proof We will ignore issues of convergence, which can be demonstrated. Consider

$$\begin{aligned} \prod_{n \in S} (1 - q^n)^{-1} &= \prod_{n \in S} (1 + q^n + q^{2n} + \cdots + q^{kn} + \cdots) \\ &= \sum{}^* q^{x1h1 + \cdots + xkhk} \end{aligned} \tag{33}$$

where $h_1, h_2, \ldots$ is an enumeration of H, and the sum is over all finite sequences of nonnegative x_i. q^n occurs each time $n = x_1h_1 + \cdots + x_kh_k \Rightarrow n$ has a partition in S'. Also each partition of n with parts in S will occur as an exponent in Σ^*. Thus

$$\prod_{i \in S} (1 - q^i)^{-1}: = \sum_{i=0}^{\infty} p(S'_m, i)q^i \tag{34}$$

and similarly for the other result.

Theorem 41 (Euler) Let O denote the set of odd positive integers, then

$$p(O', n) = \mathrm{p}(N'_1, n)$$

Proof

$$\sum_{i=0}^{\infty} p(O', i)q^i = \prod_{i=1}^{\infty} (1 - q^{2i-1})^{-1} \tag{35}$$

and

$$\sum_{i=0}^{\infty} p(N'_1, i)q^i = \prod_{i=1}^{\infty} (1 + q^i) \tag{36}$$

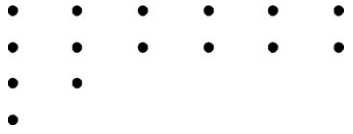
because

$$\prod_{i=1}^{\infty} (1 + q^i) = \prod_{i=1}^{\infty} (1 - q^{2i})/(1 - q^i) = \prod_{i=1}^{\infty} [1/(1 - q^{2i-1})] \tag{37}$$

We have the result.

Graphical Representation

Each partition of a number can be represented by a series of dots representing the summands. For example (ordering the summands in descending order), the partition of 15 given by 6 + 6 + 2 + 1 can be represented by



This dot pattern can be reflected along its diagonal; alternatively, it can be viewed vertically instead of horizontally, giving the partition 4 + 3 + 2 + 2 + 2 + 2. These two partitions are called conjugate.

Theorem 42 The number of partitions of n with at most m parts is equal to the number of partitions of m in which no part exceeds m.

Proof Consider the conjugates.

Next, we cite Euler's pentagonal number theorem, which gives an algorithm for enumerating $p(n)$. [A pentagonal number is one of the form $m(3m + 1)/2$ or $m(3m - 1)/2$.]

Theorem 43 Let $p_1(S, n)$ [respectively, $p_2(H, n)$] denote the number of partitions of n in H with an odd (respectively, even) number of parts. Then,

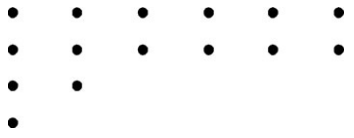
$$p_2(N_1, n) - p_1(N_1, n) = \begin{cases} (-1)^n & \text{if } n = r(3r \pm 1)/2 \\ 0 & \text{otherwise} \end{cases}$$

Proof Let a be a partition of n into r parts, $n = a_1 + \cdots + a_r$, with $a_i > a_{i+1}$ for all i. Let $s(a) = a_r$ (the smallest part); $t(a)$ the largest integer c. $\ni$. $a_1 = a_2 + 1, a_2 = a_3 + 1, \ldots, a_{c-1} = a_c + 1$; and let $t(a) = 1$ if $a_1 \neq a_2 + 1$.

CASE 1 $s(a) \leq t(a)$. Take a, add one to the first $s(a)$ parts, and delete the last part, yielding

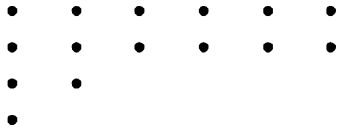
$$(a_1 + 1) + \cdots + (a_{s(a)} + 1) + a_{s(a)+1} + \cdots + a_{r-1} \tag{38}$$

Graphically,



CASE 2 $s(a) > t(a)$. Take a, subtract one from each of the first $t(a)$ parts, and add the new part $t(a)$. Then the new partition is

$$(a_1 + 1) + \cdots + (a_{t(a)} - 1) + a_{t(a)+1} + \cdots + a_r + t(a) \tag{39}$$



Because $a_{t(a)} - 1 > a_{t(a)} + 1$, $a_r > t(a)$, this is a partition into distinct parts.

These two cases provide a one-to-one correspondence between the partitions enumerated by $p_1(N_1, n)$ and $p_2(N_1, n)$, except in the following cases:

1. $s(a) = t(a) = r$ when

$$n = (2r-1) + (2r-2) + \cdots + r = r(3r-1)/2$$

and

2. $s(a) = t(a) + 1 = r + 1$ when

$$n = 2r + (2r-1) + \cdots + (r+1) = r(3r+1)/2$$

Theorem 44 (Euler Pentagonal Number Theorem) For $0 < q < 1$,

$$\prod_{n=1}^{\infty}(1-q^n) = 1 + \sum_{m=1}^{\infty}(-1)^m q^{m(3m-1)/2}(1+q^m) = \sum_{m=-\infty}^{\infty}(-1)^m q^{m(3m-1)/2}$$

Proof

$$\sum_{m=-\infty}^{\infty}(-1)^m q^{m(3m-1)/2}$$

$$= 1 + \sum_{m=1}^{\infty}(-1)^m [q^{m(3m-1)/2} + q^{m(3m+1)/2}] \tag{40}$$

$$= \sum_{n=0}^{\infty}[p_2(S_1', n) - p_1(S_1', n)]q^n$$

Also, as in Theorem 40,

$$\prod_{n=1}^{\infty}(1-q^n) = \sum{}^{*}(-1)^{a1+a2+\cdots} q^{a1+2a2+\cdots} \tag{41}$$

$a_1 + 2a_2 + 3a_3 + \cdots$ is a partition into distinct parts, with an (odd) even number of parts $\Leftrightarrow (-1)a_1 + \cdots + a_k = -1$ (1, respectively); thus combining Eqs. (77) and (41),

$$\prod_{n=1}^{\infty}(1-q^n) = \sum_{n=1}^{\infty}(p_2(S_1', n) - p_1(S_1', n))q^n \tag{42}$$

Theorem 45 If $n > 0$ then

$$p(n) - p(n-1) - p(n-2) + \cdots + (-1)^m p(n - m(3m-1)/2 + (-1)^m p(n - m(3m+1)/2) + \cdots = 0$$

A further extension of the Euler Pentagonal Number Theorem follows.

Theorem 46 Let $N_{j,k}$ denote the set of all partitions with distinct parts in which each part is congruent to $-j$, 0, or j modulo $2k+1$. Then,

$$p_2(N_{j,k}, n) - p_1(N_{j,k}, n) = (-1)^m \begin{cases} \text{for } n = [(2k \pm 1)m(m+1) \pm 2jm]/2 \\ 0 \quad \text{otherwise} \end{cases}$$

One of Ramanujan's remarkable results follows.

Theorem 47 $p(5n+4) \equiv 0 \pmod 5$.

Another important family of identities are the Rogers–Ramanujan identities.

Theorem 48 (Rogers–Ramanujan) The number of partitions of n with minimal difference 2 is equal to the number of partitions of the form $5m+1$ and $5m+4$, equivalently:

$$\prod_{n=0}^{\infty}\frac{1}{(1-x^{5n+1})(1-x^{5n+4})} = \sum_{n=1}^{\infty}\frac{x^{n^2}}{(1-x)(1-x^2)\cdots(1-x^n)}$$

Also, the number of partitions of n with parts not less than 2, and with minimal difference 2, is equal to the number of partitions of the form $5m+2$ and $5m+3$. Alternatively,

$$\prod_{n=0}^{\infty}\frac{1}{(1-x^{5n+2})(1-x^{5n+3})} = \sum_{n=1}^{\infty}\frac{x^{n(n+1)}}{(1-x)(1-x^2)\cdots(1-x^n)}$$

A further estimate on the size of the function $p(n)$ follows.

Theorem 49 For all $n > 1$,

$$2^{\sqrt{n}} < p(n) < e^{\pi\sqrt{(2n/3)}}$$

PRIME NUMBERS

The primes themselves have been a subject of much inquiry in number theory. We have seen earlier that there are an infinite number of primes, and that they are most useful in finite fields from modular arithmetic systems.

One subject of interest has been the development of a function to approximate the frequency of occurrence of primes. This function is usually called $\pi(n)$—it denotes the number of primes less than or equal to n.

In addition to establishing various estimates for $\pi(n)$, concluding with the so-called *Prime Number Theorem,* we will also state a number of famous unproven conjectures involving prime numbers.

An early estimate for $\pi(n)$, by Chebyshev, follows.

Theorem 50 (Chebyshev) If $n > 1$, then $n/(8 \log n) < \pi(n) < 6n/(\log n)$.

The Chebyshev result tells us that, up to a constant factor, the number of primes is of the order of $n/(\log n)$. In addition to the frequency of occurrence of primes, the greatest gap between successive primes is also of interest.

Theorem 51 (Bertrand's Partition) If $n \geq 2$, there is a prime p between n and $2n$.

Two other estimates for series of primes follow.

Theorem 52 $\Sigma_{p\le n} \log p/p = \log n + O(1)$.

Theorem 53 $\Sigma_{p\le n} 1/p = \log\log x + a + O(1/\log x)$.

Another very remarkable result involving the generation of primes is due to Dirichlet.

Theorem 54 (Dirichlet) Let a and b be fixed positive integers such that GCD(a,b) = 1. Then there are an infinite number of primes in the sequence $\{a + bn | n = 1, 2, \dots\}$.

Finally, we have the best known approximation to the number of primes.

Theorem 55 (Prime Number Theorem) $\pi(n) \sim n/(\log n)$.

The conjectures involving prime numbers are legion, and even some of the simplest ones have proven elusive for mathematicians. A few examples are the Goldbach conjecture, the twin primes conjecture, the interval problem, the Dirichlet series problem, and the Riemann hypothesis.

Twin Primes Two primes p and q are called twins if $q = p + 2$. Examples are (p, q) = (5,7); (11, 13); (17, 19); (521, 523). If $\pi_2(n)$ counts the number of twin primes less than n, the twin prime conjecture is that $\pi_2(n) \to \infty$ as $n \to \infty$. It is known, however, that there are infinitely many pairs of numbers (p, q), where p is prime, $q = p + 2$, and q has at most two factors.

Goldbach Conjecture Stated earlier, this conjecture is that every even number is the sum of two primes. What is known is that every large even number can be expressed as $p + q$ where p is prime and q has at most two factors. Also, it is known that every large odd integer is the sum of three primes.

Interval Problems It was stated earlier that there is always a prime number between n and $2n$. It is not known, however, whether the same is true for other intervals, for example such as n^2 and $(n + 1)^2$.

Dirichlet Series In Theorem 54, a series containing an infinite number of primes was demonstrated. It was not known if there are other series that have a greater frequency of prime occurrences, at least until recent research by Friedlander and Iwaniec, who showed that series of the form $\{a^2 + b^4\}$ not only have an infinite number of primes, but also that they occur more rapidly than in the Dirichlet series.

Riemann Hypothesis Although the connection to prime numbers is not immediately apparent, the Riemann hypothesis has been an extremely important pillar in the theory of primes. It states that, for the complex function

$$\zeta(s) = \sum_{n=1}^{\infty} n^{-s} \qquad s = \sigma + it \in \mathbb{C}$$

there are zeros at $s = -2, -4, -6, \dots$, and no more zeros outside of the [critical strip] $0 \le \sigma \le 1$. The Riemann hypothesis states that all zeros of ζ in the critical strip lie on the line $s = 1/2 + it$.

Examples of important number-theoretic problems whose answer depends on the Riemann hypothesis are: (i) the existence of an algorithm to find a nonresidue mod p in polynomial time; (ii) if n is composite, there is at least one b for which neither $b^t \equiv 1 \pmod n$ nor $b^{2^r t} \equiv -1 \pmod n$. This latter is important in algorithms needed to find large primes.

DIOPHANTINE EQUATIONS

The term *Diophantine equation* is used to apply to a family of algebraic equations in a number system such as $\mathbb{Z}$ or $\mathbb{Q}$. To date, we have certainly seen many examples of Diophantine equations. A good deal of research in this subject has been directed at polynomial equations with integer or rational coefficients, the most famous of which being the class of equations $x^n + y^n = z^n$, the subject of Fermat's Last Theorem.

One result in this study, due to Legendre, follows.

Theorem 56 Let $a, b, c \in \mathbb{Z}$ such that (i) $a > 0, b, c < 0$; (ii) a, b, and c are square-free; and (iii) GCD(a,b) = GCD(b,c) = GCD(a,c) = 1. Then

$$ax^2 + by^2 + cz^2 = 0$$

has a nontrivial integer solution $\Leftrightarrow$

$$-ab \in \mathrm{QR}(c)$$
$$-bc \in \mathrm{QR}(a)$$
$$-ca \in \mathrm{QR}(b)$$

Example Consider the equation $3x^2 - 5y^2 - 7z^2 = 0$. With $a = 3$, $b = -5$, and $c = -7$, apply Theorem 56. Note that $ab \equiv 1 \pmod 7$, $ac \equiv 1 \pmod 5$, and $bc \equiv 1 \pmod 3$. Thus, all three products are quadratic residues, and the equation has an integer solution. Indeed, the reader may verify that $x = 3$, $y = 2$, and $z = 1$ is one such solution.

Another result, which, consequently, proves Fermat's Last Theorem in the case $n = 4$, follows. (Incidentally, it has also been long known that Fermat's Last Theorem holds for $n = 3$.)

Theorem 57 $x^4 + y^4 = z^2$ has no nontrivial solutions in the integers.

A final class of Diophantine equations is known generically as Mordell's equation: $y^2 = x^3 + k$. In general, solutions to Mordell's equation in the integers are not known. Two particular solutions follow.

Theorem 58 $y^2 = x^3 + m^2 - jn^2$ has no solution in the integers if

1. $j = 4$, $m \equiv 3 \pmod 4$ and $p \not\equiv 3 \pmod 4$ when $p|n$;

2. $j = 1, m \equiv 2 \pmod 4$, n is odd, and $p \not\equiv 3 \pmod 4$ when $p|n$.

Theorem 59 $y^2 = x^3 + 2a^3 - 3b^2$ has no solution in the integers if $ab \neq 0$, $a \not\equiv 1 \pmod 3$, $3 \nmid b$, a is odd if b is even, and $p = t^2 + 27u^2$ is soluble in integers t and u if $p|a$ and $p \equiv 1 \pmod 3$.

ELLIPTIC CURVES

Many of the recent developments in number theory have come as the by-product of the extensive research done in a branch of mathematics known as elliptic curve theory.

An elliptic curve represents the set of points in some appropriate number system that are the solutions to an equation of the form $y2z = x^3 + mxz^2 + nz^3$ when $m, n \in \mathbb{Z}$.

A major result in this theory is the theorem of Mordell and Weil. If K is any algebraic field, and $C(K)$ are the points with rational coordinates on an elliptic curve, then this object $C(K)$ forms a finitely generated Abelian group.

APPLICATIONS

To this point, all of our discussion has centered around the basic ideas in the development of number theory. We will conclude with three important and recent applications of this very pure mathematical theory in areas of enormous importance for business, government, engineering, and computing. These three areas are: (i) public-key cryptology (for secure computer network development); (ii) digital signatures and authentication (for electronic funds transfer and indeed all electronic communications requiring authentication); and (iii) multiple-radix arithmetic or residue number systems (for signal processing, error correction, and fault tolerance in computer and communications systems design).

Public-Key Cryptology

Despite other applications of number theory that have been discussed in recent years, there can be no doubt that the area of greatest application has been in the field of public-key cryptology.

Cryptology, literally the science of secret writing or code-making and code-breaking, is probably as old as writing itself. For much of its history, cryptology has been the province of the military forces of the world—and mathematical puzzlers. Only since the dawn of the computer era have the techniques involved in cryptology moved from simple mathematics involving permutations to the sophisticated approaches we now see in the computer era.

The fundamental model for any cryptologic system can be described as consisting of M, the message space, or set of finite strings defined over some alphabet; C, the ciphertext space, a set of finite strings over some (possibly different) alphabet; K, the key space, another set of finite strings over a possible third alphabet; and a family of invertible transformations, one for each $k \in K$, $t_k: M \to C$. All the security of the system must lie in the specific choice of key k.

A familiar form of simple cryptosystem is the one whose examples can often be found in daily newspapers under the heading "cryptogram." In this cryptosystem, the alphabet consists of the 26 letters of the Roman alphabet $\Sigma = \{a, b, c, \ldots, z\}$. M, the message space, is the set of all strings over Σ, as is C. Finally, the set $\{t_k\}$ consists of keys which are defined by all of the permutations of the 26 objects which are the symbols of Σ. Thus, $|\{t_k\}| = 26!$

This system is not very secure—otherwise it would not be very appealing as a challenge to daily newspaper readers—and the main technique in breaking this system arises from the knowledge that certain letters in English language text occur far more frequently than others.

A method devised in the early 1970s by Feistel at IBM and code named "Lucifer" evolved into what is now known as the Data Encryption Standard (*DES*). The DES is based on a complicated set of permutations and transpositions. In short, its formal description uses the alphabet $\Sigma = \{0,1\}$; the message space M_{64} consists of 64-bit strings, as does the ciphertext space C_{64}; the key space K_{56} consists of all 56-bit strings; and each t_k, for $k \in K_{56}$, is a composition of 18 transformations, $t_k = \mathrm{IP}^{-1} \circ T_{16} \circ T_{15} \circ \cdots \circ T_1 \circ \mathrm{IP}$, each of which maps 64-bit strings to 64-bit strings; and the individual components T^i depend on the specific choice of $k \in K_{56}$.

Although the DES has been the backbone of commercial data encryption for over 20 years, it has not been without detractors even from the time of its creation and adoption as a national standard.

The Public-Key Paradigm. For any number of reasons, the modern view of cryptology has indicated that the model that we have been using for cryptography has numerous weaknesses.

Historically, cryptology required that both the sending and the receiving parties possessed exactly the same information about the cryptosystem. Consequently, that information that they both must possess must be communicated in some way.

The Key Management Problem. Envision the development of a computer network consisting of one thousand subscribers where each pair of users requires a separate key for private communication. (It might be instructive to think of the complete graph on n vertices, representing the users; with the $n(n-1)/2$ edges corresponding to the need for key exchanges. Thus in the 1000-user network, approximately 500,000 keys must be exchanged in some way, other than by the network!)

In considering this problem, Diffie and Hellman asked the following question: Is it possible to consider that a key might be broken into two parts, $k = (k_p, k_s)$, such that only k_p is necessary for encryption, while the entire key $k = (k_p, k_s)$ would be necessary for decryption?

If it were possible to devise such a cryptosystem, then the following benefits would accrue. First of all, because the information necessary for encryption does not, a priori, provide an attacker with enough information to decrypt, then there is no longer any reason to keep it secret. Consequently, k_p can be made public to all users of the network. A cryptosystem devised in this way is called a public-key

cryptosystem (or *PKC*).

Furthermore, the key distribution problem becomes much more manageable. Consider the hypothetical network of 1000 users, as before. For each user, choose a key $k_i = (k_{pi}, k_{si})$, $i = 1, \ldots, 1000$. In a system-wide public directory, list all of the "public" keys k_{pi}, $i = 1, \ldots, 1000$. Then, to send a message m to user j, select the public key, k_{pi}, and apply the encryption transformation $c = T(k_{pi}, m)$. Send the ciphertext c.

Only user j has the rest of the key necessary to compute $T[(k_p, k_s), c] = m$.

Thus, rather than having to manage the secret distribution of $O(n^2)$ keys in a network of n users, only n keys are required, and they need not be distributed secretly.

Therefore, if we could devise a PKC, it would certainly have most desirable features. But many questions remain to be asked. First of all, can we devise a PKC? What should we look for? Second, if we can find one, will it be secure? Will it be efficient?

In 1978, Rivest, Shamir, and Adelman described a public-key cryptosystem based on principles of number theory, with the security being dependent upon the inherent difficulty of factoring large integers.

Factoring. How hard is factoring numbers in any case? The method most often encountered in elementary courses relies upon generating all the prime numbers up to the square root of the number to be factored, using, for example, the Sieve of Eratosthenes.

If the number we sought to factor contained, let us say, 200 digits, then we would need to be able to generate all the prime numbers of ≤ 100 digits.

Then we would have to test each of these prime numbers for factorization. The direct approach will be $O(n)$, which is infeasible to compute when n is a 200-digit number.

The best general-purpose factoring algorithm is called the *number field sieve*. Its runtime is approximately $O(e^{1.9(\ln n)^{1/3}(\ln \ln n)^{2/3}})$, where n is the size of the number being factored.

In part to maintain momentum in factoring research, the security company RSA Security has issued various monetary challenges for the solution of selected factoring problems. The eight numbers in the challenge range from 174 to 617 decimal digits, and the prizes range from US$10,000 to US$200,000. The first six of the eight challenge numbers have been factored.

Rivest–Shamir–Adelman Algorithm. The basic idea of Rivest, Shamir, and Adelman (*RSA*) was to take two large prime numbers, p and q (for example, p and q each being $\sim 10^{100}$) and to multiply them together to obtain $n = pq$. n is published. Furthermore, two other numbers, d and e, are generated, where d is chosen randomly, but relatively prime to the Euler function, $\phi(n)$, in the interval $[\max(p, q) + 1, n - 1]$. As we have seen, $\phi(n) = (p - 1)(q - 1)$.

Key Generation.

1. Choose two 100-digit prime numbers randomly from the set of all 100-digit prime numbers. Call these p and q.
2. Compute the product $n = pq$.
3. Choose d randomly in the interval $[\max(p, q) + 1, n - 1]$, such that $\mathrm{GCD}[d, \phi(n)] = 1$.
4. Compute $e \equiv d^{-1}$ [modulo $\phi(n)$].
5. Publish n and e. Keep p, q and d secret.

Encryption.

1. Divide the message into blocks such that the bit-string of the message can be viewed as a 200-digit number. Call each block m.
2. Compute and send $c \equiv m^e$ (modulo n).

Decryption. Compute

$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{k\phi(n)+1} \equiv$$
$$m^{k\phi(n)} \times m \equiv 1 \times m \equiv m \quad (\mathrm{modulo}\, n)$$

Note that the result $m^{k\phi(n)} \equiv 1$ used in the preceding line is the Little Fermat Theorem.

Although the proof of the correctness and the security of RSA are established, there are a number of questions about the computational efficiency of RSA that should be raised.

1. Is it possible to find prime numbers of 200 decimal digits in a reasonable period of time?

   The Prime Number Theorem 55 assures us that after a few hundred random selections, we will probably find a prime of 200 digits.

   We can never actually be certain that we have a prime without knowing the answer to the Riemann hypothesis; instead we create a test (the Solovay–Strassen test), which, if the prime candidate passes, we assume the probability that p is not a prime is very low. We choose a number (say 100) numbers a_i at random, which must be relatively prime to p. For each a_i, if the Legendre symbol

$$\left(\frac{a}{p}\right) = a_i^{(p-1)/2} (\mathrm{mod}\ p)$$

   then the chance that p is not a prime and passes the test is 1/2 in each case; if p passes all tests, the chances are $1/2^{100}$ that it is not prime.
2. Is it possible to find an e which is relatively prime to $\phi(n)$?

   Computing the $\mathrm{GCD}[e, \phi(n)]$ is relatively fast, as is computing the inverse of e mod n. Here is an example of a (3×2) array computation which determines both GCD and inverse. In each successive column $(k + 1)$, subtract the largest multiple m of column k less than the first row entry of column $(k - 1)$ to form the new column. When 0 is reached in the first row, both the

inverse (if it exists) and the GCD are found.

m			4	4	1	2	17
$A[1, \bullet]$	1024	243	52	35	17	1	0
$A[2, \bullet]$	1	0	1	4	5	−14	
$A[3, \bullet]$	0	1	−4	17	−21	59	

Once $A[1, \bullet]$ becomes zero, the GCD is at $A[1, \bullet -1]$, and if it is 1, the desired inverse is the value of $A[3, \bullet -1]$, that is 59.

3. Is it possible to perform the computation $e \times d$ where e and d are themselves 200-digit numbers?

   Computing $m^e \pmod n$ consists of repeated multiplications and integer divisions. In Mathematica version 3.0, running on a Pentium machine, such a computation with 400-digit integers was done in 1.59 s.

   One shortcut in computing a large exponent is to use the "fast exponentiation" algorithm. Express the exponent as a binary, $e = b_n b_{n-1} \cdots b_0$. Then compute m^e as follows:

   $\text{ans} = m$

   for $i = n - 1$ to 0 do ans = ans $\times$ ans if $b_i = 1$ then ans = ans $\times$ x end;

   The result is ans. Note that the total number of multiplies is proportional to the log of e.

Example Compute x^{123}.

$$123 = (1111011)_{\text{binary}}$$

So $n = 6$.

$$\text{ans} = x$$
$$i = 5 \quad \text{ans} = \text{ans} \times \text{ans} \times (1 \times x) = x \times x \times x = x^3$$
$$i = 4 \quad \text{ans} = \text{ans} \times \text{ans} \times (1 \times x) = x^3 \times x^3 \times x = x^7$$
$$i = 3 \quad \text{ans} = \text{ans} \times \text{ans} \times (1 \times x) = x^7 \times x^7 \times x = x^{15}$$
$$i = 2 \quad \text{ans} = \text{ans} \times \text{ans} \times (1 \times x) = x^{15} \times x^{15} \times x = x^{31}$$
$$i = 1 \quad \text{ans} = \text{ans} \times \text{ans} = x^{31} \times x^{31} = x^{62}$$
$$i = 0 \quad \text{ans} = \text{ans} \times \text{ans} \times (1 \times x) = x^{62} \times x^{62} \times x = x^{123}$$

In a practical version of the RSA algorithm, it is recommended by Rivest, Shamir, and Adelman that the primes p and q be chosen to be approximately of the same size, and each containing about 100 digits.

The other calculations necessary in the development of an RSA cryptosystem have been shown to be relatively rapid. Except for finding the primes, the key generation consists of two multiplications, two additions, one selection of a random number, and the computation of one inverse modulo another number.

The encryption and decryption each require is at most $2 \log_2 n$ multiplications (in other words, one application of the Fast Exponentiation algorithm) for each message block.

Digital Signatures

It seems likely that, in the future, an application similar to public-key cryptology will be even more widely used. With vastly expanded electronic communications, the requirements for providing a secure way of authenticating an electronic message will be required far more often than the requirement for transmitting information in a secure fashion.

As with public-key cryptology, the principles of number theory have been essential in establishing methods of authentication.

The authentication problem follows. Given a message m, is it possible for a user u to create a "signature," s_u, dependent on some information possessed only by u, so that the recipient of the message (m, s_u) could use some public information for u (a public key), to be able to determine whether or not the message was authentic.

Rivest, Shamir, and Adelman showed that their public-key encryption method could also be used for authentication. However, a number of other authors, particularly El Gamal and Ong–Schnorr–Shamir, developed more efficient solutions to the signature problem. More recently, in 1994, the National Institute for Standards and Technology, an agency of the US government, established such a method as a national standard, now called the *DSS* or Digital Signature Standard.

The DSS specifies an algorithm to be used in cases where a digital authentication of information is required. We assume that a message m is received by a user. The objective is to verify that the message has not been altered and that we can be assured of the originator's identity. The DSS creates for each user a public and a private key. The private key is used to generate signatures, and the public key is used in verifying signatures.

A DSS system begins with

1. The identification of a prime p, with $2^{N-1} < p < 2^N$, for $512 \le N \le 1024$, and N is a multiple of 64.
2. q, a prime number, is chosen, such that $q|(p-1)$, with $2^{159} < q < 2^{160}$.
3. $g \equiv h^{(p-1)/q} \pmod p$, is computed, where h is any integer such that $1 < h < p - 1$ and $h^{(p-1)/q} \pmod p > 1$; that is, g has order $q \bmod p$.
4. x is a randomly or pseudorandomly generated integer with $0 < x < q$.
5. $y \equiv g^x \pmod p$.
6. k is a randomly or pseudorandomly generated integer with $0 < k < g$.

p, q, and g can be public and common to a group of users. A user's private and public keys are x and y, respectively. In addition to x, k is also kept secret.

Signature Generation. For any message m, of arbitrary bit length, the signature of m is a triple, (m, r, s), where

$$r = (g^k \bmod p) \bmod q$$
$$s = k^{-1}[H(m) + xr)] \bmod q$$

where k^{-1} is the inverse of k mod q, and $H(m)$ is a 160-bit string computed by H, a secure hash algorithm. Note that r and s will each be 160 bits. Thus the length of (m, r, s) is 320 bits more than the length of m.

Signature Verification. If the received message is denoted (m', r', s'), then the verification proceeds as follows:

1. If $r' \leq 0$ or $r' \geq q$, then reject.
2. If $s' \leq 0$ or $s' \geq q$, then reject.
3. If (i) and (ii) are satisfied, then compute:

$$w = (s')^{-1} \pmod{q}$$
$$u1 = [H(m')w] \pmod{q}$$
$$u2 = [r'w] \pmod{q}$$
$$v = [(g^{u1}y^{u2}) \bmod p] \bmod q$$

If $v = r'$, then the signature is verified; otherwise, the message should be considered invalid.

Proof of Correctness. If $m = m'$, $r = r'$, and $s = s'$, we need to show that $v = r$. First establish that with p, q, g, h as given,

$$g^q \equiv (h^{(p-1/q)})^q \equiv h^{p-1} \equiv 1 \pmod{p}$$
$$g^{m1} \equiv g^{m2+1q} \equiv g^{m2}g^{1q} \equiv g^{m2} \pmod{p} \text{ by Eq. (43)} \tag{43}$$

Now $y \equiv g^x \pmod{p}$, so by the Eq. (96)

$$v = [(g^{u1}y^{u2}) \bmod p] \bmod q \equiv (g^{H(m)w}y^{rw} \bmod p) \bmod q$$
$$\equiv [(g^{H(m)w}g^{xrw}) \bmod p] \bmod q \equiv (g^{H(m)w+xrw} \bmod p) \bmod q \tag{44}$$

But $s \equiv [k^{-1}(H(m) + xr)] \pmod{q}$, and since $q = s^{-1}$, $w \equiv k[H(m) + xr]^{-1} \pmod{q}$. Then, by substitution in Eq. (97), obtain

$$v = (g^k \bmod p) \bmod q = r, \text{ by definition}$$

Secure Hash Function. The definition of a secure hash function, H, is that it is a function with the following properties:

1. H is a mapping of the set of all bit strings (in the DSS standard, limited to bitstrings of length, 2^{64}) to the set of all bit strings of length 160.
2. For any bitstring b, of length 160, it is computationally infeasible to find a message m such that $H(m) = b$.
3. It is computationally infeasible to find two distinct messages m, m' such that $H(m) = H(\mathrm{m}')$.

The secure hash function most commonly used in DSS is called SHA-1 or secure hash algorithm 1.

Security. Suppose a forger wants to forge a message, that is, alter the value of m, r, or s by the requirements of H, it will not be feasible to find another message m' such that $H(m) = H(m')$. Thus the forger must create an authentic r' and s'. Although p, q, and g are public, and r is transmitted, to solve $[(r = g^{\mathrm{k}}) \bmod p] \bmod q$ is a problem known as the discrete log problem. Its solution is essentially equivalent to the problem of factoring large integers. Finally, because x is chosen at random and is kept secret, s is similarly infeasible to compute.

Despite the widespread use and acceptance of the Digital Signature Standard, in 2005 a number of efforts, led by Xiaoyun Wang, have seriously compromised major components of this United States federal standard. Dr. Wang has found "collisions" in the secure hash algorithm, SHA-1. A collision occurs when two different inputs hash to the same value. This could lead to the possibility of digital forgery. As of this writing, of the various secure Hash algorithm in the standard, SHA-0 is no longer felt to be secure. It may soon be joined by SHA-1. The remaining standards (SHA-224, SHA-256, SHA-384 and SHA-512) are considered secure, but their additional computational overhead would seem to lead to additional cost factors in the use of the DSS.

Multiple-Radix Arithmetic or Residue Number Systems

A final application of elementary number theory is the use of multiple-radix arithmetic or *MRA* systems, which is also known as residue number systems or *RNS*. Recall from the discussion of the Chinese Remainder Theorem 16 that for every product of distinct primes p_{i} (or, more generally, products of distinct numbers that are pairwise relatively prime) there is a one-to-one correspondence that preserves addition and multiplication between the subset of the integers defined by $[0, P-1]$, where $P = \Pi\, p_{\mathrm{i}}$ and the product of rings $\mathbb{Z}_{\mathrm{p1}} \times \cdots \times \mathbb{Z}_{\mathrm{p}n}$, where the operation in this latter system is componentwise.

There are a growing number of applications where the rapid computation of $a \times b \pmod{n}$ or of $a^{\mathrm{b}} \pmod{n}$ is important. In addition to cryptology and signatures as described earlier, this computation is useful in digital signal processing as well.

On the assumption that a and b consist of n decimal digits, the number of one-digit multiplies represented by the normal multiplication algorithm is n^2. Although certain methods, such as the Karatsubo method, can reduce the algorithm to the order of n^{1+k}, for $k < 1$, this is still costly for large n.

Multiplication in an MRA or RNS system is $O(n)$. Consequently, this approach is very attractive in all the areas mentioned previously. Indeed, as we enter an era of more widespread parallel and distributed computing, an even more appealing use of MRA or RNS is to consider distributing large numbers, using their MRA representation, over many processors, with each processor $\mathrm{P_i}$ only required to perform those parts of the computation related to prime p_{i}.

There has been one barrier to the use of this computational method. Although $m \pm n$ and $m \times n$ are computationally efficient in MRA, it was thought that the computation of $m \bmod n$, or integer division of m by n, was essentially as costly as ordinary long division. However, some recent research of Abdelguerfi–Dunham–Patterson, and others has established a fast division algorithm.

To solve problems of computing $A \bmod N$ in an MRA system, assume first that N is fixed. Furthermore, assume

that $N^2 < P$. Then any product AB will be $< P$. We precompute

$$e_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0) \quad f_i = e_i/P \in \mathbb{Q}$$
$$g_i = e_i (\text{mod } N) h_i = g_i/N \in \mathbb{Q}$$

Also, precompute (P mod N)/N.

Let $A \to (a_1, a_2, \ldots, a_m)$ in the MRA system. Then

$$\sum a_i e_i = hP + a \qquad (\in \mathbb{Z})$$
$$\sum a_i f_i = h + (A/P) \quad (\in \mathbb{Q})$$

Therefore, $h = \Sigma\, a_i f_i - (A/P) = \text{floor}\,[\Sigma\, a_i f_i]$, where floor[ ] = greatest integer less than. Also,

$$\sum a_i e_i = hP + A (\in \mathbb{Z}) \Rightarrow$$
$$\sum a_i (e_i \text{mod } N) = h(P \text{ mod } N) + kN + Y \quad (0 \le Y < N) \Rightarrow$$
$$\sum a_i g_i/N = h(P \text{ mod } N)/N + k + Y/N \quad (\in \mathbb{Q})$$

Thus $k = \text{floor}[\Sigma\, a_i h_I - h(P \text{ mod } N) - kN]$.

After computing k, substitute to obtain $Y = \Sigma\, a_i g_i - h(P \text{ mod } N) + kN$ in MRA. Because $Y < N$, and $X \equiv Y \pmod N$, the remainder is Y.

Example Use the system $(p_1, p_2, p_3, p_4) = (7, 11, 13, 17)$; $P = 17017$.

Compute 395 mod 42. In the MRA system, this is (3, 10, 5, 4) mod (0, 9, 3, 8).

$e_1 = 9724$	$e_2 = 12376$	$e_3 = 3927$	$e_4 = 8008$
$f_1 = 0.5714$	$f_2 = 0.7273$	$f_3 = 0.2308$	$f_4 = 0.4706$
$g_1 = 9724 \text{ mod } 42 = 22$	$g_2 = 28$	$g_3 = 21$	$g_4 = 28$
$h_1 = 0.5238$	$h_2 = 0.6667$	$h_3 = 0.5000$	$h_4 = 0.6667$
$P \text{ mod } N = 7$	$(P \text{ mod } N)/N = 0.1667$		

Then, $h = \text{floor}[3 \times 0.5714 + 10 \times 0.7273 + 5 \times 0.2308 + 4 \times 0.4706] = 12$; and $k = \text{floor}[3 \times 0.5238 + 10 \times 0.6667 + 5 \times 0.5000 + 4 \times 0.6667 - 12 \times 0.1667 - Y/N] = 11$; therefore, $3 \times (1,0,9,5) + 10 \times (0,6,2,11) + 5 \times (0,10,8,4) + 4 \times (0,6,2,11) = 12 \times (7,7,7,7) + 11 \times (0,9,3,8) + Y$, and so $Y = (3,6,4,0)$. If one needed the standard form, one could note that Y in standard form is 17, as (17 mod 7, 17 mod 11, 17 mod 13, 17 mod 17) = 3, 6, 4, 0).

BIBLIOGRAPHY

Most of the references in this section discuss most of the topics contained in the main body of this article.

W. W. Adams L. J. Goldstein *Introduction to Number Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1976.

A. Adler J. E. Coury *The Theory of Numbers: A Text and Source Book of Problems*, Boston: Jones and Bartlett, 1995.

W. S. Anglin *The Queen of Mathematics: An Introduction to Number Theory*, Boston: Kluwer, 1995.

A. Baker *A Concise Introduction to the Theory of Numbers*, Cambridge: Cambridge Univ. Press, 1984.

D. M. Burton *Elementary Number Theory*, New York: McGraw-Hill, 1998.

H. Cohen *A Second Course in Number Theory*, New York: Wiley, 1962.

L. E. Dickson *A History of the Theory of Numbers*, 3 Vols, Washington: Carnegie Inst., 1919–1923.

G. H. Hardy E. M. Wright *An Introduction to the Theory of Numbers*, 3rd ed., Oxford: Clarendon Press, 1954.

K. Ireland M. Rosen *A Classical Introduction to Modern Number Theory*, 2nd ed., New York: Springer-Verlag, 1990.

D. E. Flath *Introduction to Number Theory*, New York: Wiley, 1989.

R. K. Guy *Unsolved Problems in Number Theory*, New York: Springer-Verlag, 1994.

N. Koblitz *A Course in Number Theory and Cryptography*, New York: Springer-Verlag, 1994.

I. Niven H. S. Zuckerman *An Introduction to the Theory of Numbers*, 4th ed., New York: Wiley, 1991.

H. E. Rose *A Course in Number Theory*, Oxford: Oxford Univ. Press, 1988.

H. N. Shapiro *Introduction to the Theory of Numbers*, New York: Wiley-Interscience, 1983.

H. Stark *An Introduction to Number Theory*, Chicago: Markham, 1970.

Specialized references on Fermat's Last Theorem, elliptic curve theory, research on Dirichlet-like series, and Mersenne numbers can be found in these references.

C. K. Caldwell The Great Internet Mersenne Prime Search [Online]. Available www:http://www.utm.edu/research/primes/mersenne.shtml

J. Friedlander H. Iwaniec Using a parity-sensitive sieve to count prime values of a polynomial, *Proc. Natl. Acad. Sci. USA*, **94**: 1054–1058, 1997.

Great Internet Mersenne Prime Search, http://www.mersenne.org/.

N. Koblitz *Introduction to Elliptic Curves and Modular Forms*, New York: Springer-Verlag, 1984.

A. Wiles Modular elliptic curves and Fermat's last theorm, *Ann. Math.*, **141** (3): 443–551, 1995.

Mathematical software systems particularly useful in computational number theory are Maple and mathematica.

B. W. Char *et al.* Maple V language reference manual, New York: Springer-Verlag, 1991.

S. Wolfram *Mathematica: A system for doing mathematics by computer*, Redwood City, CA: Addison-Wesley, 1991.

References to Public-key cryptology and factoring include the following.

J. P. Buhler, H. W. Lenstra, and C. Pomerance, *The development of the number field sieve, Volume 1554 of Lecture Notes in Computer Science*, Springer-Verlag, 1994.

J. Buchmann, J. Loho, and J. Zayer, An implementation of the general number field sieve, *Advances in Cryptology - Crypto '93*, Springer-Verlag, 1994. 159–166.

W. Diffie M. E. Hellman New directions in cryptography, *IEEE Trans. Inf. Theory*, **IT-22**: 644–654, 1976.

H. Feistel Cryptography and computer privacy, *Scientific American*, **228** (5): 15–23, 1973.

National Bureau of Standards, Data Encryption Standard, FIPS PUB 46, January 1977.

W. Patterson *Mathematical Cryptology*, Totowa, NJ: Rowman and Littlefield, 1987.

R. L. Rivest A. Shamir L. Adelman A method for obtaining digital signatures and public-key cryptosystems, *Comm. ACM*, **21** (2): 120–126, 1978.

RSA Laboratories, The RSA Factoring Challenge, http://www.rsasecurity.com/rsalabs/node.asp?id=2092.

B. Schneier *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, New York: Wiley, 1994.

J. Seberry J. Pieprzyk *Cryptography: An Introduction to Computer Security*, New York: Prentice-Hall, 1989.

R. Solovay V. Strassen Fast Monte-Carlo tests for primality, *SIAM J. Comput.*, **6** (1): 84–85, 1977.

D. R. Stinson *Cryptography: Theory and and practice*, Boca Raton, FL: CRC Press, 1995.

The following references discuss the issues concerning digital signatures and their standards.

T. ElGamal A public key cryptosystem and a signature scheme based on discrete logarithms, Proc. Crypto 84, New York: Springer, 1985, pp. 10–18.

National Institute for Standards and Technology, Digital Signature Standard (DSS), Federal Inf. Processing Standards Publ. 186, May 19, 1994.

H. Ong C. P. Schnorr A. Shamir An efficient signature scheme based on polynomial equations, Proc. Crypto 84, New York: Springer, 1985, pp. 37–46.

X. Wang, Y. L. Yin and H. Yu, Finding Collisions in the Full SHA-1, *Advances in Cryptology, CRYPTO '05*, Springer-Verlag(2005),pp. 17–36.

And these references address multiple-radix arithmetic and residue number systems.

M. Abdelguerfi A. Dunham W. Patterson MRA: A computational technique for security in high-performance systems, Proc. IFIP/Sec '93, Internation Federation Inf. Processing Soc., World Symp. Comput. Security 1993, 1993, pp. 381–397.

G. Davida B. Litow Fast parallel arithmetic via modular representation, *SIAM J. Comput.*, **20** (4): 756–765, 1991.

M. A. Hitz E. Kaltofen Integer division in residue number systems, *IEEE Trans. Comput.*, **44**: 983–989, 1995.

H. Krishna *et al. Computational Number Theory and Digital Signal Processing: Fast Algorithms and Error Control Techniques*, Boca Raton, FL: CRC Press, 1994.

K. C. Posch R. Posch Modulo reduction in residue number systems, *IEEE Trans. Parallel Distrib. Syst.*, **6**: 449–454, 1995.

WAYNE PATTERSON
Howard University and the National Science Foundation, Washington, DC

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Ordinary Differential Equations
Standard Article

Dan B. Marghitu[1] and S. C. Sinha[1]
[1]Auburn University, Auburn, AL

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2410
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (461K)

Abstract

The sections in this article are

Introduction

First Order Differential Equations

Second Order Differential Equation

Differential Equations of Arbitrary Order

Partial Differential Equations

Applications

| | |

Copyright © 1999-2008 All Rights Reserved.

ORDINARY DIFFERENTIAL EQUATIONS

INTRODUCTION

An *ordinary differential equation* is a relation involving one or several derivatives of a function $y(x)$ with respect to x. The relation may also be composed of constants, given functions of x, or y itself.

The equation

$$y'(x) = e^x, \tag{1}$$

where $y' = dy/dx$, is of a first order ordinary differential equation, the equation

$$y''(x) + 2y(x) = 0, \tag{2}$$

where $y'' = d^2y/dx^2$ is of a second order ordinary differential equation, and the equation

$$2x^2\, y''(x)\, y'(x) + 3e^{-x}\, y''(x) = (x^2+1)y^2(x), \tag{3}$$

where $y''' = d^3y/dx^3$ is a third order ordinary differential equation.

The *order* of an ordinary differential equation the highest derivative of y in the equation.

Definition [1] The *explicit solution* of a first-order differential equation is a function

$$y = g(x), \quad a < x < b, \tag{4}$$

defined and differentiable on (a, b), with the property that the equation becomes an identity when y and y' are replaced by g and g', respectively. The solution of a differential equation $G(x, y) = 0$ it is called the *implicit solution*.

Example. The explicit solution of the first-order differential equation

$$y'(x) = xy(x), \tag{5}$$

is

$$y(x) = ce^{x^2/2}, \tag{6}$$

where c is an arbitrary constant. The differential equation (5) has many solutions. The function (6), with arbitrary c, represents the *general solution* (the totality of all solutions of the equation). If we consider a definite value of c, for example $c = 1$, then the solution obtained $y(x) = e^{x^2/2}$ is called a *particular solution*.

FIRST ORDER DIFFERENTIAL EQUATIONS

Separable Equations

The equation

$$g(y)y' = f(x), \tag{7}$$

or

$$g(y)dy = f(x)dx, \tag{8}$$

is called an *equation with separable variables*, or a *separable equation*. The variable x appears only on the right hand side and the function y appears only on the left hand side in Eq. (8). Integrating both sides we obtain

$$\int g(y)dy = \int f(x)dx + c. \tag{9}$$

If f and g are continuous functions the general solution of Eq. (7) is obtained evaluating Eq. (9).

Example. Solve the equation

$$(y^2+1)xdx + (x+1)ydy = 0.$$

The above equation can be rewritten in the form

$$\frac{x}{x+1}dx + \frac{y}{y^2+1}dy = 0.$$

By integration we obtain

$$x - ln|1+x| + \frac{1}{2}\, ln(1+y^2) = c, \quad x+1 \neq 0.$$

With $x = 0$ and $y = 0$ we calculate $c = \frac{1}{2}\, ln\, 2$ and

$$2x + ln\frac{1+y^2}{2} = ln(1+x)^2, \quad x \neq -1$$

Definition A first-order differential equation together with an *initial condition* is called an *initial value problem*. The initial condition is the condition that at some point $x = x_0$ the solution $y(x)$ has a prescribed value $y(x_0) = y_0$.

Equations Reducible to Separable Form

The first-order differential equation

$$y' = g(\frac{y}{x}), \tag{10}$$

where g is any given function of y/x ($g(x) = f(y/x)$), can be made separable equation by a simple change of variables. The change of variable is

$$\frac{y}{x} = u.$$

The function $y = u\,x$ and by differentiation we obtain

$$y' = u + u'x. \tag{11}$$

Combining the equations (16) and (14), and taking into account that $g(y/x) = g(u)$ we obtain

$$u + u'x = g(u).$$

By separating the variables u and x, the previous equation takes the form

$$\frac{du}{g(u) - u} = \frac{dx}{x}.$$

After integration and replacement of u by y/x the general solution of Eq. (14) is obtained.

Example. Solve the equation

$$\frac{dy}{dx} = \frac{2x+3y}{3x+2y} \quad \text{and} \quad 3x + 2y \neq 0.$$

With the change of function $y = ux$ we obtain

$$u'x + u = \frac{2+3u}{3+2u}$$

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 2007 John Wiley & Sons, Inc.

or

$$u'x = \frac{2-2u^2}{3+2u}$$

and

$$\frac{1}{2}\frac{3+2u}{1-u^2}du = \frac{dx}{x}.$$

Integrating

$$\int \frac{dx}{x} = \frac{1}{2}\int \frac{3+2u}{1-u^2}du$$

or

$$ln|x| = -\frac{1}{2}ln|u^2-1| - \frac{3}{4}ln|\frac{1-u}{1+u}| + ln|c|.$$

We obtain $x^4(u^2-1)^2(\frac{1-u}{1+u})^3 = c$. Replacing u by y/x, the general integral will be

$$(x^2-y^2)^2(x-y)^3 = c(x+y)^3.$$

Exact Differential Equations

A first-order differential equation

$$M(x, y)dx + N(x, y)dy = 0, \tag{12}$$

is *exact* if the left-hand side is an exact differential,

$$d\,u(x, y) = \frac{\partial u}{\partial x}dx + \frac{\partial u}{\partial y}dy. \tag{13}$$

Equation (26) can be rewritten as

$$du = 0.$$

and by integration the general solution is

$$u(x, y) = c. \tag{14}$$

If there is a function $u(x, y)$ with the properties

$$(a)\frac{\partial u}{\partial x} = M, \quad (b)\frac{\partial u}{\partial y} = N, \tag{15}$$

then $M(x,y)\,dx + N(x,y)\,dy = 0$ is an exact differential equation.

The necessary and sufficient condition for $M\,dx + N\,dy$ to be an exact differential [1] is

$$\frac{\partial M}{\partial y} = \frac{\partial N}{\partial x}. \tag{16}$$

To find $u(x, y)$ we have the following steps [1].

From Eq. (30a), if we consider y to be a constant, we obtain

$$u = \int Mdx + k(y), \tag{17}$$

where $k(y)$ is the "constant" of integration. $k(y)$ is determined from Eq. (17) by deriving $\partial u/\partial y$. From Eq. (30b) we get dk/dy.

Example. Solve the equation

$$-\frac{x}{2x-y}y' + ln(2x-y) + \frac{2x}{2x-y} = 02x - y > 0.$$

Writing the equation in the form (26), we get

$$-\frac{x}{2x-y}dy + [ln(2x-y) + \frac{2x}{2x-y}]dx = 0 \tag{18}$$

Equation (34) is exact. Consider $M = ln(2x-y) + \frac{2x}{2x-y}$ and $N = -\frac{x}{2x-y}$. Then by differentiation we obtain

$$\begin{aligned}\frac{\partial M}{\partial y} &= \frac{-1}{2x-y} + \frac{2x}{(2x-y)^2},\\ \frac{\partial N}{\partial x} &= \frac{-1}{2x-y} + \frac{2x}{(2x-y)^2}.\end{aligned}$$

From Eq. (30b) we have $N = \frac{\partial u}{\partial y}$ and by integration

$$u = xln(2x-y) + k(x).$$

To determine $k(x)$ we differentiate u and apply Eq. (30a):

$$\frac{\partial u}{\partial x} = ln(2x-y) + \frac{2x}{2x-y} + \frac{dk}{dx} = M.$$

By simple algebraic manipulations, we find that $\frac{dk}{dx} = 0$ and, consequently, $k(x) = c$, where c is an arbitrary constant. We obtain the final form

$$xln(2x-y) = c.$$

Linear Differential Equations

We consider the first-order differential equation

$$y' + f(x)\,y = r(x), \tag{19}$$

which is linear in y and y' (f and r may be any given functions of x).

If $r(x) = 0$, $\forall\ x$ (for all x) the equation is *homogeneous*. For $r(x) \neq 0$ the equation is said to be *nonhomogeneous*.

Assuming that $f(x)$ and $r(x)$ are continuous for $x \in I$, we need to find a general formula for Eq. (39).

Case I Homogeneous equation

For the equation

$$y' + f(x)y = 0, \tag{20}$$

separating variables we have

$$\frac{dy}{y} = -f(x)dx \quad \text{or} \quad ln|y| = -\int f(x)dx + c^*,$$

and the solution is

$$y(x) = ce^{-\int f(x)dx} \quad (c = \pm e^{c^*} \quad \text{when} \quad y < 0 \text{ or } y > 0). \tag{21}$$

Case II Nonhomogeneous equation

Multiplying Eq. (39) by

$$F(x) = e^{h(x)} \quad \text{where} \quad h(x) = \int f(x)dx.$$

we find

$$e^h(y' + fy) = e^h r.$$

Since $h' = f$, we obtain

$$\frac{d}{dx}(ye^h) = e^h r.$$

Integrating the above relation we have

$$ye^h = \int e^h r dx + c.$$

The general solution of Eq. (39) in the form of an integral may be written

$$y(x) = e^{-h}[\int e^h r dx + c], \quad h = \int f(x)dx. \tag{22}$$

Example. Solve the differential equation

$$xy' + (1-x)y = xe^x.$$

We can rewrite the equation in the form

$$y' + (\frac{1}{x} - 1)y = e^x.$$

Comparing the previous equation to Eq. (47) we can identify

$$h = \int (\frac{1}{x} - 1)dx = \log x - x,$$

no constant being added in the integration. Thus, the solution will be

$$\begin{aligned} y &= e^{-\log x + x}(\int e^{\log x - x} e^x \, dx + c) \\ &= \frac{e^x}{x}(\int \frac{x}{e^x} e^x \, dx + c), \end{aligned}$$

or

$$y = e^x(\frac{x}{2} + \frac{c}{x}),$$

where c is an arbitrary constant.

Variation of Parameters

Another way of finding the general solution of linear differential equation

$$y' + f(x)y = r(x). \tag{23}$$

is the method of variation of parameters.

The solution corresponding to a homogeneous equation ($r(x) = 0$) is

$$v(x) = e^{-\int f(x)dx}. \tag{24}$$

With Eq. (54) we try to determine a function $u(x)$ such that

$$y(x) = u(x)v(x), \tag{25}$$

is the general solution of Eq. (53). This approach is called the method of variation of parameters [1].

Equations (55) and Eq. (53) can be combined into

$$u'v + u(v' + fv) = r,$$

or $u'v = r$, since $v' + fv = 0$. We find $u' = \frac{r}{v}$ and by integration

$$u = \int \frac{r}{v} dx + c.$$

We obtain the general solution

$$y = uv = v(\int \frac{r}{v} dx + c), \tag{26}$$

which is identical with Eq. (47) of the previous section.

SECOND ORDER DIFFERENTIAL EQUATION

Homogeneous Linear Equations

A second-order differential equation which can be written as

$$y'' + f(x)y' + g(x)y = r(x) \tag{27}$$

is said to be *linear*. It is said to be *nonlinear* if it cannot be written in the form of Eq. (59). The functions f and g are called the *coefficients* of the equation (59).

If $r(x) \neq 0$, then Eq. (59) is said to be *nonhomogeneous*. Otherwise, it is said to be *homogeneous* and takes the form

$$y'' + f(x)y' + g(x)y = 0. \tag{28}$$

It is called a *solution* of a differential equation of the second order on an interval J a function $y = \phi(x)$ which is defined and two times differentiable on J. Moreover, the equation becomes an identity if ϕ and its derivative replace the unknown function y and its derivatives, respectively. For the case of homogeneous equations, the following theorem states that solutions of Eq. (60) can be obtained from known solutions by multiplication by constants and by addition.

Fundamental Theorem [1] *If a solution of the homogeneous linear differential equation* (60) *on the interval J is multiplied by any constant, the resulting function is also a solution of Eq.* (60) *on J. The sum of two solutions of Eq.* (60) *on J is also a solution of Eq.* (60) *on that interval.*

Proof We assume that $\phi(x)$ obeys the conditions to be a solution of Eq. (60) on J. If we replace y by $c\phi(x)$ is into Eq. (60), we obtain

$$(c\phi)'' + f(c\phi)' + gc\phi = c[\phi'' + f\phi' + g\phi].$$

Since ϕ is a solution of Eq. (60), then $\phi'' + f\phi' + g\phi = 0$ and we find that $c\,\phi$ is also a solution of Eq. (60). The second part of the theorem can be proved in the same way.

Example. The functions $y_1 = \phi_1 = x$ and $y_2 = \phi_2 = x^2$, $x \in \mathcal{R} - \{0\}$ ($J \equiv \mathcal{R} - \{0\}$), are two solutions of the equation

$$x^2y'' - 2xy' + 2y = 0.$$

The function $y_3 = c_1\phi_1 + c_2\phi_2 = c_1x + c_2x^2$ is also a solution of the equation.

Homogeneous Equations with Constant Coefficients

We consider the homogeneous equations of the form

$$y'' + ay' + by = 0, \tag{29}$$

where $a, b \in \mathcal{R}$ are constants, and $x \in \mathcal{R}$. The solution of the first-order homogeneous linear equation with constant coefficients

$$y' + ky = 0,$$

is an exponential function,

$$y = C\, e^{-kx}.$$

We assume that

$$y = e^{\lambda x}, \tag{30}$$

may be a solution of Eq. (63) if λ is properly chosen. Substituting Eq. (66) and its derivatives

$$y' = \lambda e^{\lambda x} \quad \text{and} \quad y'' = \lambda^2 e^{\lambda x},$$

into Eq. (63), we obtain

$$(\lambda^2 + a\lambda + b)e^{\lambda x} = 0.$$

So Eq. (66) is a solution of Eq. (63), if λ is a solution of the equation

$$\lambda^2 + a\lambda + b = 0. \tag{31}$$

Eq. (69) is called the *characteristic equation* of Eq. (63). Its roots are

$$\lambda_1 = \frac{1}{2}(-a + \sqrt{a^2 - 4b}), \quad \lambda_2 = \frac{1}{2}(-a - \sqrt{a^2 - 4b}). \tag{32}$$

From derivation it follows that the functions

$$y_1 = e^{\lambda_1 x} \quad \text{and} \quad y_2 = e^{\lambda_2 x}, \tag{33}$$

are solutions of Eq. (63). This result can be verified by substituting Eq. (71) into Eq. (63).

Elementary algebra states that, since a and b are real, the characteristic equation may have

Case I	two distinct real roots,
Case II	two complex conjugate roots, or
Case III	a real double root.

Example 1. Solve the equation

$$2y'' - 5y' + 2y = 0.$$

The characteristic equation of the given differential equation will be

$$2\lambda^2 - 5\lambda + 2 = 0$$

so that

$$\lambda_1 = \frac{1}{2}, \quad \lambda_2 = 2.$$

Then the general solution is

$$y = c_1 e^{x/2} + c_2 e^{2x}.$$

Example 2. The equation

$$y'' + 2y' + 5y = 0$$

has the characteristic equation

$$\lambda^2 + 2\lambda + 5 = 0$$

from which

$$\lambda_{1,2} = -1 \pm 2i.$$

The general solution will be

$$y = e^{-x}(c_1 cos 2x + c_2 sin 2x).$$

Example 3. The equation

$$y'' - 2y' + 1 = 0$$

has the characteristic equation

$$(\lambda - 1)^2 = 0$$

which gives

$$\lambda_{1,2} = 1.$$

We obtain the general solution

$$y = e^x.$$

General Solution. Fundamental System

Definition The *general solution* of a second order differential equation is a solution which contains two arbitrary independent constants, i.e. the solution cannot be reduced to a form containing only one arbitrary constant or none. A *particular solution* is a solution obtained from the general solution assigning specific values to the arbitrary constants.

We consider the general homogeneous linear equation

$$y'' + f(x)y' + g(x)y = 0, \tag{34}$$

and two solutions $y_1(x)$ and $y_2(x)$ of this equation. The Fundamental Theorem states that

$$y(x) = c_1\, y_1(x) + c_2\, y_2(x), \tag{35}$$

is a general solution of Eq. (84), where c_1 and c_2 are two arbitrary constants.

Two functions $y_1(x)$ and $y_2(x)$ are *linearly dependent* on an open interval I where both functions are defined, if they are proportional on I

$$\text{(a)} y_1 = m\, y_2 \quad \text{or} \quad \text{(b)} y_2 = n\, y_1, \tag{36}$$

for all $x \in I$, where m and n are numbers. If the functions are not proportional, they are *linearly independent* on I.

If at least one of the functions y_1 and y_2 is identically zero on I, then the functions are linearly dependent on I. In any other case the functions are linearly dependent on I if and only if the quotient y_1/y_2 is constant on I. Hence, if y_1/y_2 depends on x on I, then y_1 and y_2 are linearly independent on I [1].

Example 1. The functions

$$y_1 = 9x \quad \text{and} \quad y_2 = 3x$$

are linearly dependent, because the quotient $y_1/y_2 = 3 =$ const while the functions

$$y_1 = x^2 + x \quad \text{and} \quad y_2 = x$$

are linearly independent because $y_1/y_2 = x + 1 \neq$ const.

Two linearly independent solutions of Eq. (84) on I constitute a *fundamental system* or a *basis* of solutions on I.

Theorem [1] *The solution*

$$y(x) = c_1 y_1(x) + c_2 y_2(x) \quad (c_1, c_2 \, arbitrary)$$

is a general solution of the differential equation Eq. (84) *on an interval I of the x-axis if and only if the functions* y_1 *and* y_2 *constitute a fundamental system of solutions of Eq.* (84) *on I.* y_1 *and* y_2 *constitute such a fundamental system if and only if their quotient* y_1/y_2 *is not constant on I but depends on x.*

Example 2. The equation

$$y'' - 2y' - 15y = 0$$

has the solutions

$$y_1 = e^{5x} \quad \text{and} \quad y_2 = e^{-3x}.$$

These solutions constitute a fundamental system because the ratio y_1/y_2 is not constant. The general solution is

$$y = c_1 y_1 + c_2 y_2 = c_1 e^{5x} + c_2 e^{-3x}.$$

Complex Roots of the Characteristic Equation. Initial Value Problem

The solutions of the homogeneous linear equation with constant coefficients

$$y'' + ay' + by = 0 \quad (a, b \, \text{real}) \tag{37}$$

are

$$y_1 = e^{\lambda_1 x} \quad \text{and} \quad y_2 = e^{\lambda_2 x}, \tag{38}$$

where λ_1 and λ_2 are the roots of the corresponding characteristic equation

$$\lambda^2 + a\lambda + b = 0. \tag{39}$$

In the case of $\lambda_1 \neq \lambda_2$, the quotient y_1/y_2 is not constant, and the solutions constitute a fundamental system for all x. The general solution is

$$y = c_1 \, e^{\lambda_1 x} + c_2 \, e^{\lambda_2 x}. \tag{40}$$

The solutions of the Eq. (94) are real if the distinct roots of the corresponding characteristic equation are real (Case I). If λ_1 and λ_2 are complex conjugate roots of the form (Case II)

$$\lambda_1 = \ p + iq, \quad \lambda_2 = \ p - iq,$$

then the solutions Eq. (94) are complex

$$y_1 = e^{(\,p+iq)x}, \quad y_2 = e^{(\,p-iq)x}.$$

The real solutions can be derived from the complex solutions by applying the *Euler formulas*

$$e^{i\theta} = cos\,\theta + i\,sin\,\theta, \quad e^{-i\theta} = cos\,\theta - i\,sin\,\theta,$$

for $\theta = qx$. The first solution becomes

$$y_1 = e^{(\,p+iq)x} = e^{\,px} e^{iqx} = e^{\,px}(cos\,qx + isin\,qx),$$

while the second one is

$$y_2 = e^{(\,p-iq)x} = e^{\,px} e^{-iqx} = e^{\,px}(cos\,qx - i\,sin\,qx).$$

From Fundamental Theorem we can conclude that they are solutions of the differential equation Eq. (93). The corresponding general solution is

$$y(x) = e^{\,px}(A\,cos\,qx + B\,sinqx) \tag{41}$$

where A and B are arbitrary constants.

Example 1. Let us consider the second order differential equation with constant coefficients

$$y'' - 4y' + 5y = 0$$

The corresponding characteristic equation is

$$\lambda^2 - 4\lambda + 5 = 0,$$

with the roots

$$\lambda_1 = \ p + iq = 2 + i \quad \text{and} \quad \lambda_2 = \ p - iq = 2 - i.$$

For this example $p = 2$, $q = 1$, and from Eq. (102) the answer is

$$y = e^{2x}(A\,cos\,x + B\,sin\,x).$$

Let us consider the values of the solution $y(x)$ and its derivative $y'(x)$ at an initial point $x = x_0$

$$y(x_0) = K, \quad y'(x_0) = L, \tag{42}$$

The conditions Eq. (107) and the equation Eq. (93) constitute an *initial value problem*. To solve such a problem we must find a particular solution of Eq. (93) satisfying Eq. (107). Such a problem has a unique solution.

Example 2. Let us consider the initial value problem

$$y'' - 4y' + 5y = 0, \quad y(0) = 2, \quad y'(0) = 0.$$

A fundamental system of solutions is

$$e^{2x}\,cos\,x \quad \text{and} \quad e^{2x}\,sin\,x,$$

and the corresponding general solution is

$$y(x) = e^{2x}(A\,cos\,x + B\,sin\,x),$$

with the initial condition $y\,(0) = A$. The derivative

$$y' = e^{2x}[(2A + B)cos\,x + (2B - A)sin\,x)],$$

has the initial value $y'(0) = 2A + B$. Solving the initial conditions system,

$$\begin{aligned} y(0) = A &= 2, \\ y'(0) = 2A + B &= 0. \end{aligned}$$

we get $A = 4, B = -1$, and the general solution of the differential equation is

$$y = e^{2x}(2\,cos\,x - 4\,sin\,x). \tag{43}$$

Double Root of the Characteristic Equation

Now we consider the case when the characteristic equation associated to a homogeneous linear differential equation with constant coefficients has a double root (*critical case*). If the differential equation takes the general form

$$y'' + ay' + by = 0, \tag{44}$$

then the characteristic equation will be

$$\lambda^2 + a\lambda + b = 0. \tag{45}$$

A double root appears if an only if the discriminant of Eq. (115) is zero, that is

$$a^2 - 4b = 0, \quad \text{and, then,} \quad b = \frac{1}{4}a^2.$$

The double root of the characteristic equation is $\lambda = -a/2$. Then, the first solution of the differential equation is

$$y_1 = e^{ax/2}. \tag{46}$$

To find another solution $y_2(x)$ the method of variation of parameters may be applied. The second solution takes the form

$$y_2(x) = u(x)y_1(x) \quad \text{where} \quad y_1(x) = e^{-ax/2}.$$

Substituting y_2 in the differential equation with $b = a^2/4$ we obtain

$$u(y_1'' + ay_1' + \frac{1}{4}a^2 y_1) + u'(2y_1' + ay_1) + u'' y_1 = 0.$$

The expression in the first parentheses is zero because y_1 is a solution. The second parentheses is also zero because

$$2y_1' = 2(-\frac{a}{2})e^{-ax/2} = -ay_1.$$

The equation reduces to $u''\, y_1 = 0$, and a solution is $u = x$. Consequently, the second solution is

$$y_2(x) = xe^{\lambda x} \quad (\lambda = -\frac{a}{2}). \tag{47}$$

We can observe that the solutions y_1 and y_2 are linearly independent. This case can be summarized by the following theorem

Theorem (Double root) [1] *In the case of a double root of Eq.* (115) *the functions* (117) *and* (121) *are solutions of Eq.* (114). *They constitute a fundamental system. The corresponding general solution is*

$$y = (c_1 + c_2 x)e^{\lambda x} \quad (\lambda = -\frac{a}{2}). \tag{48}$$

Example. Solve the following differential equation

$$y'' - 4y' + 4y = O.$$

The double root of the characteristic equation is $\lambda = -4$. Then, the fundamental system of solutions is

$$e^{2x} \quad \text{and} \quad xe^{2x}$$

and the corresponding general solution is

$$y = (c_1 + c_2 x)e^{2x}.$$

All three cases are summarized in the following table:

Case	Roots of Eq. (115)	Fundamental system of Eq. (114)	General solution of Eq. (114)
I	Distinct real λ_1, λ_2	$e^{\lambda_1 x}, e^{\lambda_2 x}$	$y = c_1 e^{\lambda_1 x} + c_2 e^{\lambda_2 x}$
II	Complex conjugate $\lambda_1 = p + iq, \lambda_2 = p - iq$	$e^{px} \cos qx\ e^{px} \sin qx$	$y = e^{px}(A \cos qx + B \sin qx)$
III	Real double root $\lambda = -a/2$	$e^{\lambda x}, xe^{\lambda x}$	$y = (c_1 + c_2 x)e^{\lambda x}$

Series Solutions

We consider the general homogeneous linear second-order equation

$$P(x)\frac{d^2y}{dx^2} + Q(x)\frac{dy}{dx} + R(x)y = 0 \tag{49}$$

with $P(x) \neq 0$ in the interval $\alpha < x < \beta$. We want to determine a polynomial solution $y(x)$ of Eq. (126).

Definition A functions $f(x)$ can be expanded in power series so that

$$f(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \ldots = n = 0 \sum a_n(x - x_0)^n. \tag{50}$$

Such functions are said to be *analytic* at $x = x_0$ and the series (127) is called the Taylor series of f about $x = x_0$. The coefficients a_n can be computed with the formula $a_n = f^{(n)}(x_0)/n!$ where $f^{(n)}(x) = d^n\ f(x)/dx^n$.

We consider the functions $P(x)$, $Q(x)$, and $R(x)$ as power series about x_0

$$\begin{aligned} P(x) &= p_0 + p_1(x - x_0) + \ldots, \; Q(x) = q_0 + q_1(x - x_0) + \ldots, \\ R(x) &= r_0 + r_1(x - x_0) + \ldots \end{aligned}$$

and $y(x) = a_0 + a_1(x - x_0) + \ldots$.

Theorem [2] Let the functions $Q(x)/P(x)$ and $R(x)/P(x)$ have convergent Taylor series expansions about $x = x_0$ for $|x - x_0| < \rho$. Then, every solution $y(x)$ of the differential equation

$$P(x)\frac{d^2y}{dx^2} + Q(x)\frac{dy}{dx} + R(x)y = 0 \tag{51}$$

is analytic at $x = x_0$, and the radius of convergence of its Taylor series expansion about $x = x_0$ is at least ρ. The coefficients $a_2, a_3, \ldots$ in the Taylor series expansion

$$y(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \ldots \tag{52}$$

are determined by plugging the series (130) into the differential equation (129) and setting the sum of the coefficients of the like powers of x in this expression equal to zero.

Example. Solve the equation

$$x^2\frac{d^2y}{dx^2} + (x^2 + x)\frac{dy}{dx} - y = 0.$$

Assuming a solution of the form

$$y = k = 0 \sum a_k x^k,$$

we obtain

$$\frac{dy}{dx} = k = 0 \sum k a_k x^{k-1}, \quad \frac{d^2y}{dx^2} = k = 0 \sum k(k-1) a_k x^{k-2},$$

and hence

$$k=0\sum k(k-1)a_kx^k+k=0\sum ka_kx^{k+1}+k=0\sum ka_kx^k-k=0\sum a_kx^k.$$

The first, third, and fourth summations may be combined to give

$$k=0\sum[k(k-1)+k-1]a_kx^k=k=0\sum(k^2-1)a_kx^k,$$

and hence there follows

$$k=0\sum(k^2-1)a_kx^k+k=0\sum ka_kx^{k+1}.$$

In order to combine these sums, we replace k by n in the first and $(k+1)$ by n in the second, to obtain

$$n=0\sum(n^2-1)a_nx^n+n=1\sum(n-1)a_{n-1}x^n.$$

Since the ranges of summation differ, the term corresponding to $n=0$ must be extracted from the first sum, after which the remainder of the first sum can be combined with the second. In this way we find

$$-a_0+n=1\sum[(n^2-1)a_n+(n-1)a_{n-1}]x^n.$$

In order that the previous relation may vanish identically, the constant term, as well as the coefficients of the successive powers of x, must vanish independently, giving the condition

$$a_0=0$$

and the recurrence formula

$$(n-1)[(n+1)a_n+a_{n-1}]=0\quad(n=1,2,3,\ldots).$$

The recurrence formula is automatically satisfied when $n=1$. When $n\geq 2$, it becomes

$$a_n=-\frac{a_{n-1}}{n+1}\quad(n=2,3,4\ldots).$$

Hence, we obtain

$$a_2=-\frac{a_1}{3},\quad a_3=-\frac{a_2}{4}=\frac{a_1}{3\cdot 4},\quad a_4=-\frac{a_3}{5}=-\frac{a_1}{3\cdot 4\cdot 5},\ldots.$$

Thus, in this case $a_0=0$, a_1 is arbitrary, and all succeeding coefficients are determined in terms of a_1. The solution becomes

$$y=a_1(x-\frac{x^2}{3}+\frac{x^3}{3\cdot 4}-\frac{x^4}{3\cdot 4\cdot 5}+\ldots).$$

If this solution is put in the form

$$\begin{aligned}y&=\frac{2a_1}{x}(\frac{x^2}{2!}-\frac{x^3}{3!}+\frac{x^4}{4!}-\frac{x^5}{5!}+\ldots)\\&=\frac{2a_1}{x}[x-1+(1-\frac{x}{1!}+\frac{x^2}{2!}-\frac{x^3}{3!}+\frac{x^4}{4!}-\ldots)],\end{aligned}$$

the series in parentheses in the final form is recognized as the expansion of e^{-x}, and, writing $2a_1=c$, the solution obtained may be put in the closed from

$$y=c(\frac{e^{-x}-1+x}{x}).$$

In this case only one solution was obtained. This fact indicates that any linearly independent solutions cannot be expanded in power series near $x=0$. That is, it is not regular at $x=0$.

Regular Singular Points

We consider the differential equations

$$x^2\frac{d^2y}{dx^2}+\alpha x\frac{dy}{dx}+\beta y=0 \tag{53}$$

which can be rewritten in the form

$$\frac{d^2y}{dx^2}+\frac{\alpha}{x}\frac{dy}{dx}+\frac{\beta}{x^2}y=0. \tag{54}$$

A generalization of Eq. (147) is the equation

$$\frac{d^2y}{dx^2}+\ p(x)\frac{dy}{dx}+q(x)y=0 \tag{55}$$

where $p(x)$ and $q(x)$ can be expanded in series of the form

$$\begin{aligned}p(x)&=\frac{p_0}{x}+p_1+\ p_2x+\ p_3x^2+\ldots\\q(x)&=\frac{q_0}{x^2}+\frac{q_1}{x}+q_2+q_3x+q_4x^2\ldots\end{aligned} \tag{56}$$

Definition [2] Equation (148) is said to have a *regular singular point* at $x=0$ if $p(x)$ and $q(x)$ have series expansions of the form (149). Equivalently, $x=0$ is a regular singular point of Eq. (148) if the functions $x\ p(x)$ and $x^2q(x)$ are analytic at $x=0$. Equation (148) is said to have a regular singular point at $x=x_0$ if the functions $(x-x_0)\,p(x)$ and $(x-x_0)^2q(x)$ are analytic at $x=x_0$. A singular point of Eq. (148) which is not regular is called *irregular*.

Example. Classify the singular points of Bessel's equation of order ν

$$x^2\frac{d^2y}{dx^2}+x\frac{dy}{dx}+(x^2-\nu^2)y=0, \tag{57}$$

where ν is a constant 1.

For $x=0$ we have $P(x)=x^2=0$. Hence, $x=0$ is the only singular point of Eq. (150). Dividing both sides of Eq. (150) by x^2 gives

$$\frac{d^2y}{dx^2}+\frac{1}{x}\frac{dy}{dx}+(1-\frac{\nu^2}{x^2})y=0.$$

The functions

$$x\ p(x)=1\quad\text{and}\quad x^2\,q(x)=x^2-\nu^2$$

are both analytic at $x=0$. Hence Bessel's equation of order ν has a regular singular point at $x=0$.

Nonhomogeneous Linear Equations

Let us consider a second-order linear nonhomogeneous equation

$$y''+\ f(x)y'+g(x)y=r(x). \tag{58}$$

A general solution $y(x)$ of Eq. (153) can be obtained from a general solution $y_h(x)$ of the corresponding homogeneous equation

$$y''+\ f(x)y'+g(x)y=0,$$

by adding to $y_h(x)$ any particular solution $\tilde{y}$ of Eq. (153) involving no arbitrary constant [1]

$$y(x)=y_h(x)+\tilde{y}(x). \tag{59}$$

To show that $y(x)$ is a solution of the nonhomogeneous differential equation we substitute Eq. (155) into Eq. (153). Then the left-hand side of Eq. (153) becomes

$$(y_h + \tilde{y})'' + f(y_h + \tilde{y})' + g(y_h + \tilde{y}).$$

or

$$(y_h'' + f y_h' + g y_h) + \tilde{y}'' + f\tilde{y}' + g\tilde{y}.$$

The expression in the parentheses is zero because y_h is a solution of Eq. (155). The sum of the other terms is equal to $r(x)$ because $\tilde{y}$ satisfies Eq. (153). Hence $y(x)$ is a general solution of the Eq. (153).

Theorem [1] Suppose that $f(x), g(x)$, and $r(x)$ in Eq. (153) are continuous functions on an open interval I. Let $Y(x)$ be any solution of Eq. (153) on I containing no arbitrary constants. Then $Y(x)$ is obtained from Eq. (155) by assigning suitable values to the two arbitrary constants contained in the general solution $y_h(x)$ of Eq. (155). In Eq. (155), the function $\tilde{y}(x)$ is any solution of Eq. (153) on I containing no arbitrary constants.

Proof Let set $Y - \tilde{y} = y^*$. Then

$$y^{*\prime\prime} + f y^{*\prime} + g y^* = (Y'' + fY' + gY) - (\tilde{y}'' + f\tilde{y}' + g\tilde{y}) = r - r = 0,$$

that is, y^* is a solution of Eq. (155) which does not contain arbitrary constants. It can be obtained from y_h by assigning suitable values to the arbitrary constants in y_h. From this, since $Y = y^* + \tilde{y}$, the statement follows.

Theorem [1] A general solution $y(x)$ of the linear nonhomogeneous differential equation Eq. (153) is the sum of a general solution $y_h(x)$ of the corresponding homogeneous equation Eq. (155) and an arbitrary particular solution $y_p(x)$ of Eq. (153):

$$y(x) = y_h(x) + y_p(x) \tag{60}$$

Example. Solve the equation

$$y'' + y = sec\, x.$$

The homogeneous equation $y'' + y = 0$ has the characteristic equation $\lambda^2 + 1 = 0$ with roots $\lambda_1 = i$ and $\lambda_2 = -i$, so, the general solution of the homogeneous equation is

$$y = c_1 \cos x + c_2 \sin x.$$

Using the method of variation of parameter we have the following system of equations

$$\begin{aligned} c_1' cos\, x + c_2' sin\, x &= 0, \\ -c_1' sin\, x + c_2' cos\, x &= sec\, x, \end{aligned}$$

with the solution

$$c_1' = -tan\, x, \quad c_2' = 1.$$

Thus by integrating,

$$c_1 = -ln\, sec\, x + A_1, \quad c_2 = x + A_2,$$

and the general solution is of the nonhomogeneous equation is

$$y = A_1 \cos x + A_2 \sin x - \cos x \, ln\, sec\, x + x \sin x.$$

The Method of Variation of Parameters

This method can be applied to solve the nonhomogeneous equation of the form

$$\frac{d^2y}{dx^2} + p(x)\frac{dy}{dx} + q(x)y = g(x), \tag{61}$$

once the solutions of the homogeneous equation

$$\frac{d^2y}{dx^2} + p(x)\frac{dy}{dx} + q(x)y = 0 \tag{62}$$

are known. Let $y_1(x)$ and $y_2(x)$ be two linearly independent solutions of the homogeneous equation (167). We will try to find a particular solution $\psi(x)$ of the nonhomogeneous Eq. (166) of the form [2]

$$\psi(x) = u_1(x)y_1(x) + u_2(x)y_2(x). \tag{63}$$

The differential equation (166) imposes only one condition on the two unknown functions $u_1(x)$ and $u_2(x)$. We may impose an additional condition on $u_1(x)$ and $u_2(x)$ such that the left hand side of the nonomogeneous equation be as simple as possible. Computing

$$\begin{aligned} \frac{d}{dx}\psi(x) &= \frac{d}{dx}[u_1y_1 + u_2y_2] \\ &= [u_1y_1' + u_2y_2'] + [u_1'y_1 + u_2'y_2] \end{aligned}$$

we see that $d^2\psi/dx^2$ will contain no second-order derivatives of u_1 and u_2 if

$$y_1(x)u_1'(x) + y_2(x)u_2'(x) = 0. \tag{64}$$

Imposing the condition (170) on the functions $u_1(x)$ and $u_2(x)$ the left hand side of the Eq. (166) becomes

$$\begin{aligned} & [u_1y_1' + u_2y_2']' + p(x)[u_1y_1' + u_2y_2'] + q(x)[u_1y_1 + u_2y_2] \\ = \; & u_1'y_1' + u_2'y_2\prime + u_1[y_1'' + p(x)y_1' + q(x)y_1] + u_2[y_2'' + p(x)y_2' + q(x)y_2] \\ = \; & u_1'y_1' + u_2'y_2'. \end{aligned}$$

If $u_1(x)$ and $u_2(x)$ satisfy the two equations

$$\begin{aligned} y_1(x)u_1' + y_2(x)u_2'(x) &= 0 \\ y_1'(x)u_1'(x) + y_2'(x)u_2'(x) &= g(x), \end{aligned}$$

then $\psi(x) = u_1y_1 + u_2y_2$ is a solution of the nonhomogeneous equation (166). We solve the above system of equations as follows

$$\begin{aligned} [y_1(x)y_2'(x) - y_1'(x)y_2(x)]u_1'(x) &= -g(x)y_2(x) \\ [y_1(x)y_2'(x) - y_1'(x)y_2(x)]u_2'(x) &= g(x)y_1(x). \end{aligned}$$

The function $u_1'(x)$ and $u_2'(x)$ are

$$u_1'(x) = -\frac{g(x)y_2(x)}{W[y_1, y_2](x)} \quad \text{and} \quad u_2'(x) = \frac{g(x)y_1(x)}{W[y_1, y_2](x)}, \tag{65}$$

where $W[y_1, y_2](x)$ is the Wronskian of the solutions

$$W[y_1, y_2](x) = \begin{vmatrix} y_1 & y_2 \\ y_1' & y_2' \end{vmatrix}.$$

Integrating the right-hand sides of Eqs. (174) we obtain $u_1(x)$ and $u_2(x)$.

Example.

a. Find a particular solution $\psi(x)$ of the equation

$$\frac{d^2y}{dx^2} + 4y = 8\sin x \tag{66}$$

b. Find the solution $y(x)$ of Eq. (176) which satisfies the initial conditions $y(0) = 1$, $y'(0) = 1$.

0.1. The functions $y_1(x) = cos\,2x$ and $y_2(x) = sin\,2x$ are two linearly independent solutions of the homogeneous equation $y'' + 4y = 0$ with
$W[y_1, y_2](x) = y_1y_2' - y_1'y_2 = (cos\,x)cos\,x - (-sin\,x)sin\,x = 1$.
Thus, from Eqs. (174),

$$u_1'(x) = -8\,sin^2 x \quad \text{and} \quad u_2'(x) = 8\,sin\,x\,cos\,x. \tag{67}$$

Integrating the first equation of (178) gives

$$\begin{aligned} u_1(x) &= -8\int sin^2 x\,dx = -4\int(1 - cos\,2x)\,dx \\ &= -4\int dx + 4\int cos\,2x\,dx \\ &= -4x + 2\,sin\,2x. \end{aligned}$$

while integrating the second equation of (178) gives

$$u_2(x) = \int 4\,sin\,2x\,dx = 4\int sin\,2x\,dx = -2\,cos\,2x.$$

Consequently,
$\psi(x) = cos\,x[-4x + 2\,sin\,2x] + sin\,x(-2\,cos\,2x)$
is a particular solution of Eq. (176).

0.2. $y(x) = c_1\,cos\,x + c_2\,sin\,x + cos\,x(-4x + 2\,sin\,2x) - 2\,sin\,x\,cos\,2x$
for some choice of constants c_1, c_2. The constants c_1 and c_2 are determined from the initial conditions
$1 = y(0) = c_1 \quad \text{and} \quad 1 = y'(0) = c_2 - 2$.
Hence, $c_1 = 1, c_2 = 3$ and
$y(x) = cos\,x + 3\,sin\,x + cos\,x(-4x + 2\,sin\,2x) - 2\,sin\,x\,cos\,2x$.

DIFFERENTIAL EQUATIONS OF ARBITRARY ORDER

Homogeneous Linear Equations

A *linear* differential equation of nth order can be written in the following general form

$$y^{(n)} + f_{n-1}(x)y^{(n-1)} + \ldots + f_1(x)y' + f_0(x)y = r(x) \tag{68}$$

where the function r on the right-hand side and the coefficient $f_0, f_1, \ldots, f_{n-1}$ are any given functions of x, and $y^{(n)}$ is the nth derivative of y.

Eq. (185) is said to be *homogeneous* if $r(x) = 0$. Then, Eq. (185) becomes

$$y^{(n)} + f_{n-1}(x)y^{(n-1)} + \ldots + f_1(x)y' + f_0(x)y = 0. \tag{69}$$

If $r(x) \neq 0$, Eq. (185) is said to be *nonhomogeneous*.

A function $y = \phi(x)$ is called a *solution* of a differential equation of nth order on an interval I if $\phi(x)$ is defined and n times differentiable on I and is such that the equation becomes an identity when we replace the unspecified function y and its derivatives in the equation by ϕ and its corresponding derivatives [1].

Existence and uniqueness theorem [1], [3]. If $f_0(x), \ldots, f_{n-1}(x)$ in Eq. (186) are continuous functions on an open interval I, then the initial value problem consisting of the equation Eq. (186) and the n initial conditions

$$y(x_0) = K_1,\ y'(x_0) = K_2, \ldots, y^{(n-1)}(x_0) = K_n,$$

has a unique solution $y(x)$ on I; here x_0 is any fixed point in I, and $K_1, \ldots, K_n$ are given numbers.

A set of functions, $y_1(x), \ldots, y_n(x)$ are *linearly dependent* on some interval I where they are defined, if one of them can be represented on I as a "*linear combination*" of the other $n - 1$ functions. Otherwise the functions are *linearly independent* on I.

A *fundamental system* or a *basis* of solutions of the linear homogeneous equation Eq. (186) is a set of n linearly independent solutions $y_1(x), \ldots, y_n(x)$ of that equation.

If $y_1, \ldots, y_n$ is such a fundamental system, then

$$y(x) = c_1y_1(x) + \ldots + c_ny_n(x) \quad (c_1, \ldots, c_n \text{ arbitrary}) \tag{70}$$

is a general solution of Eq. (186) on I. The test for linear dependence and independence of solutions can be generalized to nth order equations as follows

Theorem [1] Suppose that the coefficients $f_0(x), \ldots, f_{n-1}(x)$ of Eq. (186) are continuous on an open interval I. Then n solutions $y_1, \ldots, y_n$ of Eq. (186) on I are linearly dependent on I if and only if their Wronskian

$$W(y_1, \ldots, y_n) = \begin{vmatrix} y_1 & y_2 & \ldots & y_n \\ y_1' & y_2' & \ldots & y_n' \\ \vdots & \vdots & \ldots & \vdots \\ y_1^{(n-1)} & y_2^{(n-1)} & \ldots & y_n^{(n-1)} \end{vmatrix} \tag{71}$$

is zero for some $x = x_0$ in I. (If $W = 0$ at $x = x_0$, then $W \equiv 0$ on I).

Theorem [1] Let Eq. (188) be a general solution of Eq. (186) on an open interval I where $f_0(x), \ldots, f_{n-1}(x)$ are continuous, and let $Y(x)$ be any solution of Eq. (186) on I involving no arbitrary constants. Then $Y(x)$ is obtained from Eq. (188) by assigning suitable values to the arbitrary constants $c_1, \ldots, c_n$.

Example. The equation

$$y''' - 2y'' - y' + 2y = 0. \tag{72}$$

has the solutions $y_1 = e^x$, $y_2 = e^{2x}$, and $y_3 = e^{3x}$.

The Wronskian is

$$W(e^x, e^{2x}, e^{3x}) = \begin{vmatrix} e^x & e^{2x} & e^{3x} \\ e^x & 2e^{2x} & 3e^{3x} \\ e^x & 4e^{2x} & 9e^{3x} \end{vmatrix} = 2e^{6x} \neq 0,$$

which shows that the functions constitute a fundamental system of solutions of Eq. (190). The corresponding general solution is

$$y = c_1e^x + c_2e^{2x} + c_3e^{3x}.$$

Homogeneous Linear Equations with Constant Coefficients

A linear homogeneous equation of order n with constant coefficients

$$y^{(n)} + a_{n-1}y^{(n-1)} + \ldots + a_1y' + a_0y = 0, \tag{73}$$

has the correspondent characteristic equation

$$\lambda^n + a_{n-1}\lambda^{n-1} + \ldots + a_1\lambda + a_0 = 0. \tag{74}$$

If this equation has n distinct roots $\lambda_1, \ldots, \lambda_n$, then the n solutions

$$y_1 = e^{\lambda_1 x}, \ldots, y_n = e^{\lambda_n x} \tag{75}$$

constitute a fundamental system for all x, and the corresponding general solution of Eq. (193) is

$$y = c_1 e^{\lambda_1 x} + \ldots + c_n e^{\lambda_n} x. \tag{76}$$

If λ is a root of order m, then

$$e^{\lambda x}, xe^{\lambda x}, \ldots, x^{m-1} e^{\lambda x} \tag{77}$$

are m linearly independent solutions of Eq. (193) corresponding to that root.

Example. Consider the differential equation

$$y''' + 3y'' - 4y' - 12y = 0.$$

The characteristic equation

$$\lambda^3 + 3\lambda^2 - 4\lambda + 12 = 0$$

has the solutions $\lambda_1 = -2, \lambda_2 = 2$, and $\lambda_3 = -3$, and the corresponding general solution Eq. (196) is

$$y = c_1 e^{-2x} + c_2 e^{2x} + c_3 e^{-3x}.$$

Linear Differential Equations in State Space Form

The nth-order differential equation

$$a_n(t)\frac{d^n y}{dt^n} + a_{n-1}(t)\frac{d^{n-1} y}{dt^{n-1}} + \ldots + a_0 y = 0,$$

can be transformed into a system of n first order equations.

With the notations

$$x_1(t) = y,\ x_2(t) = dy/dt, \ldots x_n(t) = d^{n-1} y/dt^{n-1},$$

we obtain the system

$$\frac{dx_1}{dt} = x_2, \quad \frac{dx_2}{dt} = x_3, \ldots, \quad \frac{dx_{n-1}}{dt} = x_n,$$

and

$$\frac{dx_n}{dt} = -\frac{a_{n-1}(t)x_n + a_{n-2}(t)x_{n-1} + \ldots + a_0 x_1}{a_n(t)}.$$

A system of n first-order linear equations has the general form

$$\begin{aligned} \frac{dx_1}{dt} &= a_{11}(t)x_1 + \ldots + a_{1n}(t)x_n + g_1(t), \\ &\vdots \\ \frac{dx_n}{dt} &= a_{n1}(t)x_1 + \ldots + a_{nn}(t)x_n + g_n(t), \end{aligned} \tag{78}$$

and is said to be *nonhomogeneous* ($g_i(t) \neq 0, i = 1, \ldots, n$).

The system

$$\begin{aligned} \frac{dx_1}{dt} &= a_{11}(t)x_1 + \ldots + a_{1n}(t)x_n, \\ &\vdots \\ \frac{dx_n}{dt} &= a_{n1}(t)x_1 + \ldots + a_{nn}(t)x_n, \end{aligned} \tag{79}$$

is said to be *homogeneous* ($g_i(t) = 0, i = 1, \ldots, n$).

The homogeneous linear system with constant coefficients (a_{ij} do not depend on t)

$$\begin{aligned} \frac{dx_1}{dt} &= a_{11}x_1 + \ldots + a_{1n}x_n, \\ &\vdots \\ \frac{dx_n}{dt} &= a_{n1}x_1 + \ldots + a_{nn}x_n, \end{aligned} \tag{80}$$

can be written in matrix notation as

$$\dot{\mathbf{x}} = A\,\mathbf{x}, \tag{81}$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}.$$

Theorem (existence-uniqueness theorem) [2] There exists one, and only one, solution of the initial-value problem for $-\infty < t < \infty$

$$\dot{\mathbf{x}} = A\mathbf{x}, \quad \mathbf{x}(t_0) = \mathbf{x}^0 = \begin{bmatrix} x_1^0 \\ x_2^0 \\ \vdots \\ x_n^0 \end{bmatrix}. \tag{82}$$

The dimension of the space of all solutions of the homogeneous linear system of differential equations (208) is n.

Solution via the Eigenvalue-Eigenvector Method. Consider the linear homogeneous differential system

$$\dot{\mathbf{x}} = A\mathbf{x}. \tag{83}$$

Assuming a solution of the form

$$\mathbf{x}(t) = e^{\lambda t}\,\mathbf{v}, \quad \mathbf{v} = \text{constant vector}.$$

Eq. (211) becomes

$$\lambda e^{\lambda t}\mathbf{v} = e^{\lambda t} A\mathbf{v},$$

or

$$A\mathbf{v} = \lambda\mathbf{v}. \tag{84}$$

The solution of Eq. (211) is $\mathbf{x}(t) = e^{\lambda t}\mathbf{v}$ if, and only if, λ and $\mathbf{v}$ satisfy Eq. (214). A vector $\mathbf{v} \neq 0$ satisfying Eq. (214) is called an *eigenvector* of A with *eigenvalue* λ.

The eigenvalues λ of A are the roots of the equation

$$det(A - \lambda I) = det\begin{bmatrix} a_{11} - \lambda & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} - \lambda & \ldots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} - \lambda \end{bmatrix} = 0.$$

Case I Distinct eigenvalues

The matrix A has n linearly independent eigenvectors $\mathbf{v}^1, \ldots, \mathbf{v}^n$ with distinct eigenvalues $\lambda_1 \neq \lambda_2 \neq \ldots \lambda_{n-1} \neq \lambda_n$. For each eigenvalue λ_j we have an eigenvector $\mathbf{v}^j$ and a solution of Eq. (211) is of the form

$$\mathbf{x}^j(t) = e^{\lambda_j t}\,\mathbf{v}^j.$$

There are n linearly independent solutions $\mathbf{x}^j(t)$ of Eq. (211). Then the general solution of Eq. (211) is given by

$$\mathbf{x}(t) = c_1 e^{\lambda_1 t}\mathbf{v}^1 + c_2 e^{\lambda_2 t}\mathbf{v}^2 + \ldots + c_n e^{\lambda_n t}\mathbf{v}^n. \tag{85}$$

Case II Complex eigenvalues

If $\lambda = \alpha + i\beta$ is a complex eigenvalue of A with eigenvector $\mathbf{v} = \mathbf{v}^1 + i\mathbf{v}^2$, then a complex-valued solution of Eq. (211) is $\mathbf{x}(t) = e^{\lambda t}\mathbf{v}$.

Lemma [2] Let $\mathbf{x}(t) = \mathbf{y}(t) + i\mathbf{z}(t)$ be a complex-valued solution of Eq. (211). Then both $\mathbf{y}(t)$ and $\mathbf{z}(t)$ are real-valued solutions of Eq. (211).

The function $\mathbf{x}(t)$ can be written as

$$\begin{aligned}\mathbf{x}(t) &= e^{(\alpha+i\beta)t}(\mathbf{v}^1 + i\mathbf{v}^2)\\ &= e^{\alpha t}(cos\,\beta t + i\,sin\,\beta t)(\mathbf{v}^1 + i\mathbf{v}^2)\\ &= e^{\alpha t}[(\mathbf{v}^1 cos\,\beta t - \mathbf{v}^2 sin\,\beta t) + i(\mathbf{v}^1 sin\,\beta t + \mathbf{v}^2 cos\,\beta t)].\end{aligned}$$

If $\lambda = \alpha + i\beta$ is an eigenvalue of A with eigenvector $\mathbf{v} = \mathbf{v}^1 + i\mathbf{v}^2$, then

$$\mathbf{y}(t) = e^{\alpha t}(\mathbf{v}^1 cos\,\beta t - \mathbf{v}^2 sin\,\beta t)$$

and

$$\mathbf{z}(t) = e^{\alpha t}(\mathbf{v}^1 sin\,\beta t + \mathbf{v}^2 cos\,\beta t)$$

are two real-valued solutions of Eq. (211).

Case III Equal eigenvalues

If the matrix A does not have n distinct eigenvalues, then A may not have n linearly independent eigenvectors. Let us assume that the $n \times n$ matrix A has only $k < n$ linearly independent eigenvectors. In this case Eq. (211) has only k linearly independent solutions of the form $e^{\lambda t}\mathbf{v}$.

To find additional solutions we present the following method as described in [2]:

1. We pick an eigenvalue λ of A and find all vectors $\mathbf{v}$ for which $(A - \lambda I)^2\mathbf{v} = 0$, but $(A - \lambda I)\mathbf{v} \neq 0$. For each such vector $\mathbf{v}$

   $$e^{At}\mathbf{v} = e^{\lambda t}e^{(A-\lambda I)t} = e^{\lambda t}[\mathbf{v} + t(A - \lambda I)\mathbf{v}]$$

   is an additional solution of Eq. (211). The process is repeated for all eigenvalues of A.
2. If we still do not have enough solutions, then we find all vectors $\mathbf{v}$ for which $(A - \lambda I)^3\mathbf{v} = 0$, but $(A - \lambda I)^2\mathbf{v} \neq 0$. For each such vector $\mathbf{v}$,

   $$e^{At}\mathbf{v} = e^{\lambda t}[\mathbf{v} + t(A - \lambda I)\mathbf{v} + \frac{t^2}{2!}(A - \lambda I)^2\mathbf{v}]$$

   is an additional solution of Eq. (211).
3. We keep proceeding in this fashion until n linearly independent solutions are obtained.

Fundamental Solution Matrix. Definition A matrix $\mathbf{X}(t)$ whose columns are $\mathbf{x}^1(t), \ldots, \mathbf{x}^n(t)$, the n linearly independent solutions of Eq. (208)

$$\mathbf{X}(t) = [\mathbf{x}^1(t)|\mathbf{x}^2(t)|\ldots|\mathbf{x}^n(t)].$$

is called the fundamental solution matrix of Eq. (208) Every solution $\mathbf{x}(t)$ can be written in the form

$$\mathbf{x}(t) = c_1\mathbf{x}^1(t) + c_2\mathbf{x}^2(t) + \ldots + c_n\mathbf{x}^n(t) \tag{86}$$

In the matrix vector form, equation (224) can be written as $\mathbf{x}(t) = \mathbf{X}(t)\mathbf{c}$, where $\mathbf{c}$ is a constant vector.

Example [2]. Find a fundamental matrix solution of the system of differential equations

$$\dot{\mathbf{x}} = \begin{bmatrix} 1 & -1 & 4 \\ 3 & 2 & -1 \\ 2 & 1 & -1 \end{bmatrix}\mathbf{x}.$$

It can be verified that the three linearly independent solutions of the system are given by

$$e^t\begin{bmatrix} -1 \\ 4 \\ 1 \end{bmatrix}, \quad e^{3t}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \quad \text{and} \quad e^{-2t}\begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}.$$

Therefore, the fundamental matrix solution for the system is

$$\mathbf{X}(t) = \begin{bmatrix} -e^t & e^{3t} & -e^{-2t} \\ 4e^t & 2e^{3t} & e^{-2t} \\ e^t & e^{3t} & e^{-2t} \end{bmatrix}.$$

Theorem [2] Let $\mathbf{X}(t)$ be a fundamental solution matrix of the differential equation $\dot{\mathbf{x}} = A\mathbf{x}$. Then

$$e^{At} = \mathbf{X}(t)\mathbf{X}^{-1}(0). \tag{87}$$

where e^{At} is also a fundamental solution matrix.

We consider the example given in [2] and show as to how e^{At} can be computed. In Eq. (208) let

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 3 & 2 \\ 0 & 0 & 5 \end{bmatrix}.$$

The eigenvalues are computed from the relation

$$p(\lambda) = det(A - \lambda I) = det\begin{bmatrix} 1-\lambda & 1 & 1 \\ 0 & 3-\lambda & 2 \\ 0 & 0 & 5-\lambda \end{bmatrix} = (1-\lambda)(3-\lambda)(5-\lambda).$$

Thus we have 3 distinct eigenvalues $\lambda = 1$, $\lambda = 3$, and $\lambda = 5$. The eigenvectors corresponding to those eigenvalues, respectively, are

$$\mathbf{v}^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\mathbf{v}^2 = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}\mathbf{v}^3 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}.$$

The three linear independent solutions of $\dot{\mathbf{x}} = A\mathbf{x}$ are

$$\mathbf{x}^1(t) = e^t\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{x}^2(t) = e^{3t}\begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} \quad \mathbf{x}^3(t) = e^{5t}\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}.$$

The fundamental solution matrix is

$$\mathbf{X}(\mathbf{t}) = \begin{bmatrix} e^t & e^{3t} & e^{5t} \\ 0 & 2e^{3t} & 2e^{5t} \\ 0 & 0 & 2e^{5t} \end{bmatrix}.$$

We compute

$$\mathbf{X}^{-1}(0) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix},$$

and from the theorem

$$e^{At} = \mathbf{X}(t)\mathbf{X}^{-1}(0) = \begin{bmatrix} e^t & -\frac{1}{2}e^t + \frac{1}{2}e^{3t} & -\frac{1}{2}e^{3t} + \frac{1}{2}e^{5t} \\ 0 & e^{3t} & -e^{3t} + e^{5t} \\ 0 & 0 & e^{5t} \end{bmatrix}.$$

The Nonhomogeneous Equation. The initial-value problem for a nonhomogeneous equation is

$$\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{f}(t), \mathbf{x}(t_0) = \mathbf{x}^0. \tag{88}$$

Applying variation of parameter method, the solution is assumed of the form

$$\mathbf{x}(t) = \mathbf{X}(t)\mathbf{u}(t),$$

where $\mathbf{X}(t) = [\mathbf{x}^1(t), \ldots, \mathbf{x}^n(t)]$, and

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_n(t) \end{bmatrix}.$$

Using this relation Eq. (236) yields

$$\dot{\mathbf{X}}(t)\mathbf{u}(t) + \mathbf{X}(t)\dot{\mathbf{u}}(t) = A\mathbf{X}(t)\mathbf{u}(t) + \mathbf{f}(t). \tag{89}$$

Since matrix $\mathbf{X}(t)$ satisfies

$$\dot{\mathbf{X}}(t) = A\mathbf{X}(t), \tag{90}$$

we obtain

$$\mathbf{X}(t)\dot{\mathbf{u}}(t) = \mathbf{f}(t). \tag{91}$$

Matrix $\mathbf{X}(t)$ is nonsingular ($\mathbf{X}^{-1}(t)$ exists) and therefore

$$\dot{\mathbf{u}}(t) = \mathbf{X}^{-1}(t)\mathbf{f}(t). \tag{92}$$

Integrating this expression between t_0 and t we have

$$\mathbf{u}(t) = \mathbf{u}(t_0) + t_0 \int \mathbf{X}^{-1}(s)\mathbf{f}(s)ds \tag{93}$$

$$= \mathbf{X}^{-1}(t_0)\mathbf{x}^0 + t_0 \int \mathbf{X}^{-1}(s)\mathbf{f}(s)ds. \tag{94}$$

Consequently,

$$\mathbf{x}(t) = \mathbf{X}(t)\mathbf{X}^{-1}(t_0)\mathbf{x}^0 + \mathbf{X}(t)t_0 \int \mathbf{X}^{-1}(s)\mathbf{f}(s)ds. \tag{95}$$

If $\mathbf{X}(t) = e^{At}$ then

$$\mathbf{x}(t) = e^{A(t-t_0)}\mathbf{x}^0 + t_0 \int e^{A(t-s)}\mathbf{f}(s)ds. \tag{96}$$

Example. Find the solution of the initial value problem

$$\dot{\mathbf{x}} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\mathbf{x} + \begin{bmatrix} e^{-t} \\ 0 \end{bmatrix}, \quad \mathbf{x}^0 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

From the homogeneous problem we can easily show that the fundamental solution matrix is given by

$$\mathbf{X}(t) = \begin{bmatrix} e^t & te^t \\ 0 & e^t \end{bmatrix}.$$

It is easily verified that $\dot{\mathbf{X}} = A\mathbf{X}$ and $\mathbf{X}(0) = \mathbf{I}$.

$$\mathbf{X}^{-1}(s) = \begin{bmatrix} 1 & -s \\ 0 & 1 \end{bmatrix}e^{-s}.$$

and

$$0 \int \mathbf{X}(t)\mathbf{X}^{-1}(s)\mathbf{f}(s)ds = \begin{bmatrix} \frac{1}{2}(e^t - e^{-t}) \\ 0 \end{bmatrix}.$$

Then from Eq. (245) the solution is given by

$$\mathbf{x}(t) = \begin{bmatrix} (t-1)e^t \\ e^t \end{bmatrix} + \begin{bmatrix} \frac{1}{2}(e^t - e^{-t}) \\ 0 \end{bmatrix}.$$

Equilibrium and Stability

Consider the differential equation

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}), \tag{97}$$

where

$$\mathbf{x} = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix},$$

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt},$$

and

$$\mathbf{f}(t, \mathbf{x}) = \begin{bmatrix} f_1(t, x_1, \ldots, x_n) \\ \vdots \\ f_n(t, x_1, \ldots, x_n) \end{bmatrix},$$

is a nonlinear function. In general, Eq. (252)cannot be solved explicitly. However, one can easily determine the qualitative properties of solution of Eq. (252) in the neighborhood of an equilibrium point.

The *equilibrium points* are the values

$$\mathbf{x}^0 = \begin{bmatrix} x_1^0 \\ \vdots \\ x_n^0 \end{bmatrix}$$

for which, $\mathbf{x}(t) = \mathbf{x}^0$ is a solution of Eq. (252).

Observe that $\dot{\mathbf{x}}(t)$ is identically zero if $\mathbf{x}(t) \equiv \mathbf{x}^0$. The value $\mathbf{x}^0$ is an equilibrium of Eq. (252), if, and only if,

$$\mathbf{f}(t, \mathbf{x}^0) \equiv 0. \tag{98}$$

Example [6]. Find all equilibrium values of the system of differential equations

$$\frac{dx_1}{dt} = 1 - x_2, \quad \frac{dx_2}{dt} = x_1^3 + x_2.$$

The value

$$\mathbf{x}^0 = \begin{bmatrix} x_1^0 \\ x_2^0 \end{bmatrix}$$

is an equilibrium value if, and only if, $1 - x_2^0 = 0$ and $(x_1^0)^3 + x_2^0 = 0$. This yields $x_2^0 = 1$ and $x_1^0 = -1$. Hence $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ is the only equilibrium solution of this system.

Stability: Let $\phi(t)$ be a known solution of Eq. (252). Suppose that $\psi(t)$ is a second solution with $\psi(0)$ very close to $\phi(0)$ such that $\beta(t) \equiv \psi(t) - \phi(t)$ can be viewed as the disturbance on $\phi(t)$.

The concept of stability is important in many applications.

Consider the equation of motion of a simple pendulum of mass m and length l given by

$$\frac{d^2y}{dt^2} + \frac{g}{l} \sin y = 0,$$

where y is the angular displacement from the vertical axis and g is acceleration due to gravity. With the notation $x_1 = y$ and $x_2 = dy/dt$ we have

$$\frac{dx_1}{dt} = x_2, \quad \frac{dx_2}{dt} = -\frac{g}{l} \sin x_1. \tag{99}$$

The system of Eq. (261) has equilibrium solutions $\{x_1 = 0, x_2 = 0\}$, and $\{x_1 = \pi, x_2 = 0\}$.

If we disturb the pendulum slightly from the equilibrium position $\{x_1 = 0, x_2 = 0\}$, then it will oscillate with small amplitude about $x_1 = 0$.

If we disturb the pendulum slightly from the equilibrium position $\{x_1 = \pi, x_2 = 0\}$, then it will either oscillate with very large amplitude about $x_1 = 0$, or it will rotate around and around.

The two solutions have very different properties, and, intuitively, we would say that the equilibrium value $\{x_1 = 0, x_2 = 0\}$ is stable, while the equilibrium point $\{x_1 = \pi, x_2 = 0\}$ is unstable.

In the case when $\mathbf{f}(t, \mathbf{x})$ does not depend explicitly on t i.e. $\mathbf{f} = \mathbf{f}(\mathbf{x})$ the differential equations are called *autonomous*.

Phase-Plane

Let us consider a two dimensional system

$$\frac{dx}{dt} = f(x, y), \quad \frac{dy}{dt} = g(x, y). \tag{100}$$

Every solution $x = x(t)$, and $y = y(t)$ of Eq. (262) defines a curve in the three-dimensional space $\{t, x, y\}$.

For example the solution of the system of differential equations

$$\frac{dx}{dt} = -y, \quad \frac{dy}{dt} = x,$$

is $x = \cos t, y = \sin t$. This solution describes a helix in three-dimensional space $\{t, x, y\}$.

Every solution $x = x(t)$, and $y = y(t)$, of Eq. (262), for $t_0 \le t \le t_1$, also defines a curve in the $x - y$ plane. This curve is called the *orbit*, or *trajectory*, of the solution $x = x(t)$, $y = y(t)$, and the xy plane is called the *phase-plane* of the solutions of Eq. (262).

In the general case let $\mathbf{x}(t)$ be a solution of the vector differential equation

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_n(x_1, \ldots, x_n) \end{bmatrix} \tag{101}$$

on the interval $t_0 \le t \le t_1$. As t runs from t_0 to t_1, the set of points $(x_1(t), \ldots, x_n(t))$ trace out a curve C in the n-dimensional space $x_1, x_2, \ldots, x_n$. This curve is called the orbit of the solution $\mathbf{x} = \mathbf{x}(t)$, for $t_0 \le t \le t_1$, and the n-dimensional space $x_1, \ldots, x_n$ is called the "phase-space" or "state-space" of the solution of Eq. (264).

Linear Approximation at Equilibrium Points [5]

Consider again the Eq. (262)

$$\frac{dx}{dt} = f(x, y), \quad \frac{dy}{dt} = g(x, y),$$

with $f(0, 0) = g(0, 0) = 0$ as the equilibrium point. Using Taylor expansion about this point, we can write

$$f(x, y) = ax + by + P(x, y), \quad g(x, y) = cx + dy + Q(x, y),$$

where $P(x, y) = \mathbf{O}(r^2)$ and $Q(x, t) = \mathbf{O}(r^2)$ as $r = \sqrt{x^2 + y^2} \to 0$, and

$$a = \frac{\partial f}{\partial x}(0, 0), \quad b = \frac{\partial f}{\partial y}(0, 0), \tag{102}$$

$$c = \frac{\partial g}{\partial x}(0, 0), \quad d = \frac{\partial g}{\partial y}(0, 0). \tag{103}$$

The *linear approximation* of Eq. (262) in the neighbourhood of the origin is defined as the system

$$\dot{x} = ax + by, \quad \dot{y} = cx + dy,$$

or

$$\dot{x} = \mathbf{A}\,\mathbf{x} \tag{104}$$

where

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}. \tag{105}$$

The solutions of Eq. (270) are geometrically similar to those of Eq. (262) near the origin unless one (or more) of the eigenvalues of $\mathbf{A}$ is zero or has zero real part.

The two linearly independent solutions are of the form

$$\mathbf{x} = \mathbf{u}\, e^{\lambda t}, \tag{106}$$

where

$$\mathbf{u} = \begin{bmatrix} r \\ s \end{bmatrix} \neq 0. \tag{107}$$

Then

$$\dot{\mathbf{x}} = \lambda\, \mathbf{u}\, e^{\lambda t},$$

and equations (270) and (272) yield

$$(A - \lambda I)\mathbf{u} = 0 \tag{108}$$

where I is the identity matrix. With $\mathbf{u} \neq 0$ and Eq. (275), we have

$$det(A - \lambda I) = 0,$$

or

$$\begin{vmatrix} a-\lambda & b \\ c & d-\lambda \end{vmatrix} = 0. \tag{109}$$

The two eigenvalues are given by the solution of the quadratic equation

$$\lambda^2 - (a+d)\lambda + (ad-bc) = 0. \tag{110}$$

The solutions of Eq. (275) are the *eigenvectors*: $\mathbf{u}_1$ corresponding to λ_1, and $\mathbf{u}_2$ corresponding to λ_2. The general solution of Eq. (270) is

$$\mathbf{x} = C_1\mathbf{u}_1 e^{\lambda_1 t} + C_2\mathbf{u}_2 e^{\lambda_2 t}, \quad \text{for} \quad \lambda_1 \neq \lambda_2. \tag{111}$$

Using the nonsingular linear transformation

$$\mathbf{x}_1 = S\mathbf{x}; \quad S = [\mathbf{u}_1\, \mathbf{u}_2], \tag{112}$$

Eq. (270) becomes

$$\dot{\mathbf{x}}_1 = SAS^{-1}\mathbf{x}_1 = B\mathbf{x}_1, \tag{113}$$

where B is diagonal or in Jordan form. The topological character of the transformed equilibrium point at the origin is not affected in the new variable $\mathbf{x}_1 = [x_1, y_1]^T$. The equations in the new coordinates are simpler.

Case I $\lambda_1 \neq \lambda_2 \neq 0$ and $\lambda_1, \lambda_2 \in \mathcal{R}$ (real)
We can choose S so that

$$\dot{x}_1 = \lambda_1 x_1, \quad \dot{y}_1 = \lambda_2 y_1,$$

and then the equation for the phase paths is

$$\frac{dy_1}{dx_1} = \frac{\lambda_2}{\lambda_1}\frac{y_1}{x_1}.$$

The solutions are

$$y_1 = C|x_1|^{\lambda_2/\lambda_1}, \quad \text{where} \quad C = \text{arbitrary}.$$

The origin is a *node* (Figure 1) when $\lambda_2/\lambda_1 > 0$. The node is stable when $\lambda_1, \lambda_2 < 0$ (Figure 1) and unstable when $\lambda_1, \lambda_2 > 0$.
The origin is a *saddle-point* (Figure 2) when $\lambda_2/\lambda_1 < 0$.

Case II $\lambda_1 = \lambda_2 = \lambda$ (b and c not both zero)
We can choose S so that

$$\dot{x}_1 = \lambda x_1 + y_1, \quad \dot{y}_1 = \lambda y_1, \quad \lambda \in \mathcal{R},$$

and then the equation for the phase paths is

$$\frac{dy_1}{dx_1} = \frac{\lambda y_1}{\lambda x_1 + y_1}.$$

The solutions are

$$y_1 = 0, \quad x_1 = \frac{1}{\lambda} y_1 log_e|y_1| + Cy_1 \text{ where } C = \text{arbitrary}.$$

The origin is a *inflected node*, stable if $\lambda < 0$ (Figure 3) and unstable if $\lambda > 0$.

Case III $\lambda_1 = \overline{\lambda}_2 = \alpha + \mathrm{i}\beta$ with $\beta \neq 0$
We can choose S so that the equations become

$$\dot{x}_1 = \alpha x_1 - \beta y_1, \quad \dot{y}_1 = \beta x_1 + \alpha y_1.$$

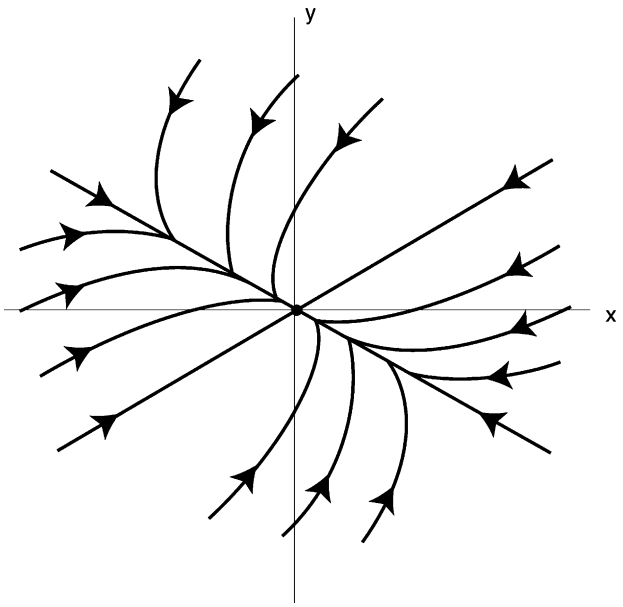


Figure 1. Stable node

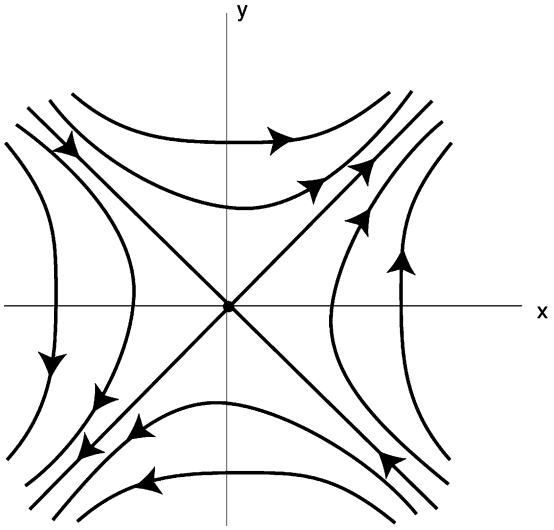


Figure 2. Saddle point

With $z(t) = x_1(t) + \mathrm{i}\, y_1(t) = r(t)e^{\mathrm{i}\theta(t)}$ we have $\dot{z} = (\alpha + \mathrm{i}\beta)z$, and $r(t) = |z(t)|$. The equations in polar coordinates are

$$\dot{r} = \alpha r, \quad \dot{\theta} = \beta.$$

The origin is a stable *spiral* (or *focus*) if $\alpha < 0$, $\beta \neq 0$ (Figure 4), and an unstable spiral if $\alpha > 0$, $\beta \neq 0$. the origin is a *center* if $\alpha = 0$, $\beta \neq 0$, (Figure 5).

We can sumarize all the above cases in the following table [5].

(1) λ_1, λ_2	real, unequal, same sign	Node
(2) $\lambda_1 = \lambda_2$	(real) $b \neq 0, c \neq 0$	Inflected node
(3) λ_1, λ_2	complex, non-zero real part	Spiral
(4) $\lambda_1 \neq 0, \lambda_2 = 0$		Parallel lines
(5) λ_1, λ_2	real, different sign	Saddle point
(6) λ_1, λ_2	pure imaginary	Center

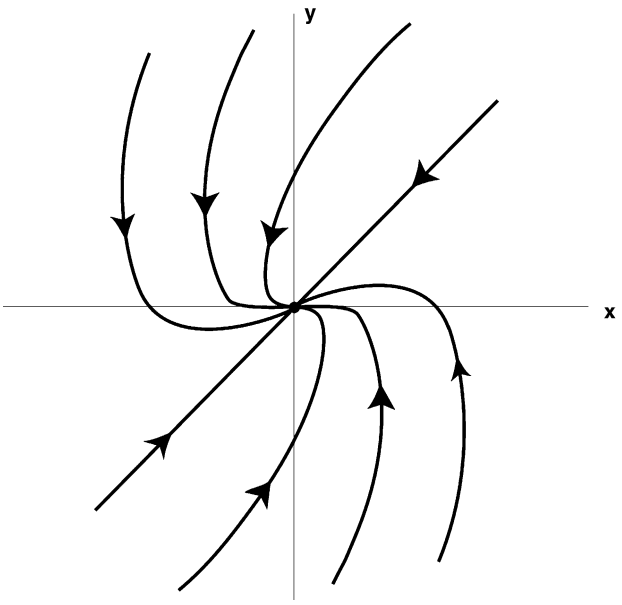


Figure 3. Stable inflected node

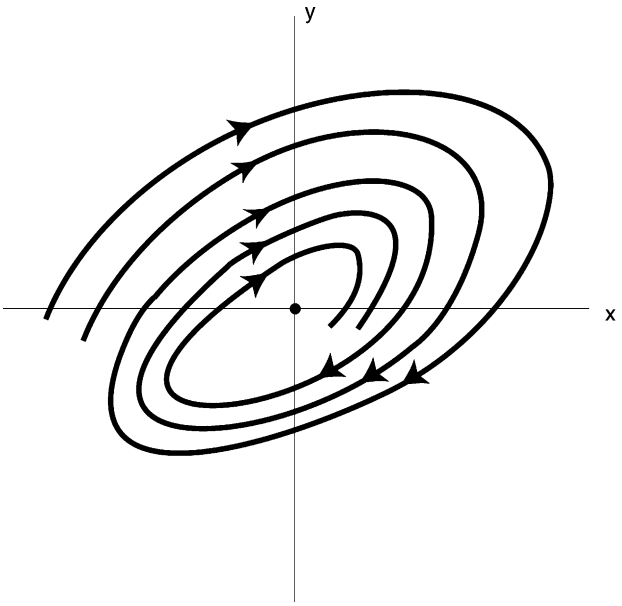


Figure 4. Stable spiral

Example. Classify the equilibrium point at (0,0) for the system

$$\dot{x} = e^{-x-3y} - 1, \quad \dot{y} = -x(1 - y^2).$$

Using Taylor expansion for the exponential function, the linearized system of equations about (0,0) is

$$\dot{x} = -x - 3y, \quad \dot{y} = -x,$$

or in matrix form

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -1 & -3 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

The eigenvalues are $\lambda_{1,2} = \dfrac{-1 \pm \sqrt{17}}{2}$ are real with different sign. The equilibrium is a saddle point.

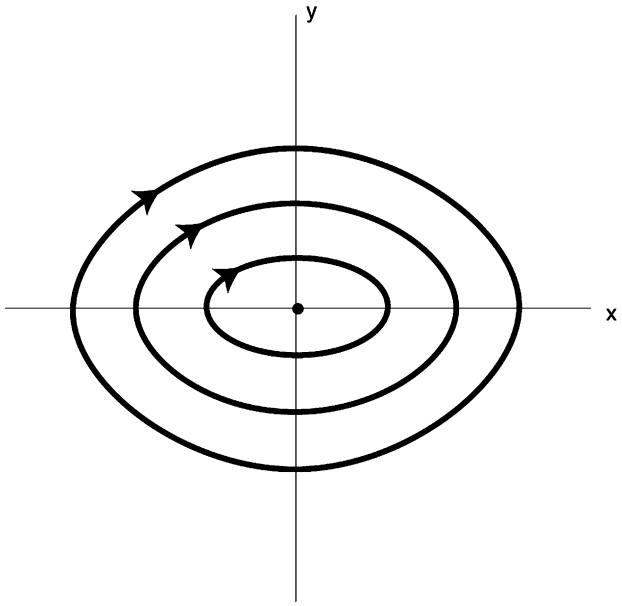


Figure 5. Center

PARTIAL DIFFERENTIAL EQUATIONS

The word "ordinary" in *ordinary differential equation* distinguishes it from *partial differential equation* (PDE), involves partial derivatives of two or more independent variables. For a first order partial differential equation, a unified general theory exists; however, this a case for higher order partial differential equations. Generally speaking, the second order PDEs may be classified into three following categories, viz., elliptic, hyperbolic, and parabolic types.

Normal Forms of Elliptic, Hyperbolic, and Parabolic Equations

Consider a linear second order differential operator for the function $u(x, y)$ given by

$$L(u) = a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2}, \tag{114}$$

where a, b, and c are either constants or functions of x and y. A corresponding quasilinear PDE may be represented by

$$L(u) + g(x, y, \partial u/\partial x, \partial u/\partial y) = L(u) + \ldots\ldots = 0, \tag{115}$$

where $g(x, y, \partial u/\partial x, \partial u/\partial y)$ is not necessarily linear and does not contain any second derivative.

Let us introduce the transformations

$$\begin{aligned} \xi &= \alpha x + \beta y, \\ \eta &= \gamma x + \delta y. \end{aligned} \tag{116}$$

Therefore, $L(u)$ in Eq. (293) takes the form

$$\begin{aligned} L(u) = {} & (a\alpha^2 + b\alpha\beta + c\beta^2)\frac{\partial^2 u}{\partial \xi^2} \\ & + (2a\alpha\gamma + b(\alpha\delta + \beta\gamma) + 2c\beta\delta)\frac{\partial^2 u}{\partial \xi \partial \eta} \\ & + (a\gamma^2 + b\gamma\delta + c\delta^2)\frac{\partial^2 u}{\partial \eta^2}. \end{aligned} \tag{117}$$

If the transformed operator is desired to be of the form $\frac{\partial^2 u}{\partial\xi\partial\eta}$, then we need

$$a\alpha^2 + b\alpha\beta + c\beta^2 = 0, \tag{118}$$

$$\alpha\gamma^2 + b\gamma\delta + c\delta^2 = 0. \tag{119}$$

If $a = c = 0$, then the trivial transformation $\xi = y$ and $\eta = y$ provides the desired form. For the non-trivial case either a or c or both are non-zero. Let us say $a \neq 0$, thereby implying that $\alpha \neq 0$, $\gamma \neq 0$. Dividing Eq. (297) by β^2 and Eq. (298) by δ^2, we obtain two quadratic equations in (α/β) and (γ/δ). These yield

$$\alpha/\beta = \frac{1}{2a}\{-b \pm \sqrt{b^2 - 4ac}\}, \tag{120}$$

$$\gamma/\delta = \frac{1}{2a}\{-b \pm \sqrt{b^2 - 4ac}\}. \tag{121}$$

The ratios α/β and γ/δ must be different (by choosing positive sign in Eq. (299) and negative sign in Eq. (300)) so that the transformation given by Eq. (295) is non-singular. Further $b^2 - 4ac$ should be positive.

Therefore, $L(u)$ reduces to the form $\frac{\partial^2 u}{\partial\xi\partial\eta}$ if and only if

$$b^2 - 4ac > 0, \tag{122}$$

and this case is said to be "hyperbolic". Then the transformation Eq. (295) takes the form

$$\begin{aligned} \xi &= (-b + \sqrt{b^2 - 4ac})x + 2ay, \\ \eta &= (-b - \sqrt{b^2 - 4ac})x + 2ay. \end{aligned} \tag{123}$$

Then the PDE given by Eq. (294) reduces to

$$-4a(b^2 - 4ac)\frac{\partial^2 u}{\partial\xi\partial\eta} + g(\xi, \eta, \frac{\partial u}{\partial\xi}, \frac{\partial u}{\partial\eta}) = 0. \tag{124}$$

If $b^2 - 4ac = 0$, then L is termed as "parabolic". In this case Eq. (299) and Eq. (300) reduce to a single equation and $\alpha/\beta = -b/2a$ forces the coefficient of $\partial^2 u/\xi^2$ in Eq. (296) to vanish. Further, since $b^2 = 4ac$ or $b/2a = 2c/b$, the coefficient of $\frac{\partial^2 u}{\partial\xi\partial\eta}$ also vanish. Thus the transformation (c.f. Eq. (302))

$$\begin{aligned} \xi &= -bx + 2ay, \\ \eta &= x(\text{arbitrary}), \end{aligned} \tag{125}$$

can be used to transform Eq. (294) into

$$a\frac{\partial^2}{\partial\eta^2} + g() = 0. \tag{126}$$

This is the normal form of a parabolic quasilinear PDE.

For the final case, $b^2 - 4ac < 0$, and the operator $L(u)$ is said to be "elliptic". In this case it is not possible to eliminate the coefficients of $\frac{\partial^2 u}{\partial\xi^2}$ or $\frac{\partial^2 u}{\partial\eta^2}$. Nevertheless, if we use the transformation

$$\xi = \frac{2ay - bx}{\sqrt{4ac - b^2}}, \quad \eta = t \quad (\text{arbitrary}), \tag{127}$$

then $L(u) = a(\frac{\partial^2 u}{\partial\xi^2} + \frac{\partial^2 u}{\partial\eta^2})$, and the general PDE has the form

$$a(\frac{\partial^2 u}{\partial\xi^2} + \frac{\partial^2 u}{\partial\eta^2}) + g(\xi, \eta, \frac{\partial u}{\partial\xi}, \frac{\partial u}{\partial\eta}) = 0. \tag{128}$$

For the linear case

$$\frac{\partial^2 u}{\partial\xi^2} + \frac{\partial^2 u}{\partial\eta^2} = 0. \tag{129}$$

which is the well-known Laplace's equation.

Once a PDE has been reduced to its normal form, the method of characteristic may be effectively used to find its solution.

However, in the following we discuss the solution of a particular hyperbolic equation, known as the "wave equation" by the use of "separation of the variables" which is a popular approach in engineering.

The equation

$$c^2\frac{\partial^2 u(x,t)}{\partial x^2} - «u(x,t) = 0, \quad c = \text{constant}, \tag{130}$$

is a partial differential equation. The following notation was used

$$«u(x,t) = \frac{\partial^2 u(x,t)}{\partial t^2}.$$

The initial conditions are

$$u(x,0) = f(x), \quad \dot{u}(x,0) = g(x). \tag{131}$$

The boundary conditions are

$$\frac{\partial u}{\partial x}(0,t) = \frac{\partial u}{\partial x}(l,t) = 0. \tag{132}$$

We seek the solution of Eq. (309) in the form of a product of a function of time and a function of position

$$u(x,t) = U(x)\varphi(t). \tag{133}$$

Introducing (313) into (309), we replace Eq. (309) by the system of two ordinary equations

$$«\varphi + \beta^2 c^2\varphi = 0, \tag{134}$$

$$\frac{d^2U}{dx^2} + \beta^2 U = 0, \tag{135}$$

where β is for the time being an undetermined parameter. The solution of Eqs. (314) and (315) is

$$\varphi(t) = A\sin\omega t + B\cos\omega t, \tag{136}$$

$$U(x) = C\sin\beta x + D\cos\beta x, \tag{137}$$

where $\omega = \beta c$.

We first consider the second boundary conditions (312). They imply that $C = 0$ and

$$D\beta\sin\beta l = 0.$$

The latter condition is satisfied if

$$\beta_n = \frac{n\pi}{l}, \quad (n = 0, 1, 2, \ldots, \infty). \tag{138}$$

It is evident that every value of β_n is associated with a particular solution of Eq. (309), viz.

$$u_n(x,t) = (A_n\sin\omega_n t + B_n\cos\omega_n t)D\cos\beta_n x. \tag{139}$$

The general solution of (309) takes the form

$$u(x,t) = n=0 \sum u_n(x,t) = n=0 \sum (A_n \sin \omega_n t + B_n \cos \omega_n t) D \cos \beta_n x. \quad (140)$$

The constants A_n, B_n are to be found from the initial conditions (311) i.e.

$$f(x) = n=0 \sum D B_n \cos \beta_n x, \quad g(x) = n=0 \sum D \omega_n A_n \cos \beta_n x. \quad (141)$$

The functions

$$U_n(x) = D \cos \beta_n x, \quad (142)$$

are the eigenfunctions of the problem. They are orthogonal, i.e.

$$\begin{aligned} &0 \int U_n(x) U_m(x) dx = 0, \quad \text{if } n \neq m, \\ &0 \int U_n^2(x) dx = \frac{l}{2} D^2, \quad \text{if } n = m, \end{aligned} \quad (143)$$

as can easily be verified by integration. The constant D is arbitrary.

Assume that $D^2 = 2/l$. Then $0 \int U_n^2(x) dx = 1$ and the eigenfunctions

$$U_n(x) = \sqrt{\frac{2}{l}} \cos \beta_n x = \sqrt{\frac{2}{l}} \cos \frac{n \pi x}{l}, \quad (144)$$

are called normalized eigenfunctions.

Making use of the normalized eigenfunctions we can rewrite relations (322) in the form

$$f(x) = \sqrt{\frac{2}{l}} n=0 \sum B_n \cos \beta_n x, \quad g(x) = \sqrt{\frac{2}{l}} n=0 \sum \omega_n A_n \cos \beta_n x. \quad (145)$$

To find the coefficient B_n we multiply the first equation (326) by $\cos \beta_n x$ and integrate with respect to x from 0 to l. Then, making use of the orthogonality relations, we obtain

$$\begin{aligned} &B_n = \sqrt{\frac{2}{l}} 0 \int f(x) \cos \beta_n x \, dx, \quad (n = 1, 2, \ldots, \infty), \\ &B_0 = \frac{1}{2} \sqrt{\frac{2}{l}} 0 \int f(x) dx. \end{aligned} \quad (146)$$

Similarly we have

$$\begin{aligned} &A_n = \frac{1}{c \beta_n} \sqrt{\frac{2}{l}} 0 \int g(x) \cos \beta_n x \, dx, \quad (n = 1, 2, \ldots, \infty), \\ &A_0 = 0. \end{aligned} \quad (147)$$

Introducing the values of A_n, B_n into Eq. (321), we arrive at the final solution.

Example. We consider next the equation

$$c^2 \frac{\partial^2 u}{\partial x^2} - \ll u = 0, \quad (148)$$

with assuming homogeneous initial conditions ($u(x,0) = \dot{u}(x,0) = 0$) and the boundary conditions

$$u(0,t) = 0, \quad \frac{\partial u}{\partial x}(l,t) = P(t). \quad (149)$$

Performing the Laplace transform in Eq. (329) for the above boundary conditions, we obtain

$$c^2 \frac{d^2 \overline{u}}{dx^2} - s^2 \overline{u} = -s u(x,0) - \dot{u}(x,0), \quad (150)$$

$$\overline{u}(0,s) = 0; \frac{d\overline{u}}{dx}(l,s) = \overline{P}(s), \quad (151)$$

where

$$\overline{u}(x,s) = 0 \int e^{-st} u(x,t) dt, \quad \overline{P}(s) = 0 \int e^{-st} P(t) dt.$$

The right-hand side of Eq. (331) vanishes in view of the homogeneous initial conditions, hence its solution can be represented in the form

$$\overline{u}(x,s) = A(s) e^{-sx/c} + B(s) e^{sx/c}. \quad (152)$$

The functions $A(s)$, $B(s)$ can be determined by means of the boundary conditions (332):

$$\begin{aligned} &A(s) = -B(s), \\ &B(s) = \frac{\overline{P}(s) c}{2s \cosh \frac{sl}{c}}. \end{aligned} \quad (153)$$

Hence

$$\overline{u}(x,s) = \frac{\overline{P}(s) l}{2} \frac{e^{sx/c} - e^{-sx/c}}{\frac{sl}{c} \cosh \frac{sl}{c}},$$

i.e.

$$\overline{u}(x,s) = \frac{\overline{P}(s) l \sinh \frac{sx}{c}}{\frac{sl}{c} \cosh \frac{sl}{c}}. \quad (154)$$

Now we invert the Laplace transform in (337). Taking into account that

$$\begin{aligned} L^{-1}\left(\frac{\sinh \frac{sx}{c}}{s \cosh \frac{sl}{c}}\right) &= \frac{2}{\pi} n=1 \sum \frac{(-1)^{n-1}}{n - \frac{1}{2}} \sin\left[\left(n - \frac{1}{2}\right) \frac{\pi x}{l}\right] \\ &\quad \sin\left[\left(n - \frac{1}{2}\right) \frac{\pi l t}{c}\right], \end{aligned}$$

and

$$L^{-1} \overline{P}(s) = P(t),$$

and making use of the convolution theorem was obtain

$$\begin{aligned} u(x,t) &= \frac{2c}{\pi} n=1 \sum \frac{(-1)^{n-1}}{n - \frac{1}{2}} \sin\left[\left(n - \frac{1}{2}\right) \frac{\pi x}{l}\right] \\ &\quad 0 \int P(\tau) \sin\left[\frac{2n-1}{2} \frac{\pi l}{c} (t - \tau)\right] d\tau. \end{aligned} \quad (155)$$

In the particular case

$$P(t) = P_0 H(t),$$

where $H(t)$ is the Heaviside function, we have from Eq. (340)

$$\begin{aligned} u(x,t) &= \frac{8 P_0 c^2}{\pi^2 L} n=1 \sum \frac{(-1)^{n-1}}{(2n-1)^2} \sin \frac{2n-1}{2} \frac{\pi x}{l} \\ &\quad \left[1 - \cos \frac{(2n-1)\pi l t}{2c}\right]. \end{aligned} \quad (156)$$

Assume that $P(t) = P_0 e^{i\omega t}$ acts at the end $x = 0$ of the fixed rod. Taking into account that $u(x,t) = U(x) e^{i\omega t}$, we transform Eq. (329) to the form

$$c^2 \frac{d^2 U}{dx^2} + \omega^2 U = 0. \quad (157)$$

The boundary conditions take the form

$$U(0) = 0, \quad \frac{dU}{dx}(l) = P_0. \tag{158}$$

The constants A, B appearing in the solution

$$U(x) = A\, sin\, \frac{\omega x}{c} + B\, cos\, \frac{\omega x}{c}, \tag{159}$$

of Eq. (343) are determined from the boundary conditions (344). Finally we obtain

$$u(x,t) = \frac{P_0 c e^{i\omega t}}{\omega}\, \frac{sin\, \frac{\omega x}{c}}{cos\, \frac{\omega l}{c}}. \tag{160}$$

If the frequency ω approaches any of the eigenfrequency, the displacement u tends to infinity. Thus, we are faced with resonance.

APPLICATIONS

Problem 1. A sphere of mass m falls on a vertical spring as shown in the Figure 7. The sphere makes contact with the spring and the spring compresses. The compression phase ends when the velocity of the sphere is zero. Next phase is the restitution phase when the spring is expanding and the sphere is moving upward. At the end of the restitution phase there is the separation of the sphere.

Find and solve the equation of motion for the sphere in contact with the spring.

Solution The x-axis selected downward as shown in the Figure 7.

At the moment $t = 0$ it is assumed that the sphere gets in contact with the spring and has the velocity $\mathbf{v}(t=0) = \mathbf{v}_0 = v_0\, \mathbf{i}$.

Using Newton's second law, the equation of motion for the sphere in contact with the spring is:

$$m\mathbf{a} = \mathbf{G} + \mathbf{F}_e \quad \text{or} \quad m\ddot{x} = mg - kx. \tag{161}$$

The acceleration of the sphere is $\mathbf{a} = \ddot{x}\mathbf{i}$, where x is the linear displacement. The weight of the sphere is $\mathbf{G} = m\, g\, \mathbf{i}$, where g is the gravitational acceleration. The contact elastic force is $\mathbf{F}_e = -k\, x\, \mathbf{i}$, where k is the elastic constant of the spring. The initial conditions are

$$x(0) = 0 \quad \text{and} \quad \dot{x}(0) = v_0.$$

With the notation

$$\frac{k}{m} = \omega^2, \quad (\omega > 0),$$

Equation (347) becomes

$$\ddot{x} + \omega^2 x = g. \tag{162}$$

Assume the solution of Eq. (350) has the following expression

$$x = a\, cos(\omega t - \varphi_0) + b. \tag{163}$$

Then

$$\dot{x} = -a\omega\, sin(\omega t - \varphi_0) \quad \text{and} \quad \ddot{x} = -a\omega^2\, cos(\omega t - \varphi_0).$$

Substituting Eq. (354) into Eq. (350)

$$-a\omega^2\, cos(\omega t - \varphi_0) + \omega^2[a\, cos(\omega t - \varphi_0) + b] = g,$$

the constant b is obtained

$$b = \frac{g}{\omega^2}. \tag{164}$$

Using the initial conditions ($x(0) = 0$ and $\dot{x}(0) = v_0$) the following expressions are obtained

$$x(0) = a\, cos(-\varphi_0) + b = a\, cos\, \varphi_0 + b = 0,$$
$$\dot{x}(0) = -a\omega\, sin(-\varphi_0) = a\omega\, sin\, \varphi_0 = v_0,$$

or

$$a\, cos\, \varphi_0 = -b = -\frac{g}{\omega^2} \quad \text{and} \quad a\, sin\, \varphi_0 = \frac{v_0}{\omega}.$$

It results

$$a = \sqrt{\frac{g^2}{\omega^4} + \frac{v_0^2}{\omega^2}},$$
$$tan\, \varphi_0 = -\frac{v_0\omega}{g} \quad \text{or} \quad \varphi_0 = -arctan\, \frac{v_0\omega}{g}. \tag{165}$$

The relation for the displacement of the sphere is

$$x - \frac{g}{\omega^2} = (\sqrt{\frac{g^2}{\omega^4} + \frac{v_0^2}{\omega^2}})\cos(\omega t + \arctan\frac{v_0\omega}{g}). \tag{166}$$

If the sphere would be connected to the spring, it would oscillate around the position $x = \frac{g}{\omega^2}$.

The sphere reaches the maximum position on x-axis at $t = t_1$ when $\dot{x}(t_1) = 0$

$$\dot{x}(t_1) = -a\omega\, sin(\omega t_1 - \varphi_0) = 0 \quad \Rightarrow \quad \omega t_1 - \varphi_0 = \pi$$

or

$$t_1 = \frac{\pi}{\omega} + \frac{1}{\omega}\varphi_0 = \frac{\pi}{\omega} - \frac{1}{\omega}\, arctan\, \frac{v_0\omega}{g}. \tag{167}$$

At the moment $t = t_2 = 2\, t_1$, the sphere attains again the reference O. At this moment, the sphere separates itself and moves upward, and the spring compresses. The velocity of the sphere at $t = t_2$ is

$$\dot{x}(t_2) = a\omega\, sin(\omega t_2 - \varphi_0) = -v_0. \tag{168}$$

The contact time between the sphere and the spring is:

$$t_2 = 2t_1 = \frac{2\pi}{\omega} - \frac{2}{\omega}\, arctan\, \frac{v_0\omega}{g}. \tag{169}$$

The jump in velocity is

$$\Delta v = \dot{x}(0) - \dot{x}(t_2) = v_0 - (-v_0) = 2v_0. \tag{170}$$

The displacement at t_1 is

$$x(t_1) = a\, cos(\omega t_1 - \varphi_0) + b = a + b = \sqrt{\frac{g^2}{\omega^4} + \frac{v_0^2}{\omega^2}} + \frac{g}{\omega^2},$$

and the relative displacement is:

$$\lambda = x(0) - x(t_1) = 0 - x(t_1) = -(\frac{g}{\omega^2} + \sqrt{\frac{g^2}{\omega^4} + \frac{v_0^2}{\omega^2}}). \tag{171}$$

Numerical Example. The sphere with mass $m = 10$ kg falls falls from the height $h = 1$ m on the spring with the elastic constant $k = 294 \times 10^3$ N/m. The initial velocity of the sphere is $v_0 = \sqrt{2gh} = 4.42945$ m/s. The total time of contact t_2 is calculated with Eq. (362), $t_2 = 0.0184728$ s.

The relative displacement is $|\lambda| = 0.0261689$ m and the jump in velocity is calculated with Eq. (363), $\Delta = 8.85889$ m/s.

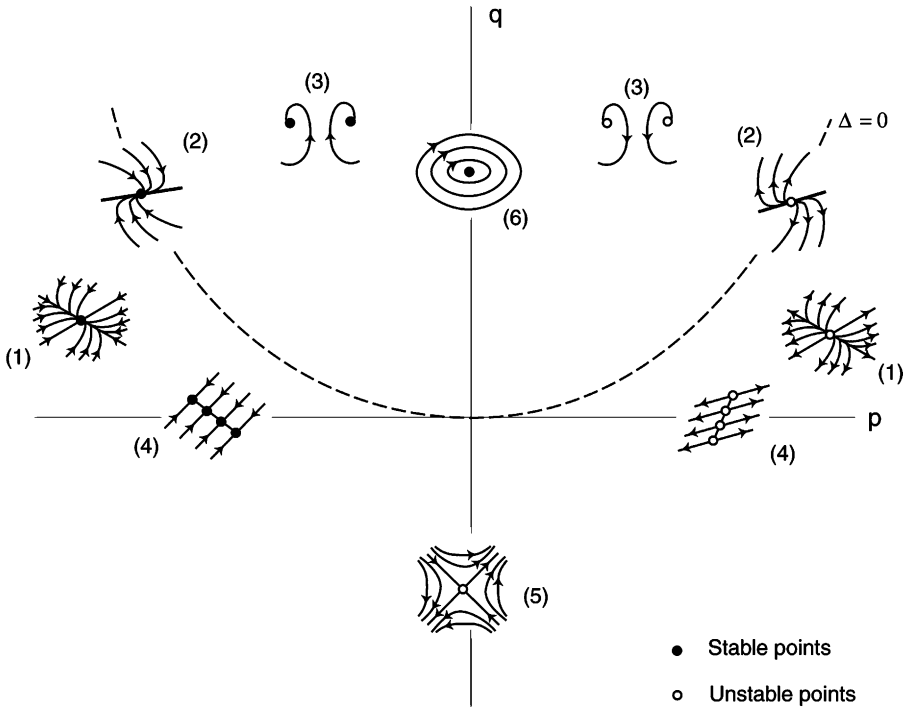


Figure 6. General classification

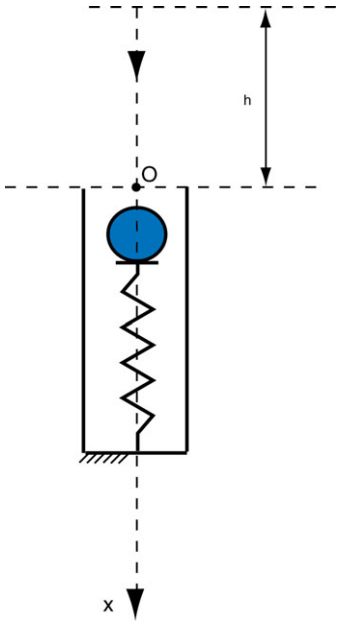


Figure 7. Sphe in contact with a spring

The maximum elastic force is $F_{e\,max} = k\ \ x(t_1) = k\ |\lambda| = 7693.65$ N. The maximum elastic force is approximative 76 greater then the weight of the sphere. For this dynamical problem the displacement of the sphere is very small, almost null, while the the jump in velocity is big.

Figure 8 represents the dependence of the displacement of the sphere with respect to time calculated with Eq. (358). Figure 9 shows the variation in time of the velocity of the sphere in contact with the spring. At $t = 0$ the sphere gets in contact with the spring and at $t = t_2$ the sphere separates from the spring. Note that the initial velocity is equal with the absolute value of the final velocity. Figure 10 shows the variation of the elastic force with respect to time.

Problem 2. A rod AB with the mass M and the length $6a$ is connected to the ground at the pin joint O as shown in Figure 11a. A mass m is attached to the rod at point A. The rod is connected to two springs, with the elastic constant k, as depicted in Figure 11a.

Determine the equation of motion of the system for small oscillations if the initial angular velocity of the rod is ω_0. The gravitational acceleration is g.

Solution At equilibrium the rod rotated around the pivot O with the angle θ_s (Figure 11b). The sum of the moments of the forces acting on the rod with respect to O are

$$\sum M_0^{\text{equil}} \quad \Rightarrow \quad mg(2a) + ka\theta_s a - Mga + k(4a)\theta_s(4a) = 0,$$

or

$$a(2mg - Mg + 17ka\theta_s) = 0. \tag{172}$$

The equation of motion of the rod in rotation is

$$-I_O\ddot{\theta} = M_O,$$

where I_O is the mass moment of inertia of the rod and mass m with respect to O

$$I_O = m(2a)^2 + \frac{M(6a)^2}{12} + Ma^2 = 4a^2(m + M). \tag{173}$$

Consider the rod in a position defined by the angle $(\theta_s + \theta)$ (Figure 11c). The sum of the moments with respect to the axis of rotation through O are

$$M_O = mg(2a) + ka(\theta_s + \theta)a - Mga + k(4a)(\theta_s + \theta)(4a).$$

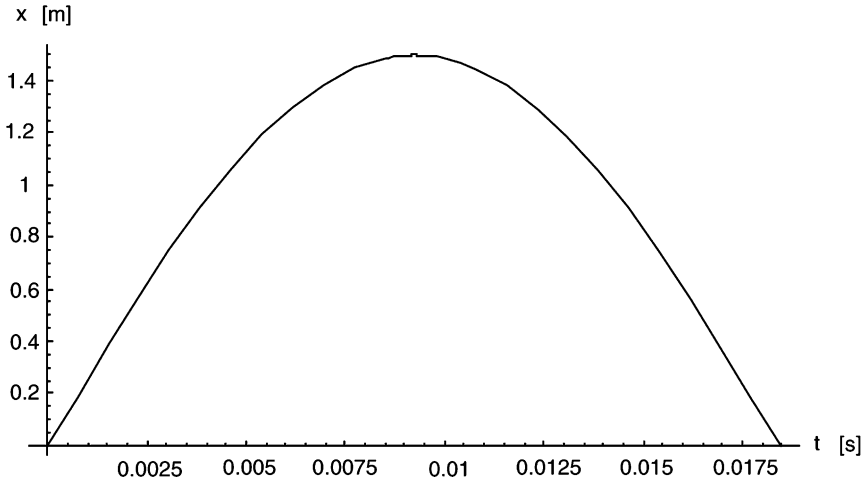


Figure 8. The displacement of the sphere

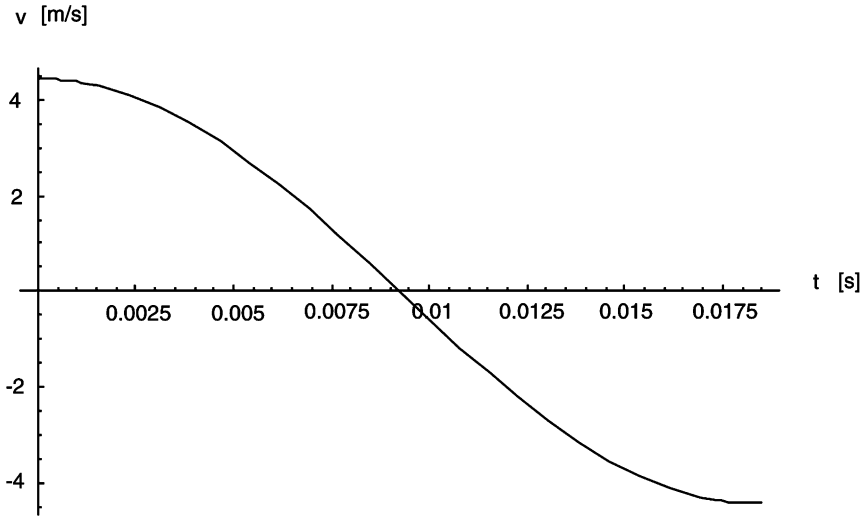


Figure 9. The velocity of the sphere

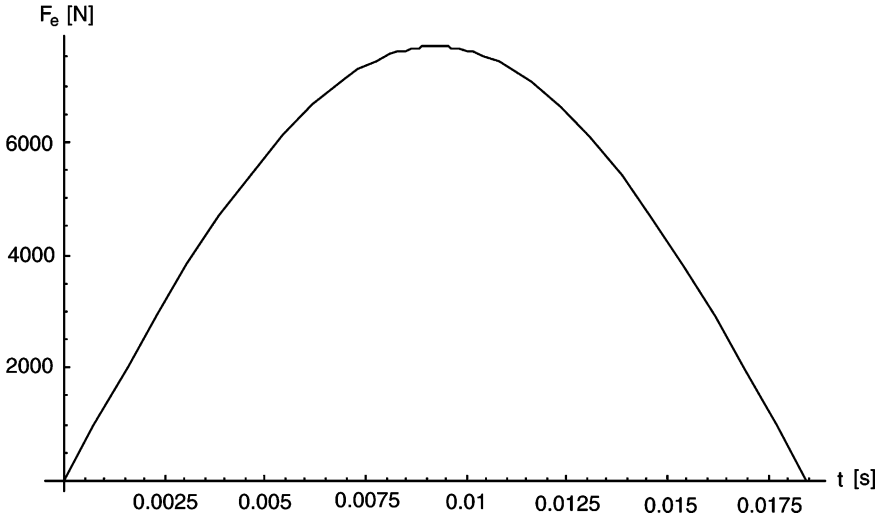


Figure 10. The elastic force

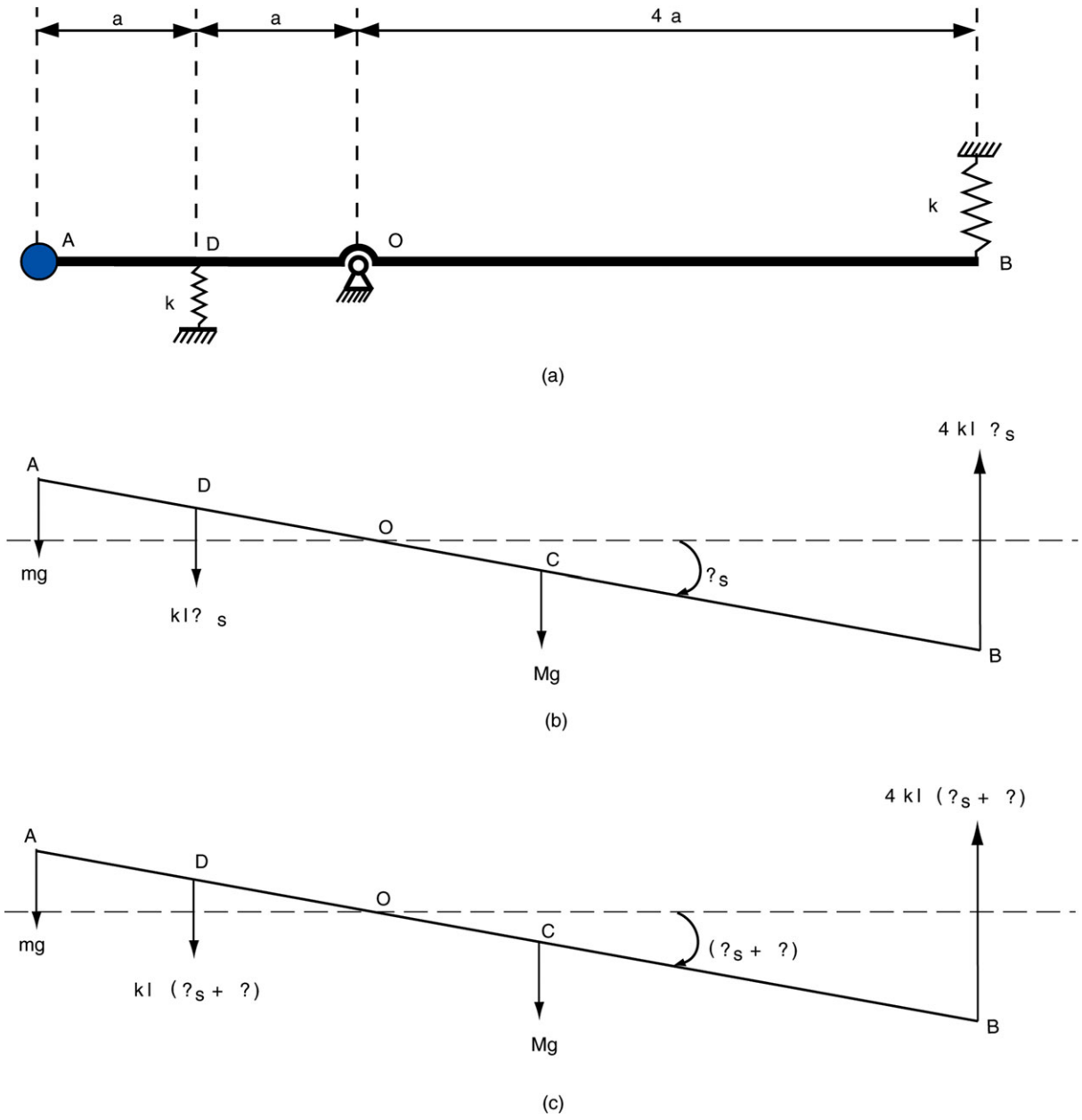


Figure 11. Small vibrations of a rod

With the equilibrium condition given by Eq. (367) the moment becomes

$$M_O = 17ka^2\theta, \tag{174}$$

and the equation of motion is

$$4a^2(m+M)\ddot{\theta} + 17ka^2\theta = 0,$$

or

$$\ddot{\theta} + \frac{17k}{4(m+M)}\theta = 0. \tag{175}$$

This is the equation of a free harmonic vibration (small oscillation) with the circular frequency

$$\omega = \sqrt{\frac{17k}{4(m+M)}} = \frac{1}{2}\sqrt{\frac{17k}{m+M}}.$$

The period of small oscillation is:

$$T = \frac{2\pi}{\omega} = 4\pi\sqrt{\frac{m+M}{17k}}.$$

The general solution of the differential equation Eq. (373) is:

$$\theta = C_1 \cos\omega t + C_2 \sin\omega t.$$

The initial conditions for $t=0$ are $\theta=0$ and $\dot{\theta}=\omega_0$. It results $C_1=0$ and $C_2=\dfrac{\omega}{\omega_0}$. The solution of the problem is

$$\theta=\frac{\omega_0}{\omega}\,sin\,\omega t.$$

Problem 3. Two external forces acts on a body with the mass m: a force proportional with time (the proportionality factor is equal to k_1) and a medium resistant force which is proportional with the velocity of the body (the proportionality factor being equal to k_2). The gravity is neglected.

Find and solve the equation of motion of the body.

Solution The differential equation of motion is $m\,\dfrac{dv}{dt}=k_1t-k_2v$. The following notation is used $k_1t-k_2v=u$. The derivative with respect to t gives $k_1-k_2\,\dfrac{dv}{dt}=\dfrac{du}{dt}$. Multiplying by m the following relation is obtained

$$k_1m-k_2m\,\frac{dv}{dt}=m\,\frac{du}{dt}\quad\text{or}\quad k_1m-k_2u=m\,\frac{du}{dt}.\qquad(176)$$

The previous relation is an equation with separable variables,

$$\frac{du}{k_1m-k_2u}=\frac{1}{m}dt.$$

After integration,

$$\int\frac{du}{k_1m-k_2u}=\frac{1}{m}\int dt+C\Rightarrow-\frac{1}{k_2}\,ln|k_1m-k_2u|=\frac{t}{m}+C.$$

From the initial condition $v(0)=0$ it results $u(0)=0$, hence

$$-\frac{1}{k_2}\,ln|k_1m|=C.$$

Replacing the value of C, yields

$$-\frac{1}{k_2}\,ln|k_1m-k_2u|=\frac{t}{m}-\frac{1}{k_2}\,ln|k_1m|.$$

Multiplying by $(-k_2)$

$$ln|k_1m-k_2u|=ln|k_1m|-\frac{k_2}{m}\,t$$

hence

$$k_1m-k_2u=k_1me^{-\frac{k_2}{m}t}\Rightarrow k_2u=k_1m-k_1me^{-\frac{k_2}{m}t}.$$

Replacing u by its expression depending on v the following relation is obtained

$$k_2k_1t-k_2^2v=k_1m-k_1m\,e^{-\frac{k_2}{m}t}\Rightarrow v(t)=\frac{k_1m}{k_2^2}\,e^{-\frac{k_2}{m}t}+\frac{k_1}{k_2}\,t-\frac{k_1m}{k_2^2}.$$

Next the dependence of the space in time is obtained using the equations

$$v(t)=\frac{ds(t)}{dt}\quad\text{or}\quad s(t)=\int v(t)dt+C,\quad\text{and}\quad s(0)=s_0.$$

Then yields,

$$s(t)=\int(\frac{k_1m}{k_2^2}\,e^{-\frac{k_2}{m}t}+\frac{k_1}{k_2}\,t-\frac{k_1m}{k_2^2})dt+C=-\frac{k_1m^2}{k_2^3}\,e^{-\frac{k_2}{m}t}+\frac{k_1}{2k_2}\,t^2-\frac{k_1m}{k_2^2}\,t+C.$$

The constant C is determined from the initial condition

$$s(0)=s_0\Rightarrow s_0=C-\frac{k_1m^2}{k_2^3}\quad\text{or}\quad C=s_0+\frac{k_1m^2}{k_2^3}.$$

The equation of the space is given by

$$s(t)=s_0+\frac{k_1m^2}{k_2^3}-\frac{k_1m}{k_2^2}\,t+\frac{k_1}{2k_2}\,t^2-\frac{k_1m^2}{k_2^3}\,e^{-\frac{k_2}{m}t}.$$

Problem 4. (The emptying of a reservoir) A reservoir has the shape of a rotational surface about a vertical axis with a hole at the bottom. The hole has the area A. Find and solve the equations of motion for the liquid located in the reservoir.

The following particular cases are considered for the reservoir:

a. spherical shape of radius R;
b. conical frustum with the smaller radius, R_1, as base radius, the larger radius, R_2, as top radius, and the height is H;
c. conical frustum with the larger radius, R_2, as base radius, the smaller radius, R_1, as top radius, and the height is H;
d. right cone with the vertex at the bottom;
e. cylindric shape.

Solution From hydrodynamics it is known the expression of the leakage velocity of a fluid through an orifice $v=k\sqrt{h}$, where h is the height of the free surface of the fluid.

The equation of the median radius of the reservoir is of the form $r=r(h)$. The volume of liquid that leaks during the elementary time dt is evaluated in the following way. Through the hole leaks the volume of liquid which fills a cylinder with base A and height $v\,dt$

$$dV=Av\,dt=Ak\sqrt{h}dt.$$

On the other side, the differential volume which leaks is $dV=-\pi\,r^2\,dh$. The following expression is obtained

$$A\,k\,\sqrt{h}\,dt=-\pi r^2dh.$$

It results a differential equation with separable variables

$$dt=-\frac{\pi}{Ak}\,\frac{r^2(h)}{\sqrt{h}}\,dh.$$

Solving the integral it is found

$$t=-\frac{\pi}{Ak}\int\frac{r^2(h)}{\sqrt{h}}\,dh+C.$$

From the initial condition $h(0)=H$ the constant C can be determined.

a. In the case of a spherical shape (Figure 12) the median radius can be written as $r^2=h(2R-h)$. Then,

or

$$t = -\frac{\pi}{Ak}[2R\int\sqrt{h}dh - \int h^{3/2}dh] + C = -\frac{\pi}{Ak}[\frac{4}{3}Rh^{3/2} - \frac{2}{5}h^{5/2}] + C.$$

Using the condition

$h(0) = H,$

yields

$$C = \frac{\pi}{Ak}[\frac{4}{3}RH^{3/2} - \frac{2}{5}H^{5/2}],$$

and hence

$$t = \frac{\pi}{Ak}[\frac{4}{3}R(H^{3/2} - h^{3/2}) - \frac{2}{5}(H^{5/2} - h^{5/2})].$$

The time T for which $h(T)=0$ is $T = \frac{\pi}{Ak}H^{3/2}(\frac{4}{3}R - \frac{2}{5}H)$.

For $H=R$ (the sphere is full) it results $T = (\frac{14}{15})\frac{\pi R^{5/2}}{Ak}$.

b. From the geometry of the conical frustum (Figure 13), $\frac{r-R_1}{h} = \frac{R_2-R_1}{H}$, and $r = R_1 + \frac{R_2-R_1}{H}h$. Then, $\frac{r^2}{\sqrt{h}} = \frac{R_1^2}{\sqrt{h}} + \frac{2R_1(R_2-R_1)}{H}\sqrt{h} + (\frac{R_2-R_1}{H})^2 h^{3/2}$ and substituting it in the expression of t, after the calculus of the integral, yields

$$t = -\frac{\pi}{Ak}[2R_1^2\sqrt{h} + (\frac{4}{3})\frac{R_1(R_2-R_1)}{H}h^{3/2} + (\frac{2}{5})\frac{R_2-R_1^2}{H}h^{5/2}] + C.$$

Using the condition

$h(0) = H,$

it is found that

$$C = \frac{\pi}{Ak}[2R_1^2\sqrt{H} + (\frac{4}{3})\frac{R_1(R_2-R_1)}{H}H^{3/2} + (\frac{2}{5})\frac{R_2-R_1^2}{H}H^{5/2}],$$

and hence,

$$t = \frac{\pi}{Ak}[2R_1^2(\sqrt{H}-\sqrt{h}) + (\frac{4}{3})\frac{R_1(R_2-R_1)}{H}(H^{3/2}-h^{3/2}) + (\frac{2}{5})\frac{R_2-R_1^2}{H}(H^{5/2}-h^{5/2})].$$

The condition $h(T)=0$ implies

$$T = \frac{\pi\sqrt{H}}{Ak}[2R_1^2 + \frac{4}{3}R_1(R_2-R_1) + \frac{2}{5}(R_2-R_1)^2].$$

c. From Figure 14, $\frac{r-R_1}{H-h} = \frac{R_2-R_1}{H}$ and yields, $r = R_2 + \frac{R_1-R_2}{H}h$.

If in the expression of r from case b), R_1 is replaced by R_2, one can find the expression of r from case c). Consequently, the expressions of t and T for the case c) will be obtained from the corresponding expressions obtained at b), in which R_1 will be replaced by R_2 and R_2 by R_1

$$t = \frac{\pi}{Ak}[2R_2^2(\sqrt{H}-\sqrt{h}) + (\frac{4}{3})\frac{R_2(R_1-R_2)}{H}(H^{3/2}-h^{3/2}) + \frac{5}{2}(\frac{R_1-R_2}{H})^2(H^{5/2}-h^{5/2})],$$

$$T = \frac{\pi\sqrt{H}}{Ak}[2R_2^2 + \frac{4}{3}R_2(R_1-R_2) + \frac{2}{5}(R_1-R_2)^2].$$

Comparing the expressions of T for the cases b) and c) and denoting by T' the expression in case c) it results

$$T' - T = \frac{\pi\sqrt{H}}{Ak}[2(R_2^2-R_1^2) + \frac{4}{3}R_2R_1 - \frac{4}{3}R_2^2 - \frac{4}{3}R_1R_2 + \frac{4}{3}R_1^2 + \frac{2}{5}(R_1-R_2)^2 - \frac{2}{5}(R_2-R_1)^2] = \frac{\pi\sqrt{H}}{Ak}\frac{2}{3}(R_2^2-R_1^2),$$

or,

$$T' = T + \frac{2}{3}\frac{\pi\sqrt{H}}{Ak}(R_2^2-R_1^2).$$

d. It is obtained from case b), taking $R_1=0$, $R_2=R$. Hence,

$$t = \frac{2\pi R^2}{5AkH^2}(H^{5/2}-h^{5/2}) \text{ and } T = \frac{2\pi R^2}{5Ak}\sqrt{H}.$$

e. It is obtained from case b), taking $R_1=R_2=R$. Then,

$$t = \frac{2\pi R^2}{Ak}(\sqrt{H}-\sqrt{h}) \text{ and } T = \frac{2\pi R^2}{Ak}\sqrt{H}.$$

Problem 5. Find the general solution of equation $\frac{y}{y'} = x + \sqrt{x^2+y^2}$.

Solution The equation can be written in the form $y\frac{dx}{dy} = x + \sqrt{x^2+y^2}$ or, $\frac{dx}{dy} = \frac{x}{y} + \sqrt{\frac{x^2}{y^2}+1}$..

Using the replacement $\frac{x}{y} = u$ or $x=yu$. It results $\frac{dx}{dy} = u + y\frac{du}{dy} \Rightarrow u + y\frac{du}{dy} = u + \sqrt{u^2+1} \Rightarrow \frac{du}{\sqrt{u^2+1}} = \frac{dy}{y} \Rightarrow \int\frac{du}{\sqrt{u^2+1}} = \int\frac{dy}{y} + \ln c \Rightarrow \ln(u+\sqrt{u^2+1}) = \ln y + \ln c \Rightarrow u + \sqrt{u^2+1} = cy \Rightarrow \frac{x}{y} + \sqrt{\frac{x^2}{y^2}+1} =$

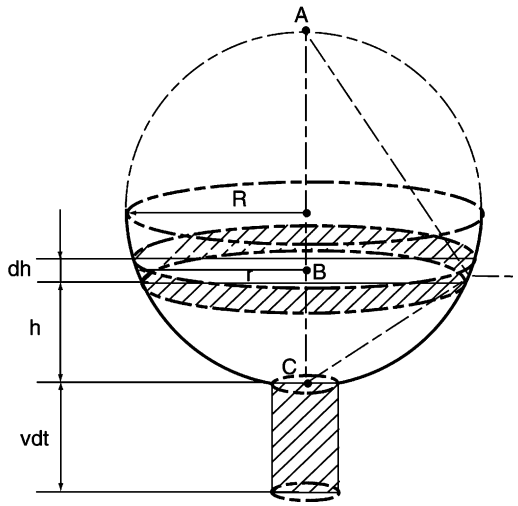


Figure 12. Spherical reservoir

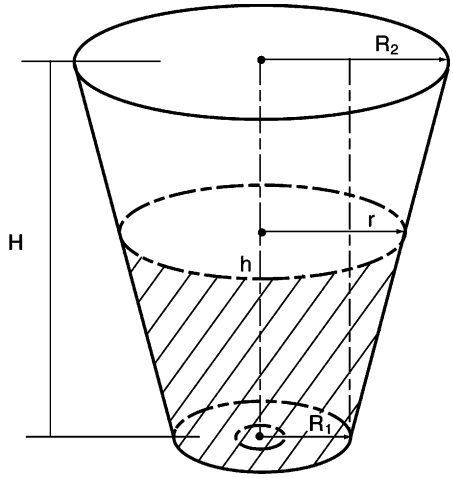


Figure 13. Conical frustum with smaller radius as base radius

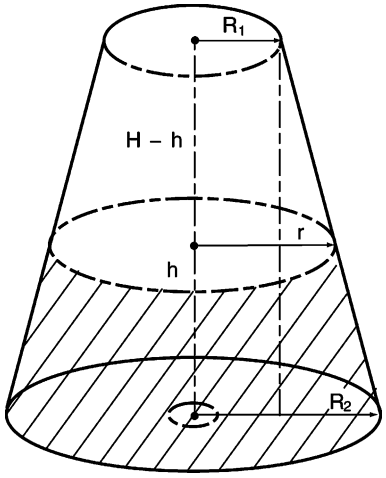


Figure 14. Conical frustum with larger radius as base radius

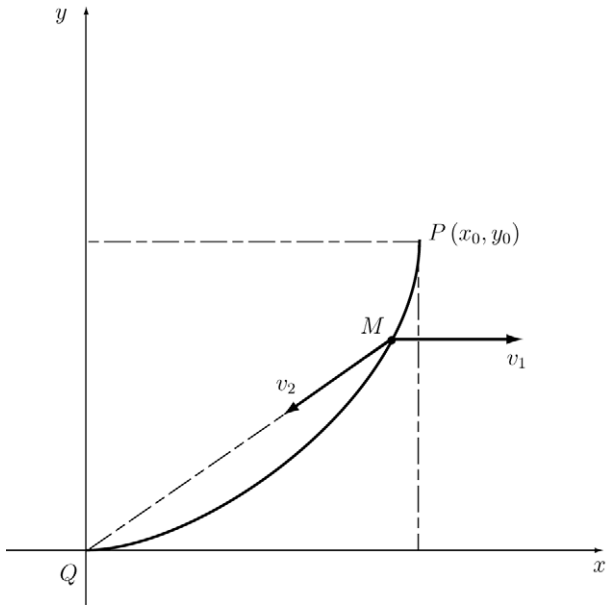


Figure 15. The path of the swimmer

$cy \Rightarrow \sqrt{x^2+y^2} = cy^2 - x \Rightarrow x^2 + y^2 = c^2y^4 - 2cxy^2 + x^2 \Rightarrow c^2y^2 = 2cx + 1$ is the general solution.

Problem 6. (The problem of the swimmer) To cross a river, a swimmer starts from a point P on the bank. He wants to arrive at the point Q on the other side. The velocity of the river is constant and equal to $v_1 = k_1$ and the velocity of swimmer the is $v_2 = k_2$ where k_2 is constant. Find the trajectory described by the swimmer, knowing that the velocity of the swimmer is always directed toward Q.

Solution Select Q as the origin of the system as shown in Figure 15. Consider that M is the swimmer position at time t. The components of the absolute velocity on the two axes Ox and Oy are

$$\begin{aligned} \frac{dx}{dt} &= k_1 - k_2 \frac{x}{\sqrt{x^2+y^2}}, \\ \frac{dy}{dt} &= -k_2 \frac{y}{\sqrt{x^2+y^2}}. \end{aligned}$$

Dividing the previous relation it results

$$\frac{dx}{dy} = \frac{x}{y} - k\sqrt{\frac{x^2}{y^2}+1},$$

where $k = \dfrac{k_1}{k_2}$.

The following notation is used $x = yu$ and $\dfrac{dx}{dy} = u + y\dfrac{du}{dy}$. The differential equation becomes

$$y\frac{du}{dy} = -k\sqrt{u^2+1} \quad \text{or} \quad \frac{du}{\sqrt{u^2+1}} = -k\frac{dy}{y}.$$

After integration results

$$ln(u + \sqrt{u^2+1}) = -k\, ln\, y + ln\, c(c>0) \quad \text{or} \quad u + \sqrt{u^2+1} = cy^{-k}.$$

Then, yields

$$u = \frac{1}{2}\left(\frac{c}{y^k} - \frac{y^k}{c}\right).$$

Returning at x and y, $x = \frac{1}{2}\, y\left(\frac{c}{y^k} - \frac{y^k}{c}\right)$. From the conditon for trajectory to pass through the initial point $P(x_0, y_0)$ the constant c is $c = y_0^{k-1}(x_0 + \sqrt{x_0^2 + y_0^2})$.

The condition for trajectory to pass through Q is written as $\lim_{y \to 0} \frac{1}{2}\, y\left(\frac{c}{y^k} - \frac{y^k}{c}\right) = 0$ and it is possible if $k < 1$.

For $k_1 = 0$, $k = 0$ and the trajectory has the equation $x = \frac{x_0}{y_0}\, y$, i.e., the linear segment between P and Q.

Problem 7. Determine the minimum velocity of a body thrown vertically upwards so that the body will not return to the Earth. The air resistance is neglected.

Solution Denote the mass of the Earth by M and the mass of the body by m. Using Newton's law of gravitation, the force of attraction f acting on the body m is $f = k\,\frac{Mm}{r^2}$, where r is the distance between the center of the Earth and the center of gravity of the body and k is the gravitational constant. The differential equation of the motion for the body is

$$m\,\frac{d^2r}{dt^2} = -k\,\frac{Mm}{r^2} \quad \text{or} \quad \frac{d^2r}{dt^2} = -k\,\frac{M}{r^2}. \tag{177}$$

The minus sign indicates a negative acceleration. The differential Eq. (415) will be solved for the following initial conditions

$$r(0) = R \quad \text{and} \quad \frac{dr(0)}{dt} = v_0. \tag{178}$$

Here, R is the radius of Earth and v_0 is the launching velocity. The following notations are used $\frac{dr}{dt} = v \Rightarrow \frac{d^2r}{dt^2} = \frac{dv}{dt} = \frac{dv}{dr}\left(\frac{dr}{dt}\right) = v\,\frac{dv}{dr}$, where v is the velocity of motion. Substituting in Eq. (415), results $v\,\frac{dv}{dr} = -k\,\frac{M}{r^2}$. Separating variables, it is found $v\,dv = -kM\,\frac{dr}{r^2}$. Integrating this equation, yields $\frac{v^2}{2} = kM\,\frac{1}{r} + c_1$. From conditions (416), c_1 is found

$$\frac{v_0^2}{2} = kM\left(\frac{1}{R}\right) + c_1,$$

or,

$$c_1 = -\frac{kM}{R} + \frac{v_0^2}{2},$$

and

$$\frac{v^2}{2} = kM\,\frac{1}{r} + \left(\frac{v_0^2}{2} - \frac{kM}{R}\right). \tag{179}$$

The body should move so that the velocity is always positive, hence $\frac{v^2}{2} > 0$. Since for a boundless increase of r the quantity $\frac{kM}{R}$ becomes arbitrarily small, the condition $\frac{v^2}{2} > 0$ will be fulfilled for any r only for the case

$$\frac{v_0^2}{2} - \frac{kM}{R} \geq 0 \quad \text{or} \quad v_0 \geq \sqrt{\frac{2kM}{k}}.$$

Hence, the minimal velocity is determined by the equation

$$v_0 = \sqrt{\frac{2kM}{R}}, \tag{180}$$

where $k = 6.66(10^{-8})$ cm^3/(g s^2), $R = 63(10^7)$ cm. At the Earth's surface, for $r = R$, the acceleration of gravity is $g = 981$ cm/s^2. For this reason, from Eq. (415) yields $g = k\,\frac{M}{R^2}$ or $M = \frac{gR^2}{k}$. Substituting this value of M into Eq. 421 it results

$$v_0 = \sqrt{2gR} = \sqrt{2(981)(63)(10^7)} \approx 11.2(10^5)\,\text{cm/s} = 11.2\,\text{km/s}.$$

BIBLIOGRAPHY

1. Kreyszig, E., *Advanced Engineering Mathematics*, John Wiley and Sons Inc., 1972.
2. Braun, M., *Differential Equations and Their Applications*, Spinger-Verlag, 1983.
3. Ince, E. L., *Ordinary Differential Equations*, Dover, New York, 1956.
4. Ayres, F., *Matrices*, Schaum, New York, 1962.
5. Jordan, D. W., and Smith, P., *Nonlinear ordinary differential equations*, Clarendon Press, Oxford, 1977.

Reading List

A. Fletcher, A., Miller, J. C., Rosenhead, L., and Comrie, L. J., *An Index of Mathematical Tables*, Blackwell, Oxford, 1962.

B. Courant, R. and Hilbert, D., *Mathematical Physics Vol. II*, Interscience Publishers, John Wiley & Sons, New York, 1989.

C. Weinberger, H. F., *Differential Equations*, Xerox College Publishing, Lexington, 1965.

DAN B. MARGHITU
S. C. SINHA
Auburn University, Auburn, AL

Polynomials
Standard Article
Peter F. Stiller[1]
[1]Texas A & M University, College Station, TX
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2443
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (280K)

Abstract

The sections in this article are

Overview of Resultants

Resultants of Polynomials in One Variable

Resultant Methods for Systems of Polynomial Equations in Several Variables

Invariant Polynomials

| | |

Copyright © 1999-2008 All Rights Reserved.

POLYNOMIALS

Polynomials of one or more variables are likely to be familiar to most readers. Expressions such as

$$3t^2 - 7t + 2 \quad \text{or} \quad x^2 + y^2 - 1$$

are easily recalled from high school mathematics. In general, polynomials involve some number n of variables, call them $x_1, \ldots, x_n$, and a set of allowable coefficients usually taken to lie in particular field or ring. Common fields are the field of rational numbers $\mathbb{Q}$, the field of real numbers $\mathbb{R}$, or the field of complex numbers $\mathbb{C}$. The ring of ordinary integers $\mathbb{Z}$ provides an example of a coefficient ring that is not a field.

A monomial in the variables $x_1, \ldots, x_n$ is a power product of the form

$$x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$$

where the exponents $\alpha_1, \ldots, \alpha_n$ are nonnegative integers. The total degree of the monomial is the sum $\alpha_1, + \cdots + \alpha_n$.

Because of the number of variables involved, a shorthand notation is needed. We let $\alpha = (\alpha_1, \ldots, \alpha_n)$ be an n-tuple of nonnegative integers, and we define

$$x^\alpha = x_1^{\alpha_1} \ldots x_n^{\alpha_n}$$

where x represents $(x_1, \ldots, x_n)$. The total degree is the denoted by $|\alpha| = \alpha_1 + \cdots + \alpha_n$.

A polynomial $f(x_1, \ldots, x_n)$ in the variables $x_1, \ldots, x_n$ with coefficients in a field K (or ring R) is a finite sum of terms of the form

$$f(x_1, \ldots, x_n) = \sum_\alpha a_\alpha x^\alpha = \sum a_{\alpha_1, \ldots, \alpha_n} x_1^{\alpha_1} \ldots x_n^{\alpha_n}$$

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

where $a_\alpha \in K$ (or R). The set of all such polynomials is written $K[x_1, \ldots, x_n]$. We call a_α the coefficient of the monomial x^α and call $a_\alpha x^\alpha$ a term in the polynomial when $a_\alpha \neq 0$.

The total degree (or just degree) of $f(x_1, \ldots, x_n)$, denoted deg f, is the maximum of the degrees $|\alpha|$ of the monomials that occur in the terms of f, that is, the maximum over the $|\alpha| = \alpha_1 + \cdots + \alpha_n$ such that a_α is not zero.

A polynomial is said to be homogeneous of degree d if every monomial occurring in a term of f has degree equal d. Thus $y^3 + x^2y + zw^2$ is homogeneous of degree 3 in four variables, whereas $x^3y + 3xwz$ is of degree 4 in four variables but is not homogeneous.

One central problem that frequently arises is the need to solve a system of m polynomial equations in n variables:

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, \ldots, x_n) &= 0 \end{aligned}$$

where the f_i are in $K[x_1, \ldots, x_n]$. Solutions are sought in K^n or in some larger field E^n where $K \subset E$. (The example to keep in mind is finding complex solutions to equations with real coefficients.) K^n in this case is just the set of n-tuples of elements of K, which we call n-space:

$$K^n = \{(a_1, \ldots, a_n) \text{ with } a_i \in K\}$$

We say that $(a_1, \ldots, a_n) \in K^n$ is a solution to the system above if $f_i(a_1, \ldots, a_n) = 0$ for all $i = 1, \ldots, n$.

Naively, we expect that a system of n equations in n variables will have a finite number of solutions. This, however, need not be the case. Consider three equations in three variables (coefficients in $\mathbb{R}$ say):

$$\begin{aligned} f_1(x, y, z) &= 0 \\ f_2(x, y, z) &= 0 \\ f_3(x, y, z) &= 0 \end{aligned}$$

Each represents a surface in three-space. If those surfaces should all contain a common curve, then the set of solutions to the system would be infinite.

For example, the system

$$\begin{aligned} x^2 + y^2 + z^2 - 1 &= 0 \\ x^2 + y^2 + \frac{z^2}{4} - 1 &= 0 \\ x^2 + y^2 - 1 &= 0 \end{aligned}$$

has as solutions the unit circle in the (x, y) plane, that is, all points $(a, b, 0)$ where $a^2 + b^2 = 1$.

Numerical methods to solve systems of polynomial equations (when those systems have isolated point solutions) are known and discussed elsewhere. Here we take up some perhaps less well known techniques for dealing with and understanding systems of polynomial equations. Later we discuss an important use of what are called invariant polynomials in image understanding applications.

OVERVIEW OF RESULTANTS

Resultants are used to solve systems of polynomial equations, to determine whether or not solutions exist, or to reduce a given system to one with fewer variables and/or fewer equations.

Input

The typical input will be a system of m equations in n variables:

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, \ldots, x_n) &= 0 \end{aligned} \tag{1}$$

Each equation has an associated degree $d_i \geq 1$. Recall that $f_i(x_1, \ldots, x_n)$ has degree d_i if all monomials $x_1^{e_1} x_2^{e_2} \cdots x_n^{e_n}$ appearing in f_i have $\sum_{i=1}^n e_i \leq d_i$ and at least one monomial has $\sum_{i=1}^n e_i = d_i$. As an example, $f(x_1, x_2, x_3) = 3x_1^2x_3 + 4x_1x_2 - x_2 + 7x_3 - 1$ has degree $d = 3$. The integers $m, n, d_1, \ldots, d_m$ are important indicators of the specific techniques that will need to be employed.

Output

There are two essentially different cases.

Case 1: $m > n$ (overdetermined). This is the case where we have more equations than unknowns and where we generally expect to have no solutions. The resultant will be a system of equations (one equation when $m = n + 1$) in the symbolic coefficients of the f_i that has the following property: when we substitute the specific numerical coefficients of the f_i, we will get zero in every equation in the resultant system if and only if the original overdetermined system has a solution.

Case 2: $m \leq n$ (exact and underdetermined). In this case the number of equations is less than or equal to the number of variables, and we expect to have solutions. In fact, if we allow complex solutions and solutions at infinity, we are guaranteed to have solutions.

Of course, only when $m = n$ do we expect a finite number s of solutions. Bezout's theorem then provides a count of $s = d_1d_2 \cdots d_m$ solutions (counting complex solutions, solutions at infinity, and counting with appropriate multiplicities). Unfortunately, as mentioned, the possibility also exists (even when $m = n$) that there will be an infinite number of solutions.

In general, for $m \leq n$, the resultant will be one equation in $n - m + 1$ of the variables. In effect, the resultant eliminates $m - 1$ of the variables. For example, if we choose to eliminate $x_{n-m+2}, \ldots, x_n$, then the resultant R will be a polynomial $R(x_1, \ldots, x_{n-m+1})$ in the remaining variables. If $(a_1, \ldots, a_{n-m+1})$ is a solution to $R = 0$, then there will exist values $a_{n-m+2}, \ldots, a_n$ such that $(a_1, a_{n-m+1}, a_{n-m+2}, \ldots, a_n)$ is a solution to the original system. [One must be a little careful here. The system should be modified to make it homogeneous with respect to $x_{n-m+2}, \ldots, x_n$ by adding appropriate powers of a variable w. The values $a_{n-m+2}, \ldots, a_n$ should be regarded as the coordinates of a point $(a_{n-m+2} : \cdots : a_n : 1)$ in projective $(m-1)$-space $\mathbb{P}^{m-1}$. We must allow for the possibility that this point will be at infinity where $w = 0$. In that case, a solution to $R = 0$ would not necessarily give rise to a solution of the original system.]

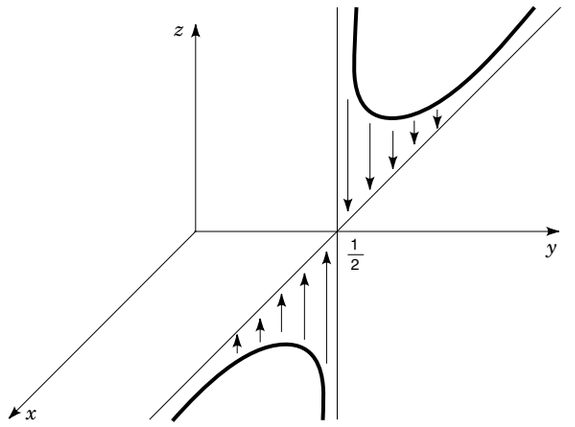


Figure 1. Resultant as projection.

For example, looking at Fig. 1, consider the system of $m = 2$ equations in $n = 3$ variables: $4xyz - 1 = 0$ and $y + xz - 1 = 0$. The resultant eliminating z is $R(x, y) = x(4y^2 - 4y + 1)$. When $x = 0$ we will have $R = 0$, but clearly our system has no solution when $x = 0$. However, homogenizing with respect to z gives the system

$$4xyz - w = 0 \quad \text{and} \quad (y-1)w + xz = 0$$

Now when we look at the condition $x = 0$, we find that $(z : w) = (1 : 0)$ is a solution. This is a point at infinity.

Notice that we also have solutions to $R = 0$ when $x \neq 0$ by taking $y = \frac{1}{2}$. This yields $z = 1/2x$. Geometrically the solution set is a hyperbola in the plane $y = \frac{1}{2}$ in space. The resultant "projects" that hyperbola to the line $y = \frac{1}{2}$ in the (x, y) plane, except that $(x, y) = (0, \frac{1}{2})$ is not hit.

In this context (the underdetermined case) the resultant can be viewed as a projection of the nominally $(n - m)$-dimensional locus of solutions in $\mathbb{R}^n$ to an $(n - m)$-dimensional locus (hypersurface) in $\mathbb{R}^{n-m+1}$. Note that in our example $n = 3$, $m = 2$, and we are projecting the one-dimensional locus of solutions in $\mathbb{R}^3$ to a one-dimensional locus in $\mathbb{R}^2$ which is described by one equation $y - \frac{1}{2} = 0$.

Approach in This Article

We begin with the first major distinction in methods, namely the one based on the number of variables n. The case $n = 1$ of a single variable is discussed first. We then move on to the multivariate case $n \geq 2$. See Table 1 (1).

RESULTANTS OF POLYNOMIALS IN ONE VARIABLE

The Basic Case: Two Polynomials and the Sylvester Matrix

Given two positive integers $r, s \geq 1$ and two polynomials in one variable

$$f(x) = a_r x^r + \cdots + a_1 x + a_0 \quad \text{and} \quad g(x) = b_s x^s + \cdots + b_1 x + b_0$$

of degree less than or equal to r and s, respectively, we define their resultant $R_{r,s}(f, g)$ by Sylvester's formula:

$$R_{r,s}(f,g) = \det \begin{pmatrix} a_0 & a_1 & & \cdots & a_r & 0 & \cdots & & 0 \\ 0 & a_0 & a_1 & \cdots & \cdots & a_r & 0 & \cdots & 0 \\ \vdots & & & & & & & & \\ 0 & 0 & \cdots & 0 & a_0 & a_1 & & \cdots & a_r \\ b_0 & b_1 & \cdots & & b_{s-1} & b_s & 0 & \cdots & 0 \\ 0 & b_0 & b_1 & & \cdots & & b_s & \cdots & 0 \\ \vdots & & & & & & & & \\ 0 & 0 & \cdots & & 0 & b_0 & b_1 & \cdots & b_s \end{pmatrix}$$

which is the determinant of an $r + s$ by $r + s$ matrix with s rows involving the a's and r rows involving the b's.

Example 1

$$\begin{aligned} R_{2,2}(a_2x^2 + a_1x + a_0, b_2x^2 + b_1x + b_0) &= a_0^2b_2^2 + a_0a_2b_1^2 - a_0a_1b_1b_2 + a_1^2b_0b_2 \\ &\quad - a_1a_2b_0b_1 + a_2^2b_0^2 - 2a_0a_2b_0b_2 \end{aligned}$$

Note that in this example each monomial in the resultant has total degree $r + s = 4$ and is bihomogeneous of bidegree $(s, r) = (2, 2)$ in the a's and b's respectively. This is true in general.

Basic Properties of the Resultant $R_{r,s}(f, g)$

1. *Relationship to common roots.*
$$R_{r,s}(f,g) = a_r^s b_s^r \prod_{i,j} (x_i - y_j)$$
where $x_1, \ldots, x_r$ are the roots of f and $y_1, \ldots, y_s$ are the roots of g. (Here we are assuming $a_r \neq 0$ and $b_s \neq 0$.) Thus $R_{r,s}(f, g)$ will be zero if and only if f and g have a root in common.
2. *Irreducibility.* $R_{r,s}(f, g) \in \mathbb{Z}[a_0, \ldots, a_r, b_0, \ldots, b_s]$ is irreducible, that is, the resultant is an irreducible polynomial with integer ($\mathbb{Z}$) coefficients in $(r + 1)(s + 1) = rs + s + r + 1$ variables.
3. *Symmetry.* $R_{r,s}(f, g) = (-1)^{rs} R_{r,s}(g, f)$
4. *Factorization.* $R_{r_1+r_2,s}(f_1 f_2, g) = R_{r_1,s}(f_1, g) R_{r_2,s}(f_2, g)$.

Discriminants and Resultants

The discriminant $\Delta(f)$ of a polynomial $f = a_r x^r + \cdots + a_0$, $a_r \neq 0$, is essentially the resultant of f and its derivative f'. The exact relationship is

$$\Delta(f) = \frac{1}{a_r} R_{r,r-1}(f, f')$$

which is a homogeneous polynomial of degree $2r - 2$ in the $r + 1$ variables $a_0, \ldots, a_r$. It provides a test for multiple roots.

Just as the discriminant can be defined in terms of the resultant, the resultant can be defined in terms of the discriminant:

$$[R_{r,s}(f,g)]^2 = (-1)^{rs} \frac{\Delta(fg)}{\Delta(f)\Delta(g)}$$

when $a_r \neq 0$ and $b_s \neq 0$.

Table 1. Table of Resultants

n	m	Type of Resultant to Use	Notes
1	2	Determinant of the Sylvester matrix	This is what is most commonly thought of as *the* resultant.
1	≥ 3	Requires a system of equations	See the discussion in van der Waerden (1).
≥ 2	$m = n + 1$	Macaulay resultant	This is computed as the quotient of two determinants. It is a polynomial in the symbolic coefficients and is zero if and only if the system has a solution.
≥ 2	$m \geq n + 2$	Requires a system of equations	See van der Waerden (1).
≥ 2	$m = n$	U resultant or generalized characteristic polynomial	This resultant is designed to find the finite set of all solutions to the system of equations.
≥ 2	$m < n$	Macaulay resultant using $m - 1$ variables, while treating the other $n - m + 1$ variables as included in the coefficients	The result is a single polynomial in the remaining $n - m + 1$ variables.

Note: One can also employ the standard Sylvester resultant in the multivariate case, using it iteratively to successively eliminate variables. For example, with three equations in three unknowns $f(x, y, z) = 0$, $g(x, y, z) = 0$, and $h(x, y, z) = 0$, we can take the resultant of f and g treating z as the only variable to get $R_1(x, y)$. Likewise we can take the resultant of g and h again treating z as the only variable to get $R_2(x, y)$. Finally, the resultant of R_1 and R_2 with y as the variable yields $R(x)$, whose roots can then be found using standard root-finding methods.

Finding the Common Roots: Subresultants

Again, suppose we are given two polynomials in a single variable x, say

$$f(x) = a_r x^r + \cdots + a_1 + a_0 \quad \text{and} \quad g(x) = b_s x^s + \cdots + b_1 x + b_0$$

of degrees $r \geq 1$ and $s \geq 1$, respectively. (We assume that $a_r \neq 0$ and $b_s \neq 0$.) As we saw previously, the resultant $R_{r,s}(f, g)$ of f and g will be zero if and only if f and g have a common root. Two questions immediately occur.

Question 1. Suppose $R_{r,s}(f, g) = 0$, so that f and g have at least one common root. Can we determine how many roots they have in common? This is the same as asking for the degree $1 \leq d \leq \min(r, s)$ of the greatest common divisor $h(x)$ of $f(x)$ and $g(x)$.

Question 2. Can we find the common roots?

The answer to Question 2 is more subtle. In general, we cannot expect to be able to express the common roots of f and g (assuming they have a root or roots in common) as rational expressions in the coefficients $a_r, \ldots, a_0, b_s, \ldots, b_0$. For example, if f and g have rational coefficients, that is, $a_r, \ldots, a_0, b_s, \ldots, b_0 \in \mathbb{Q}$, the field of rational numbers, then any polynomial expression in the coefficients would be a rational number. But polynomials with rational coefficients can have common roots that are not rational.

Example 2. $f(x) = 3x^4 + x^3 + 4x^2 + x + 1 = (x^2 + 1)(3x^2 + x + 1)$ and $g(x) = x^2 - 1 = (x^2 + 1)(x^2 - 1)$ both have rational coefficients, but the common roots $\pm i$ are not rational numbers.

We can however answer Question 2 in a special case. If $R_{r,s}(f, g) = 0$ and at least one partial derivative of the resultant computed symbolically

$$\frac{\partial R}{\partial a_0}, \ldots, \frac{\partial R}{\partial a_r}, \frac{\partial R}{\partial b_0}, \ldots, \frac{\partial R}{\partial b_s} \tag{2}$$

is nonzero when the coefficients of f and g are substituted, then f and g have exactly one common root α and it can be found via the proportions:

$$(1 : \alpha : \alpha^2 : \cdots : \alpha^r) = \left(\frac{\partial R}{\partial a_0}(f,g) : \frac{\partial R}{\partial a_1}(f,g) : \cdots : \frac{\partial R}{\partial a_r}(f,g) \right)$$
$$(1 : \alpha : \alpha^2 : \cdots : \alpha^s) = \left(\frac{\partial R}{\partial b_0}(f,g) : \frac{\partial R}{\partial b_1}(f,g) : \cdots : \frac{\partial R}{\partial b_s}(f,g) \right)$$

In particular the common root α can be computed as

$$\alpha = \frac{\dfrac{\partial R}{\partial a_1}(f,g)}{\dfrac{\partial R}{\partial a_0}(f,g)} = \frac{\dfrac{\partial R}{\partial b_1}(f,g)}{\dfrac{\partial R}{\partial b_0}(f,g)}$$

This result also has a geometric interpretation. The space of all pairs of polynomials (f, g) where the degree of f is less than or equal to r and the degree of g is less than or equal to s can be identified with $\mathbb{R}^{r+s+2}$ having coordinates $(a_r, \ldots, a_0, b_s, \ldots, b_0)$. The symbolic resultant R is a polynomial in these variables, and the locus $R = 0$ in $\mathbb{R}^{r+s+2}$ is an irreducible hypersurface (of dimension $r + s + 1$) consisting of pairs (f, g) with a root in common. A point on this hypersurface where at least one of the partial derivatives in Eq. (2) is nonzero is a smooth point. At such points we have exactly one common root. Moreover, that root can be expressed as a quotient of polynomial expressions in $a_r, \ldots, a_0, b_s, \ldots, b_0$. We remind the reader that "most" points on the locus $R = 0$ are smooth points. Those that are not are called singular points and they occur in dimension $r + s$ or less.

RESULTANT METHODS FOR SYSTEMS OF POLYNOMIAL EQUATIONS IN SEVERAL VARIABLES

Theory

The linear algebra techniques discussed next can be used to solve systems of polynomial equations in several variables. If there are only two equations, then the Sylvester technique (discussed earlier) can be employed by treating all but one variable as part of the coefficients. However, when the number of equations exceeds two, the Sylvester approach can be misleading. For example, taking the equations two at a time using the Sylvester determinant can lead the user to the conclusion that there is a common solution, when in fact there are no common solutions for the system of equations taken as a whole.

What it means to "solve" a given set of polynomial equations depends upon the number of variables and the number of equations. Assuming the equations are inhomogeneous, let n be the number of variables and m be the number of equations. The expected dimensionality of the set of solutions is $n - m$ when viewed over the complex numbers. For example, if there are three equations ($m = 3$) and five variables ($n = 5$), then the space of solutions is expected to have dimension $n - m = 5 - 3 = 2$. Geometrically, the set of solutions forms a surface. Sometimes, however, components of excess dimension occur in the set of solutions. These are geometric loci of higher dimension than the expected dimension. They occur because, in a very loose sense, the equations have certain dependencies.

Finally, a note is given about homogeneous equations. Recall that a set of polynomial equations is considered homogeneous if in each equation all the terms have the same degree. If this is not the case, even for only one of the equations, the set is regarded as inhomogeneous. For systems of homogeneous equations the number n of variables should be taken as one less than the actual number of variables when computing expected dimensions. This is because we want to regard the solutions as lying in an $(n - 1)$-dimensional projective space.

The Macaulay Resultant, the *U* Resultant, and the Generalized Characteristic Polynomial

The Macaulay resultant is the ratio of two determinants formed from the coefficients of the given polynomials in a manner to be described later in this section. If the number of equations exceeds the number of variables by one ($n - m = -1$), then the Macaulay resultant tests whether or not a common solution exists. [For systems of homogeneous equations in which the number of equations equals the number of variables, the expected dimension is still -1, and the Macaulay resultant tests for a nontrivial common solution, that is, a solution other than $(0, \ldots, 0)$]

If there are as many inhomogeneous equations as unknowns ($n - m = 0$), then the equations can often be solved by adding the U equation (explained later in this section) to the homogenized set and forming the Macaulay resultant. The Macaulay resultant is then called the U resultant.

In some cases, however, there will be a component of excess dimension (≥ 1) which masks some or all of the desired solutions. In this case Canny's generalized characteristic polynomial (GCP) approach is useful (see Ref. 2).

In order to illustrate the various methods, the following system of three polynomial equations will be used:

$$f_1 = y - 3x + 5 = 0$$
$$f_2 = x^2 + y^2 - 5 = 0$$
$$f_3 = y - x^3 + 3x^2 - 3x + 1 = 0$$

Here we have three inhomogeneous equations in two variables ($n - m = 2 - 3 = -1$). The multiresultant techniques described below can be used to test for the existence of a solution.

Step 1: Homogenization

The equations must first be homogenized. This is done by adding a third variable z. Specifically x is replaced by x/z and y is replaced by y/z, and the factors of z are cleared from the denominators. In the previous example this leads to three equations:

$$f_1 = y - 3x + 5z = 0$$
$$f_2 = x^2 + y^2 - 5z^2 = 0$$
$$f_3 = yz^2 - x^3 + 3x^2z - 3xz^2 + z^3 = 0$$

This is the homogenized version of the original system.

Step 2: Degree Determination

Each of the multiresultants being considered involves the coefficients of various monomials that appear in the equations. The variables involved in the monomials are the variables that appear in the homogeneous form of the polynomial equations. For example, the homogeneous polynomial equations above have the variables x, y, and z. All the monomials in a given equation are constrained to have the same degree because we have homogenized. The "overall degree" of the system is determined from the degrees of the individual homogeneous equations by the following rule:

$$d = 1 + \sum_{i=1}^{m} (d_i - 1)$$

where m is the number of equations and d_i the degree of the ith equation.

For the homogeneous polynomials given previously (f_1, f_2, and f_3) the degrees are

Equation	Degree
f_1	$d_1 = 1$
f_2	$d_2 = 2$
f_3	$d_3 = 3$

Therefore,

$$d = 1 + (1 - 1) + (2 - 1) + (3 - 1) = 4$$

Step 3: Matrix Size Determination

Each of the multiresultants to be discussed involves the ratio of two determinants. The numerator is the determinant of a matrix, the formation of which will be discussed in subsequent sections. The denominator determinant is formed from a submatrix of the numerator matrix.

The number of variables in the inhomogeneous equations is n. Since one additional variable has to be added to homogenize the equations, the number of variables in the homogeneous equations is $n + 1$. The size of the numerator matrix equals the number of monomials in the $n + 1$ variables that have overall degree d (discussed in the previous step).

$$\text{Numerator matrix size} = \binom{n+d}{d}$$

For the three polynomial equations (f_1, f_2, f_3) we have already calculated that $d = 4$. Since the original set of inhomogeneous variables consisted of x and y, we have that n equals 2. Thus for our example,

$$\text{Numerator matrix size} = \binom{2+4}{4} = \binom{6}{4} = \frac{6!}{(2!)(4!)} = 15$$

that is, it is a 15×15 matrix.

Step 4: Determining "Big" versus "Small" Exponents

A few of the 15 monomials involving the variables x, y, and z with an overall degree of 4 include:

$$yz^3 \quad \text{and} \quad x^2y^2$$

In the next section we will discuss whether certain of these monomials are reduced. This will be determined by whether the exponents are "big" or "small." In this section we discuss how bigness is defined.

Each variable will be associated with a particular equation. For example, the first variable, x, will be associated with the first equation, f_1. The second variable, y, will be associated with the second equation, f_2, etc. The degrees of the associated equations define bigness for the exponents of that variable. Specifically, since d_1 (the degree of f_1) is 1, if the exponent of x is greater than or equal to 1, it is considered big. Since $d_2 = 2$, whenever the exponent of y is greater than or equal to 2, it is considered big. The degree of f_3 is 3, therefore, whenever the exponent of z is greater than or equal to 3, it is considered big.

For example, consider the monomial yz^3. The exponent of y is 1. This is *less than* d_2, and is considered small. The exponent of z is 3. This is *equal to* d_3, and is therefore big. On the other hand, consider the monomial x^2y^2. The exponent of x is 2. This is *greater than* d_1 and is *big*. The exponent of y is 2. This is *equal to* d_2 and is *big*.

Step 5: Determining the Reduced Monomials

If for a particular monomial of degree d the exponent of *only one* variable is big, the monomial is said to be reduced. In the previous step the monomial yz^3 is reduced. For that monomial only the exponent of z is big, whereas for x^2y^2, both the exponent of x and the exponent of y are big. Thus the monomial x^2y^2 is not reduced.

Step 6: Creating the *A* Matrix

The Macaulay resultant is the ratio of two determinants. The numerator is the determinant of a matrix that we will call the A matrix. The denominator is the determinant of a matrix that we will call the M matrix

$$R = \frac{\det|A|}{\det|M|}$$

We have discussed above how the size of the A matrix is determined. In this section we will show how the matrix entries are obtained.

Each row and column of the matrix should be thought of as being labeled by one of the monomials of degree d. This labeling can be done in any desired order. Recall that for f_1, f_2, and f_3 in our example there were 15 possible monomials of degree 4 in x, y, z, and therefore the A matrix would be 15×15.

There are three rules for determining the elements of the A matrix. After presenting the rules, the example involving f_1, f_2, and f_3, will be used to illustrate the process. The reader may find it helpful to read the example simultaneously with the rules.

Rules for inputting the elements of each column of the A matrix:

1. Search the monomial labeling that column from left to right for the *first* variable with a big exponent. Such a variable must exist. Call it the marker variable.
2. Form a new polynomial from the polynomial associated with this marker variable by multiplying the associated polynomial by the monomial and dividing by the marker variable raised to the degree of the associated polynomial.
3. The coefficients of the new polynomial are the elements of the columns. Each coefficient goes in the row labeled by the monomial it multiples. All the other rows get zeros.

Example 3. Recall that for the system of equations f_1, f_2, f_3 there are 15 monomials of degree 4 that can be formed from x, y, and z. Two of these were considered above, namely yz^3 and x^2y^2.

- For the column labeled by yz^3:
  1. The first variable with a big exponent is z, so z is the marker variable.
  2. The polynomial associated with z is f_3. Multiply f_3 by the monomial yz^3, and divide this product by z^3.

$$\frac{f_3(yz^3)}{z^3} = \frac{(yz^2 - x^3 + 3x^2z - 3xz^2 + z^3)(yz^3)}{z^3}$$
$$= y^2z^2 - x^3y + 3x^2yz - 3xyz^2 + yz^3$$

  3. The coefficient of y^2z^2 is $+1$. Therefore the element of the row labeled y^2z^2 is $+1$. The coefficient of x^3y is -1. Therefore the element of the row labeled x^3y is -1. The coefficient of x^2yz is $+3$. Therefore the element of the row labeled x^2yz is $+3$. The coefficient of xyz^2 is -3. Therefore the element of the row labeled xyz^2 is -3. The coefficient of yz^3 is $+1$. Therefore the element of the row labeled yz^3 is $+1$. All other entries in the column are zero.
- For the column labeled by x^2y^2:
  1. The first variable with a big exponent is x, so x is the marker variable.
  2. The polynomial associated with x is f_1. Multiply f_1 by the monomial x^2y^2, and divide this product by x:

$$\frac{f_1(x^2y^2)}{x} = \frac{(y - 3x + 5z)(x^2y^2)}{x} = xy^3 - 3x^2y^2 + 5xy^2z$$

  3. The coefficient of xy^3 is $+1$. Therefore the element of the row labeled xy^3 is $+1$. The coefficient of x^2y^2 is -3. Therefore the element of the row labeled x^2y^2 is -3. The coefficient of xy^2z is $+5$. Therefore the element of the row labeled xy^2z is $+5$.

When all the columns are determined, the A matrix in our example takes the form:

			x^4	$x^3 y$	$x^3 z$	$x^2 y^2$	$x^2 yz$	$x^2 z^2$	xy^3	$xy^2 z$	xyz^2	xz^3	y^4	$y^3 z$	$y^2 z^2$	yz^3	z^4
x^4			−3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x^3	y		1	−3	0	0	0	0	0	0	0	0	0	0	0	−1	0
x^3		z	5	0	−3	0	0	0	0	0	0	0	0	0	0	0	−1
x^2	y^2		0	1	0	−3	0	0	0	0	0	0	0	1	0	0	0
x^2	y	z	0	5	1	0	−3	0	0	0	0	0	0	1	0	3	0
x^2		z^2	0	0	5	0	0	−3	0	0	0	0	0	0	1	0	3
x	y^3		0	0	0	1	0	0	−3	0	0	0	0	0	0	0	0
x	y^2	z	0	0	0	5	1	0	0	−3	0	0	0	0	0	0	0
x	y	z^2	0	0	0	0	5	1	0	0	−3	0	0	0	0	−3	0
x		z^3	0	0	0	0	0	5	0	0	0	−3	0	0	0	0	−3
	y^4		0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
	y^3	z	0	0	0	0	0	0	5	1	0	0	0	1	0	0	0
	y^2	z^2	0	0	0	0	0	0	0	5	1	0	−5	0	1	1	0
	y	z^3	0	0	0	0	0	0	0	0	5	1	0	−5	0	1	1
		z^4	0	0	0	0	0	0	0	0	0	5	0	0	−5	0	1

The determinant of the above A matrix is zero. If the determinant of the M matrix is nonzero, this would imply that the system has a solution.

Step 7: Creating the *M* Matrix

The denominator of the Macaulay resultant is the determinant of the M matrix. The M matrix is a submatrix of the A matrix. It consists of the elements that have row and column monomial labels which are *not reduced.* Recall that a monomial is not reduced if it has more than one variable with a big exponent.

The size of the M matrix equals the size of the A matrix minus D, where

$$D = \sum_{i=1}^{m} \prod_{i \neq j} d_j$$

In our example,

$$D = d_2 d_3 + d_1 d_3 + d_1 d_2 = (2)(3) + (1)(3) + (1)(2) = 11$$

so that the size of the M matrix is 15 − 11 = 4. The actual M matrix for f_1, f_2, and f_3 is

	x^2y^2	xy^3	xy^2z	xz^3
x^2y^2	−3	0	0	0
xy^3	1	−3	0	0
xy^2z	5	0	−3	0
xz^3	0	0	0	−3

The determinant of this M matrix yields a value of 81. Since the determinant of the A matrix was zero, the Macaulay resultant is zero, which implies that there is a solution to our system. The following plot of the three polynomials (f_1, f_2, and f_3) confirms that there is a common point at $x = 2$ and $y = 1$ (see Fig. 2).

Sometimes both the A matrix and the M matrix have zero determinants. This indeterminacy can often be circumvented if the polynomials are first written with symbolic coefficients. The determinants of the A and M matrices are obtained, polynomial division is performed, and then at the end, the symbolic coefficients are replaced by their numerical values to check if the resultant is zero. Since one does not know ahead of time whether or not this "division by zero" condition will arise, the symbolic coefficient approach is the best strategy. It is also often sufficient to treat just a subset of the coefficients symbolically—sometimes as few as a single symbolic coefficient will remove the indeterminacy.

The *U* Resultant

For problems with as many inhomogeneous equations as variables, the U resultant can often be used to solve for the point solutions. The three polynomial equations f_1, f_2, f_3, do not satisfy these conditions, since there are three equations in two

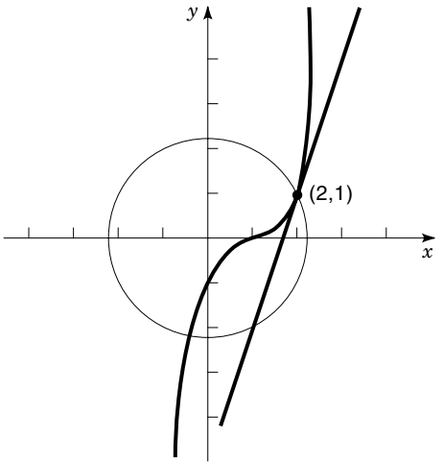


Figure 2. Common solution in our example.

inhomogeneous variables, x and y. However, if we take just the first two equations, namely f_1 and f_2, we would have a system with as many equations as variables.

The given equations must first be homogenized. This adds one additional variable. We then add one additional equation to the system. This equation is called the U equation. If x and y are the given variables and z is the homogenizing variable, then the U equation takes the form

$$u_1x + u_2y + u_3z = 0$$

The Macaulay resultant R is then computed for these $m + 1$ equations, treating the u_i as symbolic coefficients. The result is called the U resultant. Notice that R will be a polynomial in the u_i's and the coefficients of the original equations.

After R is determined, it is factored into linear factors. For each linear factor there is a point solution of the original system of equations. The coordinates of each solution are given as ratios of the coefficients of the u_i's. The denominator is always the coefficient of the u_i associated with the homogenizing variable. In our example this is the coefficient of u_3. Thus

$$x = \frac{\text{coefficient of } u_1}{\text{coefficient of } u_3} \quad \text{and} \quad y = \frac{\text{coefficient of } u_2}{\text{coefficient of } u_3}$$

For example, if a linear factor turned out to be

$$u_1 - u_2 - u_3$$

then the coordinates of the associated solution would be

$$x = \frac{+1}{-1} = -1 \quad \text{and} \quad y = \frac{-1}{-1} = +1$$

Example 4. As mentioned above, the polynomial equation system f_1, f_2, f_3 is overdetermined ($n - m = -1$). However, we can use the U resultant to solve f_1 and f_2 for x and y ($n - m = 0$). In this example, we will also demonstrate the symbolic approach alluded to in the previous section. Recall that the homogenized form of f_1 and f_2 is

$$f_1 = y - 3x + 5z = 0$$
$$f_2 = x^2 + y^2 - 5z^2 = 0$$

Rewriting these two equations with symbolic coefficients and including the U equation yields

$$f_1 = a_1x + b_1y + c_1z = 0$$
$$f_2 = a_2x^2 + b_2y^2 + c_2z^2 = 0$$
$$U = u_1x + u_xy + u_3z = 0$$

where $a_1 = -3$, $b_1 = 1$, $c_1 = 5$, $a_2 = 1$, $b_2 = 1$, and $c_2 = -5$.

The U resultant is calculated in the same way as the Macaulay resultant, that is, with the A matrix and the M matrix, except now we are using symbolic coefficients.

	x^2	xy	xz	y^2	yz	z^2
x^2	a_1	0	0	a_2	0	0
xy	b_1	a_1	0	0	u_1	0
xz	c_1	0	a_1	0	0	u_1
y^2	0	b_1	0	b_2	u_2	0
yz	0	c_1	b_1	0	u_3	u_2
z^2	0	0	c_1	c_2	0	u_3

The corresponding M matrix is a single element, namely a_1.

The determinant of M is divided into the determinant of A to obtain the U resultant. Finally, the symbolic coefficients are replaced by their numeric equivalents. (This could have been done from the outset, unless a_1 had been zero.) The result is

$$10(u_1 - 2u_2 + u_3)(2u_1 + u_2 + u_3)$$

This yields two solutions

Solution 1

$$x = \frac{\text{coefficient of } u_1}{\text{coefficient of } u_3} = \frac{+1}{-1} = -1 \quad \text{and}$$
$$y = \frac{\text{coefficient of } u_2}{\text{coefficient of } u_3} = \frac{-2}{+1} = -2$$

Solution 2

$$x = \frac{\text{coefficient of } u_1}{\text{coefficient of } u_3} = \frac{+2}{+1} = +2 \quad \text{and}$$
$$y = \frac{\text{coefficient of } u_2}{\text{coefficient of } u_3} = \frac{+1}{+1} = +1$$

We remark that the U resultant will be identically zero and give no information, if the set of common solutions contains a component of excess dimension one or more. Moreover, this component may be at infinity where the homogenizing variable is zero.

The Generalized Characteristic Polynomial Approach

The generalized characteristic polynomial (GCP) approach (2) avoids the problem of components of excess dimension in the set of solutions. It can be used together with the U resultant, which was discussed previously. If the U resultant leads to an indeterminant (0/0) form even when symbolic coefficients are used, an "excess" solution exists. The GCP takes the form

$$R = \frac{\det|A - sI|}{\det|M - sI|}$$

where A and M are the matrices defined earlier, s is a perturbation parameter, and I is the identity matrix.

One way to carry out the above operation is the following:

1. Set up the A matrix (as described previously). Subtract s along the diagonal. Evaluate the determinant. Retain the coefficient of the lowest surviving power of s.
2. Repeat step 1 for the M matrix.
3. Divide the result of step 1 by the result of step 2.

All of these multiresultant techniques have one characteristic in common. They require that there be one more equation than variable, $n - m = -1$. If there are as many equations as variables $n - m = 0$, the U equation is added and the effective situation is again $n - m = -1$. If there are more variables than equations ($n - m$ is a positive integer), then enough of these variables must be regarded as parameters in the coefficients so that effectively $n - m = -1$. Geometrically this amounts to projecting the locus of solutions to a hypersurface in a lower-dimensional space. Finally, if the number of equations exceeds the number of variables by more than one ($n - m \leq -2$), then some technique other than the multiresultant techniques noted earlier (e.g., a system of multiresultants) must be employed to determine if a solution exists.

INVARIANT POLYNOMIALS

In this section we will consider polynomials invariant under various transformation groups. Such polynomials have important applications in computer vision and image understanding. We will consider several specific cases rather than develop the general theory.

Affine Invariants of Point Sets in the Plane

Let $P_i = (x_i, y_i)$, $i = 1, \ldots, n$, be a set of n points in the plane $\mathbb{R}^2$. We will assume that these points are in general position, which means that no three are collinear. The group of affine transformations of the plane can be represented by a group of 3×3 matrices:

$$\mathrm{AFF}(2, \mathbb{R}) = \left\{ \begin{pmatrix} a & b & \xi_1 \\ c & d & \xi_2 \\ 0 & 1 & 1 \end{pmatrix} ; ad - bc \neq 0; a, b, c, d, \xi_1, \xi_2 \in \mathbb{R} \right\}$$

These affine transformations act on the plane by sending the point (x, y) to the point $(ax + by + \xi_1, cx + dy + \xi_2)$. In matrix terms this is

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} a & b & \xi_1 \\ c & d & \xi_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$\mathrm{AFF}(2, \mathbb{R})$ is a six-dimensional Lie group that is precisely the group of all transformations of $\mathbb{R}^2$ that preserve collinearity, that is, transform straight lines to straight lines.

The set of all ordered n-tuples of points in $\mathbb{R}^2$ is parametrized by $\mathbb{R}^2 \times \cdots \times \mathbb{R}^2 \cong \mathbb{R}^{2n}$ with coordinates $(x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$. Those ordered n-tuples which are in general position form a dense open subset U of $\mathbb{R}^{2n}$.

The group $\mathrm{AFF}(2, \mathbb{R})$ acts diagonally on $\mathbb{R}^{2n}$ and on U. We are interested in rational expressions that are invariant under the group action

$$\frac{p(x_1, y_1, x_2, y_2, \ldots, x_n, y_n)}{q(x_1, y_1, x_2, x_2, \ldots, x_n, y_n)}$$

Here p and q are polynomials with real coefficients. The invariant expressions will take the same value if (x_i, y_i) is replaced by $(ax_i + by_i + \xi_1, cx_i + dy_i + \xi_2)$ for every $i = 1, \ldots, n$ and for every choice of a, b, c, d, ξ_1, and ξ_2 with $ad - bc \neq 0$.

Example 5. Consider the area of a triangle formed by three of our points—say (x_1, y_1), (x_2, y_2), and (x_3, y_3). This area is

$$\frac{1}{2} \left| \det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \right|$$

After applying an affine transformation this area becomes

$$\frac{1}{2} \left| \det \left[\begin{pmatrix} a & b & \xi_1 \\ c & d & \xi_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \right] \right| = \frac{1}{2} |ad - bc| \left| \det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \right|$$

Note that the absolute value signs are not necessary because we can permute the columns of the matrix to change sign. Also note that an affine transformation has a constant Jacobian determinant, namely $|ad - bc|$, which measures the "distortion" of areas.

It is clear that the ratio of the areas of two such triangles, or the ratio of two such determinants, is an invariant:

$$\frac{\det \begin{pmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ 1 & 1 & 1 \end{pmatrix}}{\det \begin{pmatrix} x_\ell & x_m & x_s \\ y_\ell & y_m & y_s \\ 1 & 1 & 1 \end{pmatrix}}$$

We can form

$$\binom{n}{3}$$

such triangles and after dividing by the area of one of them—say the one formed by the first three points or the one of largest area—we can get

$$\binom{n}{3} - 1$$

invariants. These, however, are not all independent. For example, consider the case of four points P_1, P_2, P_3, P_4 in general position in $\mathbb{R}^2$. We can find a unique affine transformation that takes P_1 to (0, 0), P_2 to (1, 0), and P_3 to (0, 1), detailed later. The fourth point P_4 will go to some point not on the triangle of lines:

$$x = 0, \quad y = 0, \quad x + y = 1$$

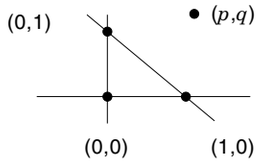


Figure 3. Transformed points.

For simplicity assume that P_4 goes to (p, q) with $p > 0$, $q > 0$, and $p + q > 1$ (see Fig. 3). Our

$$\binom{4}{3} = 4$$

triangles have areas $1/2$, $p/2$, $q/2$, and $(p + q - 1)/2$. We see that the

$$\binom{4}{3} - 1 = 3$$

ratios p, q, and $p + q - 1$ are not independent—although any two of them are.

The mathematical interpretation of these invariants is straightforward, although somewhat abstract. They are functions on the quotient space

$$U/\mathrm{AFF}(2, \mathbb{R})$$

obtained by identifying those n-tuples of points in general position that can be transformed into each other by an affine transformation.

We can specifically determine this quotient because on each orbit there is a unique n-tuple with $P_1 = (0, 0)$, $P_2 = (1, 0)$, and $P_3 = (0, 1)$. This can be seen by noting that there is a unique affine transformation which carries (x_1, y_1) to $(0, 0)$, (x_2, y_2) to $(1, 0)$ and (x_3, y_3) to $(0, 1)$. The uniqueness is clear because the only affine transformation

$$\begin{pmatrix} a & b & \xi_1 \\ c & d & \xi_2 \\ 0 & 0 & 1 \end{pmatrix}$$

which fixes $(0, 0)$, $(1, 0)$, and $(0, 1)$ is

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Existence is also easy. We translate (x_1, y_1) to $(0, 0)$ by

$$\begin{pmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{pmatrix}$$

This carries (x_2, y_2) to $(x_2 - x_1, y_2 - y_1)$ and (x_3, y_3) to $(x_3 - x_1, y_3 - y_1)$. We then construct a 2×2 invertible matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

which carries these two vectors to $(1, 0)$ and $(0, 1)$ respectively. Specifically, we need

$$\begin{aligned} a(x_2 - x_1) + b(y_2 - y_1) &= 1 \\ c(x_2 - x_1) + d(y_2 - y_1) &= 0 \end{aligned}$$

and

$$\begin{aligned} a(x_3 - x_1) + b(y_3 - y_1) &= 0 \\ c(x_3 - x_1) + d(y_3 - y_1) &= 1 \end{aligned}$$

Solving this system of four equations in four unknowns yields

$$a = \frac{y_3 - y_1}{\det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} \quad c = \frac{-(y_2 - y_1)}{\det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}}$$

$$b = \frac{-(x_3 - x_1)}{\det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} \quad d = \frac{x_2 - x_1}{\det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}}$$

The composition

$$\begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} a & b & -x_1 a - y_1 b \\ c & d & -x_1 c - y_1 d \\ 0 & 0 & 1 \end{pmatrix}$$

is the desired transformation.

A simple calculation shows that this transformation sends (x, y) to

$$\left(\frac{\det \begin{pmatrix} x_1 & x & x_3 \\ y_1 & y & y_3 \\ 1 & 1 & 1 \end{pmatrix}}{\det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}}, \quad \frac{\det \begin{pmatrix} x_1 & x_2 & x \\ y_1 & y_2 & y \\ 1 & 1 & 1 \end{pmatrix}}{\det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} \right)$$

which makes it obvious that (x_1, y_1) goes to $(0, 0)$, (x_2, y_2) goes to $(1, 0)$, and (x_3, y_3) goes to $(0, 1)$. Notice also that the remaining $n - 3$ points (x_4, y_4), . . ., (x_n, y_n) are sent to points whose coordinates are invariants. These $2n - 6$ invariant values serve as coordinate functions on the quotient space, which is clearly isomorphic to an open set W of $\mathbb{R}^{2n-6}$:

$$U/\mathrm{AFF}(2, \mathbb{R}) \cong W$$

The central theorem is the following.

Theorem. Any affine invariant expression

$$\frac{p(x_1, y_1, \ldots, x_n, y_n)}{q(x_1, y_1, \ldots, x_n, y_n)}$$

is a rational function of the invariant coordinate functions noted previously.

An equivalent formulation is that every invariant is a rational function of $2n - 6$ ratios of areas of triangles, for

example:

$$\frac{\text{area}(P_i, P_1, P_2)}{\text{area}(P_1, P_2, P_3)} \quad \text{and} \quad \frac{\text{area}(P_i, P_1, P_3)}{\text{area}(P_1, P_2, P_3)}$$

for $i = 4, \ldots, n$. Note that we do not need to consider the ratio

$$\frac{\text{area}(P_i, P_2, P_3)}{\text{area}(P_1, P_2, P_3)}$$

because, as shown above, for the four points P_1, P_2, P_3, P_i, the areas of the

$$\binom{4}{3} = 4$$

triangles are linearly related and therefore the three ratios are linearly dependent.

Affine Invariants of Two Points and Two Lines in the Plane

Consider two lines L_1 and L_2 and two points P_1 and P_2 in the plane in general position. For our purposes general position means that L_1 and L_2 are not parallel and that P_1 is not on either L_1 or L_2. Given another set of two lines L_1' and L_2' and two points P_1' and P_2', we would like to know if there is an affine transformation of the plane that carries L_i to L_i' and P_i to P_i' for $i = 1, 2$. As we shall see, this will be true if and only if the two invariants constructed below have the same value for both of the geometric configurations.

The geometric configurations of interest (an ordered pair of lines and an ordered pair of points in general position in the plane) are parametrized by an open subset U of $\mathbb{P}^2_{\mathbb{R}} \times \mathbb{P}^2_{\mathbb{R}} \times \mathbb{R}^2 \times \mathbb{R}^2$. (Recall that lines $ax + by + c = 0$ in the plane are parametrized by points $(a : b : c) \in \mathbb{P}^2_{\mathbb{R}}$, real projective two-space.) The affine group AFF(2, $\mathbb{R}$) acts on $\mathbb{R}^2$ in a way that preserves lines, and so acts on U. Note that

$$M = \begin{pmatrix} \tilde{a} & \tilde{b} & \xi_1 \\ \tilde{c} & \tilde{d} & \xi_2 \\ 0 & 0 & 1 \end{pmatrix}$$

acts on points by sending

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \text{ to } M \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

but it acts on lines by sending

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} \text{ to } (M^T)^{-1} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Since $\dim_{\mathbb{R}} U = 8$ and $\dim_{\mathbb{R}}$ AFF(2, $\mathbb{R}$) $= 6$, we expect a two-dimensional quotient AFF(2, $\mathbb{R}$)$\backslash U$. This quotient is, as we shall see, diffeomorphic to $\mathbb{R}^2$.

Let Q be the point of intersection of L_1 and L_2. We can find an affine transformation that moves Q to the origin, L_1 to the x axis, and L_2 to the y axis. In fact, if L_1 is given by $a_1x + b_1y + c_1 = 0$ and L_2 is given by $a_2x + b_2y + c_2 = 0$, then

$$M = \begin{pmatrix} a_2 & b_2 & c_2 \\ a_1 & b_1 & c_1 \\ 0 & 0 & 1 \end{pmatrix}$$

is one such transformation. Having moved L_1 and L_2 to the x and y axes, respectively, we can still act by transformations of the form:

$$N = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

If P_1 originally had coordinates (x_1, y_1), then after applying M, we will have the point $(a_2x_1 + b_2y_1 + c_2, a_1x_1 + b_1y_1 + c_1)$. Our general position assumption implies that neither coordinate is zero. Setting

$$\lambda_1 = \frac{1}{a_2x_1 + b_2y_1 + c_1}$$
$$\lambda_2 = \frac{1}{a_1x_1 + b_1y_1 + c_2}$$

in N will move this point to $(1, 1)$. Putting M and N together yields an affine transformation

$$T = NM = \begin{pmatrix} \dfrac{a_2}{a_2x_1 + b_2y_1 + c_2} & \dfrac{b_2}{a_2x_1 + b_2y_1 + c_2} & \dfrac{c_2}{a_2x_1 + b_2y_1 + c_2} \\ \dfrac{a_1}{a_1x_1 + b_1y_1 + c_1} & \dfrac{b_1}{a_1x_1 + b_1y_1 + c_1} & \dfrac{c_1}{a_1x_1 + b_1y_1 + c_1} \\ 0 & 0 & 1 \end{pmatrix}$$

which moves L_1 to the x axis, L_2 to the y axis, and P_1 to $(1, 1)$. No degrees of freedom remain, so this must be the unique such affine transformation.

Suppose P_2 originally had coordinates (x_2, y_2); then the coordinates of P_2 after transformation by T parametrize the quotient AFF(2, $\mathbb{R}$)$\backslash U$ and are the essential invariants

$$I_1 = \frac{a_2x_2 + b_2y_2 + c_2}{a_2x_1 + b_2y_1 + c_2} \quad \text{and} \quad I_2 = \frac{a_1x_2 + b_1y_2 + c_1}{a_1x_1 + b_1y_1 + c_1}$$

In general an invariant takes the form

$$\frac{p(a_1, b_1, c_1, a_2, b_2, c_2, x_1, y_1, x_2, y_2)}{q(a_1, b_1, c_1, a_2, b_2, c_2, x_1, y_1, x_2, y_2)}$$

where p and q are polynomials that are homogeneous of the same degree in a_1, b_1, c_1 and in a_2, b_2, c_2. It can be shown that every such invariant expression is a rational function of the two fundamental invariants I_1 and I_2.

Example 6

$$L_1 : x - y + 1 = 0$$
$$L_2 : 2x - y = 0$$
$$P_1 : (1, 0)$$
$$P_2 : (0, 2)$$
$$T = \begin{pmatrix} 1 & -1/2 & 0 \\ 1/2 & -1/2 & 1/2 \\ 0 & 0 & 1 \end{pmatrix}$$
$$I_1 = -1 \quad I_2 = -1/2$$

Projective Invariants of Five Points in the Plane

Consider an ordered set of five points P_1, P_2, P_3, P_4, P_5 in the plane $\mathbb{R}^2$. We regard $\mathbb{R}^2$ as an open dense subset of the projective plane $\mathbb{P}^2_{\mathbb{R}}$. We assume that these points are in general position, that is, that no three are collinear. Notice that our geometry is parametrized by a 10-dimensional space, while the group of projective transformations has dimension 8. Thus we expect a two-dimensional quotient, and therefore two fundamental invariants.

To determine these invariants, observe that there is a unique projective transformation taking P_1 to $(1:0:0) \in \mathbb{P}^2_{\mathbb{R}}$, P_2 to $(0:1:0)$, P_3 to $(0:0:1)$, and P_4 to $(1:1:1)$. The point P_5 will be sent to some point $(a:b:c)$ under this transformation; moreover none of a, b, or c will be zero by the general position assumption. The ratios

$$I_1 = \frac{a}{c} \quad \text{and} \quad I_2 = \frac{b}{c}$$

will be the basic invariants. Any other invariant we might construct will be a rational function of these two.

The matrix

$$M = \frac{1}{\det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} \begin{pmatrix} y_2 - y_3 & x_3 - x_2 & x_2 y_3 - x_3 y_2 \\ y_3 - y_1 & x_1 - x_3 & x_3 y_1 - y_3 x_1 \\ y_1 - y_2 & x_2 - x_1 & x_1 y_2 - x_2 y_1 \end{pmatrix}$$

sends P_1 to $(1:0:0)$, P_2 to $(0:1:0)$, and P_3 to $(0:0:1)$. However it sends $P_4 = (x_4 : y_4 : 1)$ to

$$Q_4 = \begin{pmatrix} \det \begin{pmatrix} x_4 & x_2 & x_3 \\ y_4 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \Big/ \det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \\ \det \begin{pmatrix} x_1 & x_4 & x_3 \\ y_1 & y_4 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \Big/ \det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \\ \det \begin{pmatrix} x_1 & x_2 & x_4 \\ y_1 & y_2 & y_4 \\ 1 & 1 & 1 \end{pmatrix} \Big/ \det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \end{pmatrix}$$

(Note that none of these determinants are zero by the general position assumption.) Multiplying M by the diagonal matrix whose entries are the reciprocals of the components of Q_4 gives

$$\begin{pmatrix} \dfrac{y_2 - y_3}{\det \begin{pmatrix} x_4 & x_2 & x_3 \\ y_4 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} & \dfrac{x_3 - x_2}{\det \begin{pmatrix} x_4 & x_2 & x_3 \\ y_4 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} & \dfrac{x_2 y_2 - x_3 y_2}{\det \begin{pmatrix} x_4 & x_2 & x_3 \\ y_4 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} \\ \dfrac{y_3 - y_1}{\det \begin{pmatrix} x_1 & x_4 & x_3 \\ y_1 & y_4 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} & \dfrac{x_1 - x_3}{\det \begin{pmatrix} x_1 & x_4 & x_3 \\ y_1 & y_4 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} & \dfrac{x_3 y_1 - y_3 x_1}{\det \begin{pmatrix} x_1 & x_4 & x_3 \\ y_1 & y_4 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} \\ \dfrac{y_1 - y_2}{\det \begin{pmatrix} x_1 & x_2 & x_4 \\ y_1 & y_2 & y_4 \\ 1 & 1 & 1 \end{pmatrix}} & \dfrac{x_2 - x_1}{\det \begin{pmatrix} x_1 & x_2 & x_4 \\ y_1 & y_2 & y_2 \\ 1 & 1 & 1 \end{pmatrix}} & \dfrac{x_1 y_2 - x_2 y_1}{\det \begin{pmatrix} x_1 & x_2 & x_4 \\ y_1 & y_2 & y_4 \\ 1 & 1 & 1 \end{pmatrix}} \end{pmatrix}$$

This is the desired projective transformation in the group PGL(3, $\mathbb{R}$) of all projective transformations of the projective plane (essentially 3×3 invertible matrices modulo scalars. It takes $P_5 = (x_5, y_5)$ to

$$\left(\frac{\det \begin{pmatrix} x_5 & x_2 & x_3 \\ y_5 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}}{\det \begin{pmatrix} x_4 & x_2 & x_3 \\ y_4 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} : \frac{\det \begin{pmatrix} x_1 & x_5 & x_3 \\ y_1 & y_5 & y_3 \\ 1 & 1 & 1 \end{pmatrix}}{\det \begin{pmatrix} x_1 & x_4 & x_3 \\ y_1 & y_4 & y_3 \\ 1 & 1 & 1 \end{pmatrix}} : \frac{\det \begin{pmatrix} x_1 & x_2 & x_5 \\ y_1 & y_2 & y_5 \\ 1 & 1 & 1 \end{pmatrix}}{\det \begin{pmatrix} x_1 & x_2 & x_4 \\ y_1 & y_2 & y_4 \\ 1 & 1 & 1 \end{pmatrix}} \right)$$

This yields the invariants

$$I_1 = \frac{\det \begin{pmatrix} x_5 & x_2 & x_3 \\ x_5 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \det \begin{pmatrix} x_1 & x_2 & x_4 \\ y_1 & y_2 & y_4 \\ 1 & 1 & 1 \end{pmatrix}}{\det \begin{pmatrix} x_4 & x_2 & x_3 \\ y_4 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \det \begin{pmatrix} x_1 & x_2 & x_5 \\ y_1 & y_2 & y_5 \\ 1 & 1 & 1 \end{pmatrix}}$$

$$I_2 = \frac{\det \begin{pmatrix} x_1 & x_5 & x_3 \\ y_1 & y_5 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \det \begin{pmatrix} x_1 & x_2 & x_4 \\ y_1 & y_2 & y_4 \\ 1 & 1 & 1 \end{pmatrix}}{\det \begin{pmatrix} x_1 & x_4 & x_3 \\ y_1 & y_4 & y_3 \\ 1 & 1 & 1 \end{pmatrix} \det \begin{pmatrix} x_1 & x_2 & x_5 \\ y_1 & y_2 & y_5 \\ 1 & 1 & 1 \end{pmatrix}}$$

Other ratios would also be just as good. Moreover, any invariant will be a rational expression in these.

Notice that the individual determinants are (up to sign and a factor of $\frac{1}{2}$) the areas of certain triangles. Thus our projective invariants are ratios of products of areas of certain pairs of triangles and are affine invariants as they should be.

Affine Invariants of Five Points in Space

Let P_1, P_2, P_3, P_4, P_5 be an ordered 5-tuple of distinct points in space $\mathbb{R}^3$. Say the coordinates of $P_i = (x_i, y_i, z_i)$. We will assume that the points are in general position, so that no four are coplanar (implying no three are collinear).

The fundamental affine invariants for any number of points in the space are formed from the ratios of the volumes of two tetrahedrons in space. If P_i, P_j, P_k, P_l are the vertices,

the volume is

$$\det \begin{pmatrix} x_i & x_j & x_k & x_\ell \\ y_i & y_j & y_k & y_\ell \\ z_i & z_j & z_k & z_\ell \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

up to a factor of $\pm 1/6$.

Five points in general position yield

$$\binom{5}{4} = 5$$

tetrahedrons. Under an affine transformation of $\mathbb{R}^3$ these five volumes all scale by the same constant factor. Thus we can regard the volumes as giving a well-defined point in $\mathbb{P}^4_{\mathbb{R}}$. However, the points we get in $\mathbb{P}^4_{\mathbb{R}}$, as we run through all 5-tuples of points in $\mathbb{R}^3$, lie in a hyperplane, that is, they all satisfy a fixed linear relation. This can be seen by expanding the following determinant

$$0 = \det \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ y_1 & y_2 & y_3 & y_4 & y_5 \\ z_1 & z_2 & z_3 & z_4 & z_5 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

along the bottom row.

Thus we have only four independent volumes. Normalizing one of them to one yields three ratios of volumes of tetrahedra, which are the fundamental affine invariants of our five points. (This squares with the fact that our geometry (5 points in general position in space) is parametrized by a 15-dimensional space while AFF(3, $\mathbb{R}$) has dimension 12.

BIBLIOGRAPHY

1. B. L. van der Waerden, *Modern Algebra,* City: Frederick Ungar, 1949 and 1950, Vols. 1 and 2.
2. J. Canny, Generalized characteristic polynomials, *J. Symbolic Comput.* **9**: 241–250, 1990.

Reading List

S. Barnett, *Polynomials and Linear Control Systems,* New York: Marcel Dekker, 1983.

E. J. Barbeau, *Polynomials,* New York: Springer-Verlag, 1989.

D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms,* Undergraduate Texts in Mathematics, New York: Springer, 1992.

W. Fulton, *Algebraic Curves,* Menlo Park, CA: Benjamin, 1969.

I. M. Gelfand, et al., *Discriminants, Resultants, and Multidimensional Determinants,* Boston: Birkhäuser, 1994.

R. F. Gleeson and R. M. Williams, *A Primer on Polynomial Resultants,* Naval Air Development Center Tech. Rep., 1991, ADA 246 883.

J. Harris, *Algebraic Geometry: A First Course,* Graduate Texts in Mathematics, Vol. 133, New York: Springer, 1992.

A. P. Morgan, *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems,* Englewood Cliffs, NJ: Prentice Hall, 1987.

B. Roth, Computation in kinematics, in J. Angeles et al. (eds.), *Computational Kinematics,* Norwell, MA: Kluwer, 1993.

G. Salmon, *Lessons Introductory to the Modern Higher Algebra,* 5th ed., New York: Chelsea, 1932.

J. P. Serre, *A Course in Arithmetic,* Graduate Texts in Mathematics, Vol. 7, New York: Springer, 1971.

PETER F. STILLER
Texas A & M University

PONTRYAGIN MAXIMUM PRINCIPLE. See OPTIMAL CONTROL.

PORTABLE COMPUTERS. See LAPTOP COMPUTERS.

Probabilistic Logic
Standard Article
Michael Pittarelli[1]
[1]SUNY Institute of Technology, Utica, NY
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2444
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (110K)

Abstract

The sections in this article are

| | |
Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

PROBABILISTIC LOGIC

A deductive argument is a claim of the form: If P_1, P_2, ..., and P_n are true, then C is true.

The statements P_1, P_2, ..., P_n are the premises of the argument; C is the conclusion. It is the logical structure of the collection of premises and of the conclusion that determines whether the argument is valid (i.e., whether the claim it makes is the case) or invalid.

For example, the argument with premises

P_1: If it is raining, then the ground is wet.

P_2: It is raining.

and conclusion

C: The ground is wet.

is valid. The argument may be expressed symbolically as

P_1: $R \rightarrow G$

P_2: R

C: G

For every consistent assignment of truth values to the propositional variables R and G under which the premises evaluate to true, the conclusion also evaluates to true; the implication with the conjunction of the premises as antecedent and the conclusion as consequent is a tautology, as the following truth table indicates.

The argument

P_1: The barometric pressure is 28.09

P_2: The relative humidity is 98%

P_3: The temperature is 81°F

C: It is raining is not valid. It is (logically) possible for the premises to be true and the conclusion to be false simultaneously.

The term "probabilistic logic" is typically used to refer to systems of logic that permit the attachment of probabilities to the premises and the conclusion of an argument. Deduction assumes full belief in the truth of the premises of an argument. However, unless the premises are tautologies (and therefore without empirical content), one cannot be absolutely certain of their truth. Although, for practical purposes, we may accept the truth of the statement "If it is raining, then the ground is wet," we may be wrong. Without our knowing it, someone may have pitched a tent over the piece of ground we are referring to, or put a tarp over it, etc.

R	G	$(R \rightarrow G$	$\&$	$R)$	$\rightarrow$	G
T	T	T	T	T	T	T
T	F	F	F	T	T	F
F	T	T	F	F	T	T
F	F	T	F	F	T	F

2 PROBABILISTIC LOGIC

Systems of probabilistic logic vary with respect to how the uncertainty associated with the premises of an argument is propagated to the conclusion. We will first discuss the version presented by Nilsson (1).

Suppose that we assign probabilities to the premises of the "ground is wet" argument:

$$p(R \rightarrow G) = .9$$
$$p(R) = .6$$

Note that the probabilities of the premises need not sum to 1. (They are not necessarily mutually exclusive and exhaustive.) However, for consistency, it is required that

$$p(R \rightarrow G) \geq 1 - p(R)$$

because R → G is logically equivalent to ~R v G,

$$p(\sim R) = 1 - p(R)$$

and

$$p(\sim R \vee G) \geq p(\sim R)$$

From the probabilities associated with the premises, a range of probabilities can be determined for the conclusion "The ground is wet" (G). The probabilities of the premises are constraints on the probabilities of consistent assignments of truth values to the sentences $R \rightarrow G$, R and G. Each consistent assignment of truth values corresponds to a set of possible worlds: the worlds in which R is true, G is true, and $R \rightarrow G$ is true; those in which R is true, G is false, and $R \rightarrow G$ is false; etc. For these sentences, there are four sets of possible worlds. The worlds in a given set are equivalent with respect to the truth values of the three sentences. The sets of possible worlds are mutually exclusive and exhaustive. (They partition the set of all possible worlds.) Therefore, their probabilities sum to 1. The probabilities of the premises are (linear) constraints on the probabilities of the sets of possible worlds.

Nilsson's method for identifying the sets of possible worlds involves constructing a semantic tree. Each node in the (binary) tree corresponds to an assignment of truth values to some subset (empty, in the case of the root) of the premises and conclusion. The level of the root node is 0. If a node is at level k, then its children are at level $k + 1$. The sentences are ordered arbitrarily. Label them

$$S_1, S_2, \ldots, S_{n+1}$$

(for an argument with n premises). We can associate the set of all possible worlds with the root node. The left child of the root corresponds to the set of possible worlds in which S_1 is true. The right child is the set of worlds in which S_1 is false. In general, for a node X at level k, its left (resp., right) child represents the intersection of the set of worlds in which S_{k+1} is true (resp., false) with the set of worlds associated with X. A terminal node, therefore, represents a simultaneous assignment of truth values to the premises and conclusion of the argument.

Without pruning, the semantic tree will have 2^{n+1} terminal nodes (each corresponding to a truth assignment over all of the sentences of the argument) and $2^{n+2} - 1$ nodes altogether. However, not all the (full or partial) truth assignments are consistent. Each (partial) truth assignment can be checked for consistency. If the assignment is inconsistent (i.e., would require that a propositional variable simultaneously had the values false

and true), the children of the node corresponding to it are not generated. For example, the partial assignment

$$R \rightarrow G/F, R/F$$

is not consistent. There is no assignment of truth values to the propositional variables R and G under which both sentences evaluate to false.

In the worst case, when the sentences are logically independent, no pruning will be possible. However, the sentences in real arguments are unlikely to be independent. (Despite this, for arguments of realistic size, the method is intractable. We will later discuss refinements that use more effectively the dependencies among the sentences and that take into account the decision-making context in which the probabilistic inference is being made.)

For the "ground is wet" argument, there are four consistent truth assignments (out of eight combinatorially possible assignments). These four assignments correspond to four sets of possible worlds (W_1, ..., W_4):

	W_1	T	T	T
$R \rightarrow G$	W_2	F	T	F
R	W_3	T	F	T
G	W_4	T	F	F

Recall that

$$p(R \rightarrow G) = .9$$

and

$$p(R) = .6$$

We would like to infer the probability of G, "the ground is wet." (We might use the inferred probability to decide whether or not to put on boots before going outside. Modifications to the method that take into account the use that is to be made of the probability of the conclusion will be discussed later.) Although we do not know the probability of G, we know that its probability is the sum of the probabilities of the sets of worlds in which it is true:

$$p(G) = p(W_1) + p(W_3)$$

Similarly,

$$p(R \rightarrow G) = p(W_1) + p(W_3) + p(W_4)$$

and

$$p(R) = p(W_1) + p(W_2)$$

4 PROBABILISTIC LOGIC

Let p_j abbreviate $p(W_j)$. The set of solutions to the system of linear equations below is the set of probability functions compatible with the probabilities associated with the premises of our argument:

$$\begin{aligned} p_1 + p_2 + p_3 + p_4 &= 1 \\ p_1 + p_3 + p_4 &= .9 \\ p_1 + p_2 &= .6 \end{aligned}$$

Upper and lower bounds on $p(G)$ are determined via two applications of a linear programming algorithm, minimizing and maximizing the objective function $p_1 + p_3$:

$$p(G) \in [.5, .9]$$

In general, as this example illustrates, the probabilities of the premises underdetermine the probability distribution over the partition of possible worlds; thus, there is a range of probability values for the conclusion each of which (due to the linearity of the constraints and the objective function) is compatible with the probabilities of the premises.

The method may be further generalized to allow probability ranges for the premises. Suppose

$$p(R \rightarrow G) \in [.85, .95]$$

and

$$p(R) \in [.55, .65]$$

Then the constraints are in the form of a system of linear inequalities:

$$\begin{aligned} p_1 + p_2 + p_3 + p_4 &= 1 \\ p_1 + p_3 + p_4 &\geq .85 \\ p_1 + p_3 + p_4 &\leq .95 \\ p_1 + p_2 &\geq .55 \\ p_1 + p_2 &\leq .65 \end{aligned}$$

The probability interval is now

$$p(G) \in [.4, .95]$$

The method reduces to ordinary deduction when the conclusion is logically implied by the premises; the probability interval calculated in such cases is [1, 1]. However, when the conclusion is not deductively entailed by the premises, the method does not necessarily return the interval [0, 1]. The probabilities of the premises may arbitrarily strongly constrain the probability of the conclusion. For example, the argument with the single premise $A \rightarrow B$ and conclusion $B \rightarrow A$ is not valid. However, using Nilsson's methods it can be determined that

$$p(B \rightarrow A) \in [1 - p(A \rightarrow B), 1]$$

In addition, the method may be applied to arguments in first-order logic if the sentences are first Skolemized (2). However, for some arguments in first-order logic, it will not be possible to enumerate all of the sets of possible worlds.

Suppose that we are concerned with the probability of the ground being wet because we are trying to decide whether or not to put on boots. (Analogous decision problems could be devised for some autonomous system.) Suppose that the wetness of the ground is the only factor we wish to consider. There are four different relevant outcomes: we put on boots and the ground is wet, we put on boots and the ground is not wet, etc. On the standard approach (3), an agent is assumed to have a preference ranking over the outcomes. Furthermore, the relative desirability of the outcomes can be quantified, on a scale from 0 to 1, where 1 is the utility of the most preferred outcome and 0 is the utility of the least preferred outcome. There are techniques for eliciting from a human decision maker his or her utilities for the remaining outcomes, which involve consideration of hypothetical lotteries (4). Suppose that for our decision maker, the utilities are as follows:

$$U(\text{boots, wet}) = .6$$
$$U(\text{boots, not wet}) = .8$$
$$U(\text{no boots, wet}) = 0$$
$$U(\text{no boots, not wet}) = 1.0$$

Our task is to pick the action whose expected utility is maximized. Suppose that a is one of the actions under consideration and that $\{c_1, \ldots, c_m\}$ is a mutually exclusive and exhaustive set of conditions. The possible outcomes of action a are the pairs (a, c_j). On the assumption of independence of the conditions and the actions, the expected utility of action a is

$$\sum_{j=1}^{m} p(c_j) \times U(a, c_j)$$

Simple algebra is sufficient to show that the action "put on boots" has higher expected utility than the action "do not put on boots" whenever the probability of "the ground is wet" exceeds .25. Therefore, either of the two probability intervals calculated here for $p(G)$ is sufficient to determine the appropriate action.

For decision support systems that cannot accommodate interval or other set-valued probability representations, there are alternatives to the calculation of a probability interval for the conclusion of the argument. One possibility is to select the maximum entropy solution to the system of linear constraints. The maximum entropy solution can be calculated using standard optimization techniques (5). The entailed probability is then the sum of the components of the maximum entropy distribution corresponding to the possible worlds in which the conclusion is true. This approach and related alternatives are discussed by Kane (6) and Deutsch-McLeish (7).

Snow (8) explores reductions in the size of the semantic tree that can be achieved by exploiting redundancy in common inference patterns, for example, modus ponens with a conjunctive antecedent:

P_1: A_1
P_2: A_2
P_{n-1}: A_{n-1}
P_n: $(A_1 \,\&\, \cdots \,\&\, A_{n-1}) \rightarrow B$
C: B

The semantic tree method yields 2^n possible worlds for $n-1$ antecedents. Snow uses "don't care" values to reduce the number of worlds to $n+2$. Similar efficiencies are possible for other patterns of inference. However,

in the worst case, there are no exploitable redundancies, and the number of nodes in the semantic tree (number of unknowns in the linear program) is exponential in the number of premises.

Frisch and Haddawy (9) have developed a deductive system for propositional probabilistic logic with an "anytime" property. The terms "anytime" and "flexible" have been used in the computing literature to refer to algorithms that provide useful output even when interrupted and whose output improves, in some sense, as increasing amounts of resources (computing time or space) are allocated to them. In their system, the probabilities associated with the premises of an argument are intervals. The probabilities are conditional probabilities; however, unconditional probabilities can be represented by conditioning on a tautology. The conclusion or "target sentence" initially is given the probability interval [0, 1]. With each inference step (application of an inference rule), the conclusion's probability interval is narrowed.

This system has the advantage of providing a usable interval that may be narrow enough for the purpose at hand relatively quickly. To illustrate, we now discuss a simple example that uses several of Frisch and Haddawy's inference rules.

Consider a decision the outcome of which is contingent on the truth or falsity of a single probabilistically entailed sentence: "It will rain this afternoon." Suppose that the actions under consideration are "Go to the beach" and "Do not go to the beach." Utilities of the four possible outcomes are

$$U(\textbf{Go}, \text{Rain}) = 0$$
$$U(\textbf{Go}, \text{No rain}) = 1$$
$$U(\textbf{Do not}, \text{Rain}) = .8$$
$$U(\textbf{Do not}, \text{No rain}) = .2$$

The agent's knowledge is represented in part by the propositions and associated probability intervals:

(1) $p(\text{Temperature} > 85) \in [.95, 1]$
(2) $p(\text{Temperature} > 85 \rightarrow \text{Rain}) \in [.4, .6]$
(3) $p((\text{Barom. pressure} < 30 \ \& \ \text{Humidity} > 80) \rightarrow \text{Rain}) \in [.65, .95]$
(4) $p(\text{Barom. pressure} < 30) \in [.95, 1]$
(5) $p(\text{Humidity} > 80) \in [.95, 1]$
(6) $p(\text{August} \rightarrow \text{Rain}) \in [.2, 1]$
(7) $p(\text{August}) \in [1, 1]$

"Go" maximizes expected utility when $p(\text{Rain}) \leq 0.5$; "Do not go" does so for $p(\text{Rain}) \geq .5$. Neither can be ruled out at this point. However, Frisch and Haddawy's probabilistic inference rules may be applied individually to narrow the interval for $p(\text{Rain})$ until a single admissible action emerges or until it is no longer economical to continue refining (e.g., the last train to the beach is about to leave) and a choice among the admissible actions must be made using some other criterion (e.g., use maximin—pick the action whose minimum utility over the various conditions is greatest, maximize expected utility relative to the midpoint of the probability interval, etc.).

Initially, we can deduce

(1) $p(\text{Rain}) \in [0, 1]$

from the "Trivial derivation" rule: $\vdash p(\alpha|\delta) \in [0, 1]$.

We may next apply "Forward implication propagation,"

$$p(\beta|\delta) \in [x, y],\ p(\beta \rightarrow \alpha|\delta) \in [u, v] \vdash p(\alpha|\delta) \in [\min(0, x + u - 1), v]$$

to statements 1 and 2, yielding

(1) $p(\text{Rain}) \in [.35, .6]$

Although it does not have any effect at this stage, the multiple derivation rule should be applied to maintain the tightest interval for the target sentence:

$$p(\alpha|\delta) \in [x, y],\ p(\alpha|\delta) \in [u, v] \vdash p(\alpha|\delta) \in [\max(x, u), \min(y, v)]$$

Because $.5 \in [.35, .6]$, both actions remain admissible. Next, "Conjunction introduction",

$$p(\alpha|\delta) \in [x, y],\ p(\beta|\delta) \in [u, v] \vdash p(\alpha\ \&\ \beta|\delta) \in [\max(0, x + u - 1), \min(y, v)]$$

is applied to statements 4 and 5, yielding

(1) $p(\text{Barom. pressure} < 30\ \&\ \text{Humidity} > 80) \in [.9, 1]$.

Applying forward implication propagation to statements 3 and 10 gives

(1) $p(\text{Rain}) \in [.55, .95]$

Although combining statement 11 with statement 9 via the multiple derivation rule will further narrow the target interval, there is no need to do so; nor is there any need to consider statements 6 and 7. "Do not go" has emerged as uniquely admissible.

Nilsson's methods may themselves be modified to yield an anytime procedure for decision making (10). Rather than construct the linear system corresponding to the full set of sentences, increasingly larger systems may be constructed by adding sentences to the subset currently in use until a single admissible action emerges or it is necessary to choose among the currently admissible actions.

This may be illustrated with the preceding sentences and decision problem. Suppose that sentences 3 and 5 are chosen for the first iteration. Using Nilsson's semantic tree method, five sets of possible worlds are identified. Both actions are admissible. "Go" is admissible because feasible solutions to the following system of linear inequalities exist. Where p_i is the probability of set W_i of possible worlds; "Rain" is true in sets W_3 and

W_5, "Humidity > 80" is true in sets W_1, W_4 and W_5, etc.:

$$\begin{aligned}
&p_1 + p_2 + p_3 + p_4 + p_5 = 1\\
&p_2 + p_3 + p_4 + p_5 \geq .65\\
&p_2 + p_3 + p_4 + p_5 \leq .95\\
&p_1 + p_4 + p_5 \geq .95\\
&(p_3 + p_5) \times 0 + (p_1 + p_2 + p_4) \times 1 \geq (p_3 + p_5) \times 0.8\\
&\qquad + (p_1 + p_2 + p_4) \times 0.2
\end{aligned}$$

"Do not go" is also admissible because the system resulting from reversing the direction of the final inequality also has feasible solutions.

Now add sentence 4. The resulting eight sets of possible worlds may be determined by expanding only the "live" terminal nodes of the semantic tree constructed at the first iteration. (To eliminate the need for a row interchange, the root of the initial tree should represent the target sentence. One may proceed in this way as long as is necessary.) "Do not go" is now identified as uniquely admissible; feasible solutions to the following system exist but do not exist for the corresponding system for "Go":

$$\begin{aligned}
&p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8 = 1\\
&p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 \geq .65\\
&p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 \leq .95\\
&p_1 + p_2 + p_6 + p_8 \geq .95\\
&p_1 + p_3 + p_5 + p_8 \geq .95\\
&(p_1 + p_2 + p_3 + p_4) \times 0.8 + (p_5 + p_6 + p_7 + p_8) \times 0.2\\
&\qquad \geq (p_1 + p_2 + p_3 + p_4)\\
&\qquad\quad \times 0 + (p_5 + p_6 + p_7 + p_8) \times 1
\end{aligned}$$

Frisch and Haddawy's system is applicable to decision problems with an arbitrary number m of mutually exclusive conditions. The $[\frac{1}{2}\, m(m-1)+1]$ statements

$$\begin{aligned}
&p(c_1 v \cdots v c_m) \in [1, 1]\\
&p(c_1 \& c_2) \in [0, 0]\\
&\vdots\\
&p(c_{m-1} \& c_m) \in [0, 0]
\end{aligned}$$

must be included to express the mutual exclusivity of the conditions c_j. Intervals must be maintained for each of the conditions. The soundness of Frisch and Haddawy's inference rules guarantees that, at any time, the interval $[l_j, u_j]$ associated with any c_j is a superset of the tightest interval entailed (algebraically) by the full collection of sentences. Thus, the sharpest intervals available at any time yield a linear system from which it can be determined whether an action would not be admissible relative to the sharper probability bounds

computable at any later time; action a_i is (ultimately) admissible only if there exist feasible solutions to

$$
\begin{aligned}
& p(c_1) + \cdots + p(c_m) = 1 \\
& p(c_1) \geq l_1 \\
& p(c_1) \leq u_1 \\
& \vdots \\
& p(c_m) \geq l_m \\
& p(c_m) \leq u_m \\
& p(c_1) \times U(a_i, c_1) + \cdots + p(c_m) \times U(a_i, c_m) \\
& \qquad \geq p(c_1) \times U(a_1, c_1) + \cdots + p(c_m) \times U(a_1, c_m) \\
& \vdots \\
& p(c_1) \times U(a_i, c_1) + \cdots + p(c_m) \times U(a_i, c_m) \\
& \qquad \geq p(c_1) \times U(a_k, c_1) + \cdots + p(c_m) \times U(a_k, c_m)
\end{aligned}
$$

where l_j and u_j are the current bounds on $p(c_j)$ and there are k actions.

Nilsson's semantic tree method can be adapted to take into account the mutual exclusivity and exhaustiveness of multiple (i.e., more than two) conditions in a decision problem. The first m levels of the tree will correspond to the m conditions. (This facilitates the anytime adaptation of Nilsson's methods discussed previously.) At level m, there will be m live nodes, one for each of the assignments in which exactly one of the conditions is true. The remaining levels of the tree are constructed as usual.

For example, with conditions c_1, c_2, and c_3, an arbitrary number $k \geq 2$ of actions a_i, and data $p(B \rightarrow c_1) \in [0.9, 1]$ and $p(B) \in [0.8, 1]$, there are six sets of possible worlds, corresponding to the matrix

$$
\begin{array}{ll}
\mathbf{110000} & (c_1) \\
\mathbf{001100} & (c_2) \\
\mathbf{000011} & (c_3) \\
\mathbf{111010} & (B \rightarrow c_1) \\
\mathbf{100101} & (B)
\end{array}
$$

Action a_i is E-admissible iff there exist feasible solutions to the system of linear inequalities:

$$
\begin{aligned}
& p_1 + \cdots + p_6 = 1 \\
& p_1 + p_2 + p_3 + p_5 \geq .9 \\
& p_1 + p_4 + p_6 \geq .8 \\
& (p_1 + p_2) \times U(a_i, c_1) + (p_3 + p_4) \times U(a_i, c_2) + (p_5 + p_6) \\
& \qquad \times U(a_i, c_3) \geq (p_1 + p_2) \times U(a_1, c_1) + (p_3 + p_4) \times U(a_1, c_2) \\
& \qquad + (p_5 + p_6) \times U(a_1, c_3) \\
& \vdots \\
& (p_1 + p_2) \times U(a_i, c_1) + (p_3 + p_4) \times U(a_i, c_2) + (p_5 + p_6) \\
& \qquad \times U(a_i, c_3) \geq (p_1 + p_2) \times U(a_k, c_1) + (p_3 + p_4) \times U(a_k, c_2) \\
& \qquad + (p_5 + p_6) \times U(a_k, c_3)
\end{aligned}
$$

10 PROBABILISTIC LOGIC

Although they are not the concern of this article, it should be mentioned that there also exist logics for reasoning *about* probability (11,12). In these systems, it is possible to infer a probability statement; for example,

$$p(G) \in [.4, .95]$$

from a given collection of probability statements. However, the inferred probability statement must also be given; it is not generated by the rules of inference.

Other systems for reasoning with and about probability include Bundy's Incidence Calculus (13) and Quinlan's Inferno (14). The Incidence Calculus, although developed independently, is quite similar to Nilsson's probabilistic logic. This similarity, unfortunately, includes intractability, in the form of a "legal assignment finder" step. The Inferno system calculates probability intervals by constraint propagation over a network whose nodes are statements and whose links are relations among (not necessarily pairs of) statements. In this respect, the Inferno system resembles systems for constructing and reasoning with Bayesian and Markov networks (15).

BIBLIOGRAPHY

1. N. Nilsson, Probabilistic logic, *Artif. Intell.*, **28** (1): 71–87, 1986.
2. J. A. Robinson, *Logic: Form and Function*, New York: North-Holland, 1979.
3. P. Gardenfors, N.-E. Sahlins (eds.), *Decision, Probability, and Utility*, Cambridge, UK: Cambridge Univ. Press, 1988.
4. R. Clemen, *Making Hard Decisions*, Belmont, CA: Wadsworth, 1991.
5. P. Cheeseman, A method of computing generalized Bayesian probability values for expert systems, *Proc. Int. Joint Conf. Artif. Intell.*, 1983, pp. 198–202.
6. T. Kane, Enhancing the inference mechanism of Nilsson's probabilistic logic, *Int. J. Intell. Syst.*, **5**: 487–504, 1990.
7. M. Deutsch-McLeish, An investigation of the general solution to entailment in probabilistic logic, *Int. J. Intell. Syst.*, **5**: 477–486, 1990.
8. P. Snow, Compressed constraints in probabilistic logic and their revision, *Proc. 7th Conf. Uncertainty Artif. Intell.*, 1991, pp. 386–391.
9. A. Frisch, P. Haddawy, Anytime deduction for probabilistic logic, *Artif. Intell.*, **69** (1–2): 93–122, 1994.
10. M. Pittarelli, Anytime decision making with imprecise probabilities, *Proc. 10th Conf. Uncertainty Artif. Intell.*, 1994, pp. 470–477.
11. F. Bacchus, *Representing and Reasoning with Probabilistic Knowledge*, Cambridge, MA: MIT Press, 1990.
12. J. Halpern, An analysis of first-order logics of probability, *Proc. Int. Joint Conf. Artif. Intell.*, 1989, pp. 1375–1381.
13. A. Bundy, Incidence calculus: A mechanism for probabilistic reasoning, *J. Autom. Reasoning*, **1**: 263–283, 1985.
14. J. R. Quinlan, Inferno: A cautious approach to uncertain inference, *Comput. J.*, **26** (3): 255–267, 1983.
15. J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, San Francisco: Morgan Kaufmann, 1988.

MICHAEL PITTARELLI
SUNY Institute of Technology

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Probability
Standard Article
Kristine L. Bell[1]
[1]George Mason University, Fairfax, VA
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2445
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (368K)

Abstract

The sections in this article are

Basic Probability

Discrete Random Variables and Distributions

Continuous Random Variables and Distributions

Expectation

Sums of Independent Random Variables

Special Discrete Distributions

Special Continuous Distributions

Limit Theorems

Summary

| | |

Copyright © 1999-2008 All Rights Reserved.

PROBABILITY

Probability theory is a branch of mathematics that deals with randomness and laws of chance. Probability theory is concerned with determining the likelihood of random events, and with characterizing their average or expected behavior. The most fundamental concept of probability theory is the likelihood, or *probability*, of an event. The probability of an event is a number between zero and one, inclusive. Probabilities near one indicate that an event is common or very likely to occur. Probabilities near zero indicate that an event is rare or not very likely to occur. A probability of .5 indicates that it is equally likely for an event to occur or not. Probabilities are often described in percentages (i.e., if an event has a 70% chance of occurring, its probability is .7).

The mathematical foundations of probability theory were developed in the 17th century during a correspondence between Pierre de Fermat and Blaise Pascal about games of chance. This work was expanded in 1713 by Jacques Bernoulli, who derived many early results in combinatorics and games involving *Bernoulli trials* (many repetitions of a procedure with two possible results, such as tossing a coin). In the 18th century, DeMoivre, Laplace, and Gauss developed the normal (or Gaussian) bell-shaped probability distribution to model various physical phenomena. From that time, probability theory was incorporated into many fields at a rapid rate. Today, probability theory has widespread applicability in science and engineering, medicine, social sciences, economics, and actuarial science, as well as many aspects of our everyday lives.

This article begins with an introduction to basic probability principles using games of chance as examples. In games of chance, the results do not depend on the skills of the players but rather on random events such as tossing coins or dice and drawing balls out of a box containing many colored balls. These games are easy to understand and also serve as models for many real-world phenomena. For example, repeated coin tosses model the bits in a binary sequence stored on a disk or transmitted on a digital communications channel.

Modeling more complex phenomena requires the concepts of random variables and probability distributions as well as mathematical techniques from calculus. The remainder of this article covers probability theory at this level. Topics include discrete and continuous random variables, probability distributions, expectation, sums of independent random variables, and limit theorems. There are numerous textbooks devoted to this material. Some representative texts are Refs. 1–6.

Probability theory is the basis for the theory of random, or stochastic, processes. The theory of *stochastic processes* is fundamental to many fields of electrical engineering dealing with signals, including communication theory, signal processing, and control theory. Many textbooks on stochastic processes also include introductory chapters on probability theory geared toward electrical engineers. Some examples are Refs. 7-13. More complex phenomena require advanced treatments of probability and the use of advanced mathematics including linear algebra, real and complex analysis, and measure theory. Some textbooks at this level include Refs. 14–19. Historical details about the development of probability theory can be found in many of the previously referenced texts, especially Ref. 1, and in textbooks on the history of mathematics including Refs. 20 and 21.

BASIC PROBABILITY

This section on basic probability introduces the concepts of random experiments, sample spaces, and events. This framework allows us to describe mathematically our intuitive notions of probability and to develop the concepts of conditional probability, independence, and expectation.

Sample Spaces and Events

The set of all possible results, or *outcomes,* in a game, or *random experiment,* is called the *sample space* and denoted by S. An *event* is a subset of the sample space that contains the desired outcomes. If an experiment has n equally likely outcomes, and f of them are desired, then the probability of the event of interest is f/n. For example, suppose that a random experiment consists of tossing a die. The sample space S contains $n = 6$ equally likely outcomes or *sample points,*

$$S = \{1, 2, 3, 4, 5, 6\}$$

Let A denote the event that an even number appears

$$A = \{2, 4, 6\}$$

and B denote the event that the result is at least three

$$B = \{3, 4, 5, 6\}$$

The event A has 3 sample points; therefore, the probability of A, denoted by $P(A)$, is $\frac{3}{6}$. The event B has four sample points and probability $P(B) = \frac{4}{6}$.

Additional events can be defined in terms of the events A and B using set operations. The *complement* of an event, denoted by A^c consists of all the points in the sample space that are not in the event A. In this example, the complement of A is the set of odd outcomes

$$A^c = \{1, 3, 5\}$$

The *union* of A and B, denoted by $A \cup B$, is the event C, which contains all the sample points in either A or B,

$$C = A \cup B = \{2, 3, 4, 5, 6\}$$

The *intersection* of A and B, denoted by $A \cap B$, is the event D, which contains the sample points that are in both A and B,

$$D = A \cap B = \{4, 6\}$$

The Venn diagrams in Fig. 1 illustrate these relationships.

Two events are *mutually exclusive* if they have no sample points in common (i.e., their intersection is the null set $\emptyset$). In this example, the events D and A^c are mutually exclusive, but A and B are not. The Venn diagram in Fig. 2(a) depicts two mutually exclusive events. A *partition* of an

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 2007 John Wiley & Sons, Inc.

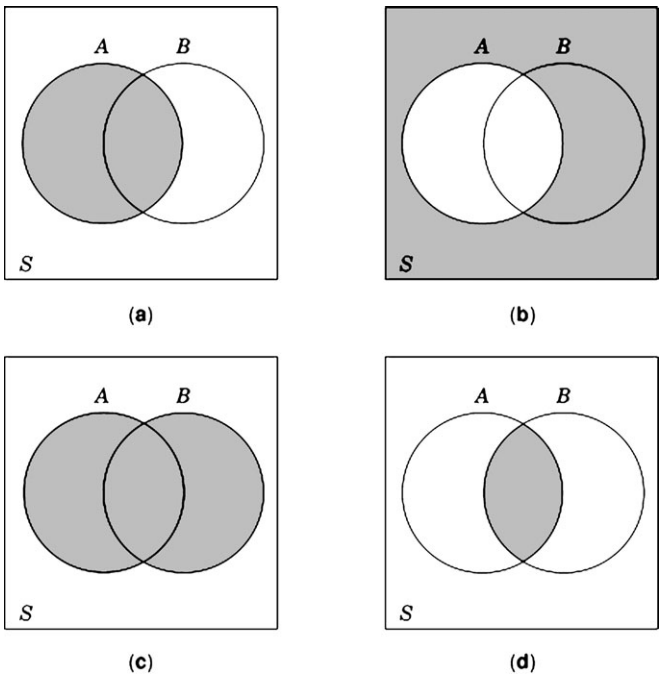


Figure 1. Venn diagrams illustrating set relationships. Shaded regions indicate the event of interest: (a) the event A, (b) the complement of A, (c) the union of A and B, and (d) the intersection of A and B.

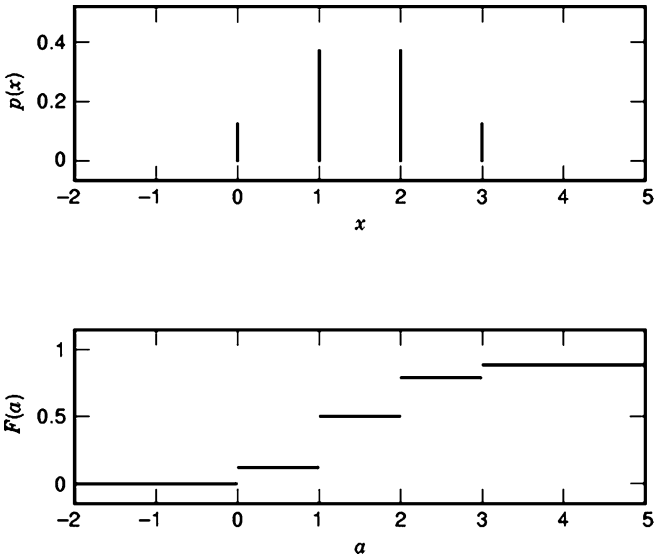


Figure 2. Venn diagrams diagrams illustrating (a) mutually exclusive events and (b) partitioning of A.

event is illustrated in Fig. 2(b). The partition of the event A consists of n smaller events, $A_1, A_2, \ldots, A_n$, which are mutually exclusive and whose union is A (i.e., $A = A_1 \cup A_2 \cup \cdots \cup A_n$). For example, the events $A_1 = \{1\}, A_2 = \{3\}, A_3 = \{5\}$ partition $A = \{1, 3, 5\}$.

Probability Axioms

The basic principles of probability theory can be stated mathematically in terms of sample spaces and events. These are referred to as *probability axioms.*

A1. $P(S) = 1$.

A2. For any event A, $0 \leq P(A) \leq 1$.

A3. For two mutually exclusive events A_1 and A_2, $P(A_1 \cup A_2) = P(A_1) + P(A_2)$.

The first two axioms define probabilities to be numbers between zero and one, with one being the total probability of all the possible outcomes. The third axiom says that if two events cannot occur simultaneously, the probability that either occurs is the sum of their individual probabilities.

Based on these axioms, we can derive additional useful probability rules, such as:

R1. $P(A^c) = 1 - P(A)$.

R2. $P(\emptyset) = 0$.

R3. For any two events A and B, $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

The first rule says that if an event has a certain portion of the total probability, its complement has the remaining portion. For example $P(A^c) = 1 - P(A) = ½$. The second rule says that the empty set has no probability. The last rule defines the probability of the union of two events that are not mutually exclusive. It is easily verified from the Venn diagrams in Fig. 1. If A and B are not mutually exclusive, then the outcomes common to A and B are counted twice in $P(A) + P(B)$. The probability of $A \cap B$ must be subtracted from $P(A) + P(B)$ to get $P(A \cup B)$. For example $P(A) = \frac{3}{6}$, $P(B) = \frac{4}{6}$, $P(A \cap B) = \frac{2}{6}$, and $P(A \cup B) = \% = \frac{3}{6} + \frac{4}{6} - \frac{2}{6}$.

Counting Techniques

In more complicated games involving multiple coins and dice or colored balls in a box, listing all the possible outcomes can be difficult. For example, suppose four balls are drawn from a box containing seven balls colored red (R), orange (O), yellow (Y), green (G), blue (B), purple (P), and white (W). The balls are drawn one at a time and not put back. The outcome of the four draws is the sequence of colors (e.g., $RWYO$ or $GYBW$). We wish to determine the probability that the sequence begins with a red ball followed by a blue ball and contains two additional balls of any color. Listing all the desired outcomes, as well as the possible outcomes in S will be very tedious. Instead, combinatorial analysis can be used to determine the number of outcomes in each set without having to list them. The basic principle is that the total number of outcomes of an experiment consisting of several sequential steps is the product of the number of outcomes at each step. This is known as the *multiplication rule.* In this case, there are seven possible outcomes on the first draw. On the second draw there are only six possibilities because one ball has been removed, on the third draw there are five possibilities, and so on. Thus there are $(7)(6)(5)(4) = 840$ sequences in S. This is an example of a *permutation,* or ordered arrangement, of four objects taken from a group of seven. In general, the number of permutations of r objects taken from a group of n is

$$P_{n,r} = n(n-1)(n-2)\cdots(n-r+1) = \frac{n!}{(n-r)!}$$

For sequences beginning with a red and blue ball, there is only one desired outcome on the first draw (R) and one desired outcome on the second draw (B). On the third draw there are five balls in the box; therefore, there are five possible outcomes, and on the fourth draw there are four possible outcomes. Thus there are a total of $(1)(1)(5)(4) = 20$ desired outcomes. The probability is then $20/840 = \frac{1}{41}$.

In other problems, the order of objects is not important. The number of distinct unordered sets, or *combinations,* of r items taken from a group of n is given by

$$C_{n,r} = \binom{n}{r} = \frac{n!}{(n-r)!(r)!}$$

Consider tossing a coin five times. On each toss, there are two possibilities, H or T. The outcome of five tosses is the sequence of heads and tails (e.g., *HTHHT*). We wish to determine the probability of obtaining exactly three heads. The total number of outcomes in the sample space is $(2)(2)(2)(2)(2) = 32$. We are not concerned with the order of the heads and tails; therefore, the number of sequences containing exactly $r = 3$ heads out of the $n = 5$ tosses is $5!/(2!3!) = 120/[(2)(6)] = 10$, and the probability of getting exactly three heads in five tosses is $\frac{10}{32}$.

Conditional Probability

Consider the experiment in which a die is tossed two times. The outcome of the two tosses is a pair of numbers [e.g., (1,2) or (3,6)]. There are six possible outcomes on each toss; therefore, the sample space consists of $(6)(6) = 36$ equally likely outcomes. Suppose that a one is obtained on the first toss and that we wish to determine the probability that the sum of the two tosses will be less than or equal to five. This is an example of a *conditional probability.*

Let $P(B|A)$ denote the conditional probability of the event B given that the event A has occurred. The conditional probability is found from

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \qquad (1)$$

provided $P(A) > 0$.

In this example, we wish to determine the probability that the sum is less than or equal to five given that the first toss is a one; therefore, A is the event that there is a one in the first position,

$$A = \{(1,1), (1,2), (1,3), (1,4), (1,5), (1,6)\}$$

and B is the event that the sum is less than or equal to five

$$B = \{(1,1), (1,2), (1,3), (1,4), (2,1), (2,2), (2,3), (3,1), (3,2), (4,1)\}$$

Examining the sample points in A, the sum is less than or equal to five for the first four sample points, which are the intersection of A and B,

$$A \cap B = \{(1,1), (1,2), (1,3), (1,4)\}$$

therefore $P(B|A)$ must be $\frac{4}{6}$. We can verify this using the definition in Eq. (1). The required probabilities are $P(A \cap B) = \frac{4}{36}$ and $P(A) = \frac{6}{36}$. As expected, $P(B|A) = (4/36)/(6/36) = \frac{4}{6}$.

We can also reverse the events and find the probability of event A given that event B has occurred from

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \qquad (2)$$

provided $P(B) > 0$. In this problem $P(B) = \frac{10}{36}$; therefore, the probability that the first number is a one given that the sum is less than or equal to five is $P(A|B) = (4/36)/(10/36) = \frac{4}{10}$.

The conditional probability formulas in Eqs. (1) and (2) can be rearranged to obtain the *multiplication rules for conditional probability*

$$P(A \cap B) = P(A)P(B|A) \qquad (3)$$

$$P(A \cap B) = P(B)P(A|B) \qquad (4)$$

These relationships are useful in determining probabilities in experiments in which the outcome of a sequence of procedures depends on the previous outcomes. For example, suppose that three boxes contain red and blue balls. Box 1 has two red and two blue balls, Box 2 has one red and two blue balls, and Box 3 has four red and one blue ball. A box is selected at random, and a ball is drawn from the box. What is the probability that a blue ball will be drawn from Box 1? Let A denote selecting Box 1 and B denote drawing a blue ball. Selecting a blue ball from Box 1 is the intersection of events A and B. The probability of selecting Box 1 is $P(A) = \frac{1}{3}$, and the probability of drawing a blue ball given that Box 1 was chosen is $P(B|A) = \frac{2}{4}$; therefore, the probability of selecting a blue ball from Box 1 is $P(A \cap B) = (\frac{1}{3})(\frac{2}{4}) = \frac{1}{6}$.

Suppose that we are also interested in finding the probability of drawing a blue ball from any box. We can find this probability by combining the multiplication rule in Eq. (3) with Axiom 3. Let A_1, A_2, and A_3 denote selecting Boxes 1, 2, and 3, respectively. The events $A_1 \cap B$, $A_2 \cap B$, and $A_3 \cap B$ represent drawing a blue ball from each of the three boxes. They are mutually exclusive events that partition B. From Axiom 3, the probability of drawing a blue ball is the sum of the probabilities of drawing a blue ball from each of the boxes

$$P(B) = P(A_1 \cap B) + P(A_2 \cap B) + P(A_3 \cap B)$$

The probability $P(A_1 \cap B) = P(A_1)P(B|A_1)$ has already been found to be $\frac{1}{6}$. Similarly, $P(A_2 \cap B) = P(A_2)P(B|A_2) = (\frac{1}{3})(\frac{2}{3}) = \frac{2}{9}$, and $P(A_3 \cap B) = P(A_3)P(B|A_3) = (\frac{1}{3})(\frac{1}{5}) = \frac{1}{15}$. The total probability of drawing a blue ball is $P(B) = \frac{1}{6} + \frac{2}{9} + \frac{1}{15} = \frac{41}{90}$. This is an example of the *rule of total probability.* In general, if $A_1, A_2, \ldots, A_n$ are mutually exclusive events that partition the sample space S, then for any event B,

$$P(B) = \sum_{i=1}^{n} P(A_i)P(B|A_i)$$
$$= P(A_1)P(B|A_1) + P(A_2)P(B|A_2) + \cdots + P(A_n)P(B|A_n) \qquad (5)$$

Now suppose that we are told that a blue ball has been drawn and wish to determine the probability that it came from the Box 1 [i.e., $P(A_1|B)$]. Using the definition of conditional probability in Eq. (2), the multiplication rule in Eq. (3), and the rule of total probability in Eq. (5), we have the following result, which is known as *Bayes Rule:*

$$P(A_1|B) = \frac{P(A_1 \cap B)}{P(B)} = \frac{P(A_1)P(B|A_1)}{\sum_{i=1}^{n} P(A_i)P(B|A_i)} \qquad (6)$$

For this example,

$$P(A_1|B) = \frac{(1/3)(2/4)}{(41/90)} = \frac{15}{41}$$

Independence

In some experiments, knowledge about one event tells us nothing about the probability of another event. For example, if a coin is tossed twice, the probability of obtaining a six on the second toss is ⅙, regardless of the outcome of the first toss. The outcomes on the two tosses are said to be *independent.* In general, two events A and B are independent if and only if

$$P(A \cap B) = P(A)P(B) \qquad (7)$$

For this example, let A denote obtaining a one on the first toss and B denote obtaining a six on the second toss. $P(A) = ⅙$, $P(B) = ⅙$, and $P(A \cap B) = \frac{1}{36}$. We see that Eq. (6) is satisfied. Now let C denote the event that the sum is less than or equal to five. From our previous example, $P(C) = \frac{10}{36}$, and $P(A \cap C) = \frac{4}{36}$. In this case, $P(A \cap C) \neq P(A)P(C)$; therefore, A and C are not independent events.

Expectation

Probability theory also deals with expected or long-term average behavior of a random experiment. For example, suppose that a person plays a game in which he or she pays \$1.00 to toss a die. The player wins \$3.00 if a 6 is thrown, \$1.50 if a 5 is thrown, and nothing if a 1, 2, 3, or 4 is thrown. On the average, how much can the player expect to win or lose? Subtracting the cost to play, net winnings are \$2.00 when a 6 is thrown, \$0.50 when a 5 is thrown, and −\$1.00 when a 1, 2, 3, or 4 is thrown. On each toss, the player will win \$2.00 with probability ⅙, \$0.50 with probability ⅙, and −\$1.00 with probability $\frac{4}{6}$. The average winnings are $2(⅙) + 0.5(⅙) - 1(\frac{4}{6}) = -1.5(⅙) = -0.25$. The expected loss per game is 25 cents. Although the player cannot actually lose 25 cents on a given game, this means that if the game is played many times, the player will lose about 25 cents per game on the average.

Summary

These basic concepts form the foundation of probability theory. The examples were based on random experiments in which the sample spaces consisted of a finite number of equally likely events. These experiments serve as models for a many phenomena arising in a variety of applications; however, there are many more phenomena for which these techniques are inadequate. To develop more sophisticated tools for solving more complex problems, we need the concepts of *random variables* and *probability distributions* and mathematical techniques from calculus.

DISCRETE RANDOM VARIABLES AND DISTRIBUTIONS

A random variable is a function that maps the sample points of an experiment onto the real line. For example, suppose that an experiment consists of tossing a coin three times. The sample space contains eight equally likely outcomes

$$S = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$$

If we let X denote the number of heads, then X is a random variable that maps the eight sample points into the numbers 0, 1, 2, and 3. In this case, X is a *discrete random variable* because it can only have one of a discrete set of values.

Probability Mass Function

For discrete random variables, the *probability mass function* (*pmf*) describes how probability is distributed among the possible values of the random variable. The pmf of X is denoted by $p(x)$ and is defined as

$$p(x) = P(X = x) \qquad (8)$$

Probability mass functions have two properties that follow from the probability axioms:

1. $0 \leq p(x) \leq 1$
2. $\Sigma_x\, p(x) = 1$.

Because the pmf is a collection of probabilities, its values must be between zero and one, inclusive. Furthermore, the pmf assigns probability to all the possible values X; therefore, it must sum to one.

In the coin toss problem, the pmf of X is

$$p(0) = P(X = 0) = P(TTT) = 1/8$$
$$p(1) = P(X = 1) = P(TTH) + P(THT) + P(HTT) = 3/8$$
$$p(2) = P(X = 2) = P(THH) + P(HTH) + P(HHT) = 3/8$$
$$p(3) = P(X = 3) = P(HHH) = 1/8$$

It is easy to verify that the properties are satisfied.

Cumulative Distribution Function

The *cumulative distribution function* (*CDF*) also characterizes the probability distribution. The CDF of X is defined for all real numbers a by

$$F(a) = P(X \leq a) = \sum_{x \leq a} p(x) \qquad (9)$$

Some properties of the CDF follow:

1. $\lim_{\alpha \to -\infty} F(a) = 0$.
2. $\lim_{\alpha \to -\infty} F(a) = 1$.
3. $F(a)$ is nondecreasing [i.e., if $a < b$, then $F(a) \leq F(b)$].

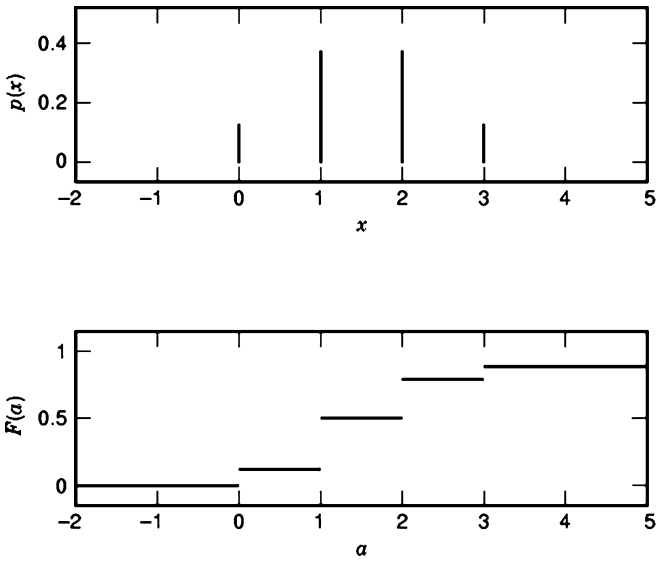


Figure 3. The probability mass function $p(x)$ and cumulative distribution function $F(a)$ of the discrete random variable X.

The CDF for X in the coin toss example is

$$F(a) = \begin{cases} 0 & a < 0 \\ 1/8 & 0 \le a < 1 \\ 1/2 & 1 \le a < 2 \\ 7/8 & 2 \le a < 3 \\ 1 & a \ge 3 \end{cases}$$

For discrete random variables, $F(a)$ has discontinuities or jumps located at the discrete values of the random variable, and the size of a jump is equal to the probability that X is equal to that value. Thus, given the CDF, the pmf can be obtained by subtracting the value of the CDF at the right side of the discontinuity from the value at the left side of the discontinuity. The pmf and CDF for the coin-tossing experiment are shown in Fig. 3.

Probability associated with X can be found from both the pmf and the CDF as follows:

$$P(a < X \le b) = \sum_{a < x \le b} p(x) \qquad (10)$$

$$= \sum_{x \le b} p(x) - \sum_{x \le a} p(x)$$

$$= F(b) - F(a) \qquad (11)$$

For example, the probability that the number of heads is one or two is $P(0 < X \le 2) = p(1) + p(2) = \frac{3}{8} + \frac{3}{8} = \frac{6}{8}$, or $P(0 < X \le 2) = F(2) - F(0) = \frac{7}{8} - \frac{1}{8} = \frac{6}{8}$.

Joint Distribution Functions

The *joint distribution* of two discrete random variables is characterized by the joint pmf and joint CDF. The joint pmf, denoted by $p(x, y)$ assigns probability to all possible joint outcomes

$$p(x, y) = P(X = x, Y = y) \equiv P(X = x \cap Y = y) \qquad (12)$$

Similar to the single random variable, or *univariate,* pmf, the joint, or *bivariate,* pmf has the following properties:

Table 1. Joint and Marginal Probability Mass Functions

		y			
	$p(x, y)$	0	1	2	$p_X(x)$
	0	.49	.14	.01	.64
x	1	.28	.04	0	.32
	2	.04	0	0	.04
	$p_Y(y)$	.81	.18	.01	1.00

1. $0 \le p(x, y) \le 1$.
2. $\Sigma_{x,y}\, p(x, y) = 1$.

The joint CDF is defined for all real numbers a and b by

$$F(a, b) = P(X \le a, Y \le b) = \sum_{x \le a, y \le b} p(x, y) \qquad (13)$$

For example, suppose that items produced by an assembly line are tested for defects. Past experience indicates that 70% of the parts have no defects, 20% have minor defects that can be corrected, and 10% have major defects and must be discarded. Suppose that two items are tested. Let X denote the number of items with minor defects and Y denote the number of items with major defects. Then $X \in \{0, 1, 2\}$ and $Y \in \{0, 1, 2\}$. The joint pmf can be found using combinatorial techniques and is summarized in Table 1. Entries for impossible events, such as $(X = 2 \cap Y = 2)$, have zero probability. The probability that X and Y are in some subset A of the possible values can be found from the joint pmf using

$$P(X, Y \in A) = \sum_{x, y \in A} p(x, y) \qquad (14)$$

The probability of at least one minor defect and no major defects is, therefore, $P(X \ge 1, Y = 0) = p(1, 0) + p(2, 0) = .28 + .04 = .32$, and the probability of one major defect is $P(Y = 1) = p(0, 1) + p(1, 1) + p(2, 1) = .14 + .04 + 0 = .18$.

Marginal Distributions

The *marginal pmf*s for X alone and Y alone are found from

$$p_X(x) = P(X = x) = \sum_y p(x, y) \qquad (15)$$

$$p_Y(y = P(Y = y) = \sum_x p(x, y) \qquad (16)$$

The marginal pmfs are also shown in Table 1.

Conditional Distributions

The *conditional pmf* of X given $Y = y$ is defined as

$$p(x|Y = y) = P(X = x|Y = y)$$

$$= \frac{P(X = x, Y = y)}{P(Y = y)}$$

$$= \frac{p(x, y)}{p_Y(y)} \qquad \text{provided } p_Y(y) > 0 \qquad (17)$$

In this example, the pmf of X given $Y = 1$ is

$$p(0|Y=1) = \frac{p(0,1)}{p_Y(1)} = \frac{.14}{.18} = .7778$$
$$p(1|Y=1) = \frac{p(1,1)}{p_Y(1)} = \frac{.04}{.18} = .2222$$
$$p(2|Y=1) = \frac{p(2,1)}{p_Y((1)} = \frac{0}{.18} = 0$$

The conditional pmf of Y given $X = x$ is defined similarly,

$$p(y|X=x) = \frac{p(x,y)}{p_X(x)} \qquad \text{provided } p_X(x) > 0 \qquad (18)$$

Independence

Two discrete random variables are said to be independent if and only if

$$p(x,y) = p_X(x)p_Y(y) \qquad \text{for all } x \text{ and } y \qquad (19)$$

In this example, $p(0, 0) = .49 \neq p_X(0)p_Y(0) = (.64)(.81) = .5184$; therefore, X and Y are not independent.

Transformations

Suppose now that we are interested in the total number of defective items. We can define a new random variable $Z = X + Y$. Then $Z \in \{0, 1, 2\}$. The pmf for Z can be determined as follows

$$p_Z(0) = P(X+Y=0) = p(0,0) = .49$$
$$p_Z(1) = P(X+Y=1) = p(0,1) + P(1,0) = .14 + .28 = .42$$
$$p_Z(2) = P(X+Y=2) = p(0,2) + p(1,1) + p(2,0)$$
$$= 0.1 + .04 + .04 = .09$$

This is an example of a *transformation* of two random variables into a new random variable.

CONTINUOUS RANDOM VARIABLES AND DISTRIBUTIONS

Consider an experiment that consists of monitoring the length of time it takes to serve a customer in a check-out line. The sample space consists of all positive real numbers

$$S = \{y : y > 0\}$$

Let Y denote the length of time it takes to check out a customer. Here Y is a *continuous random variable* because it can have any value on a continuous range, in this case the interval $(0, \infty)$.

Probability Density Function

For continuous random variables, the *probability density function* (*pdf*) denoted by $f(y)$ characterizes the distribution of probability. The probability that Y has a value in the interval $[a, b]$ is found from

$$P(a \le Y \le b) = \int_a^b f(y)\,dy \qquad (20)$$

Thus the probability that Y is in some interval is the area under the pdf over that interval. Note that if $b = a$, then

$$P(a \le Y \le a) = P(Y = a) = \int_a^a f(y)\,dy = 0$$

The probability that Y will assume a particular value is zero; however, this does not mean that it is impossible. For continuous random variables, probability can be assigned only to intervals, not to points. This means that

$$P(a \le Y \le b) = P(a \le Y < b) = P(a < Y \le b)$$
$$= P(a < Y < b) = \int_a^b f(y)\,dy$$

The pdf has the following properties:

1. $f(y) \ge 0$.
2. $\int_{-\infty}^{\infty} f(y)dy = 1$.

The area under the pdf over any interval on the real line is a probability; therefore, the pdf cannot be negative, but it does not necessarily have to be less than one. The pdf integrates to one because the probability that Y is on the real line is one.

A possible pdf for the check-out time Y is

$$f(y) = \begin{cases} e^{-y} & y > 0 \\ 0 & y \le 0 \end{cases}$$

The probability that the check-out time is more than 3 minutes is found from

$$P(Y > 3) = \int_3^{\infty} e^{-y}\,dy = -e^{-y}\Big|_{y=3}^{y=\infty} = e^{-3} = .0498$$

Cumulative Distribution Function

The cumulative distribution function for continuous random variables has the same definition as for discrete random variables and is found from

$$F(a) = P(Y \le a) = \int_{-\infty}^{a} f(y)\,dy \qquad (21)$$

The properties of the CDF are the same as in the discrete case; the probability that Y is in some interval can again be found from

$$P(a < Y \le b) = F(b) - F(a) \qquad (22)$$

For example, the CDF for the check-out time Y is

$$F(a) = \begin{cases} 1 - e^{-a} & a > 0 \\ 0 & a \le 0 \end{cases}$$

The pdf and CDF for check-out times are shown in Fig. 4. Using the CDF, we can find $P(Y > 3) = P(3 < Y < \infty) = F(\infty) - F(3) = 1 - 1 + e^{-3} = .0498$, as expected.

$F(a)$ is a continuous function for continuous random variables, and the pdf $f(y)$ can be obtained from the CDF

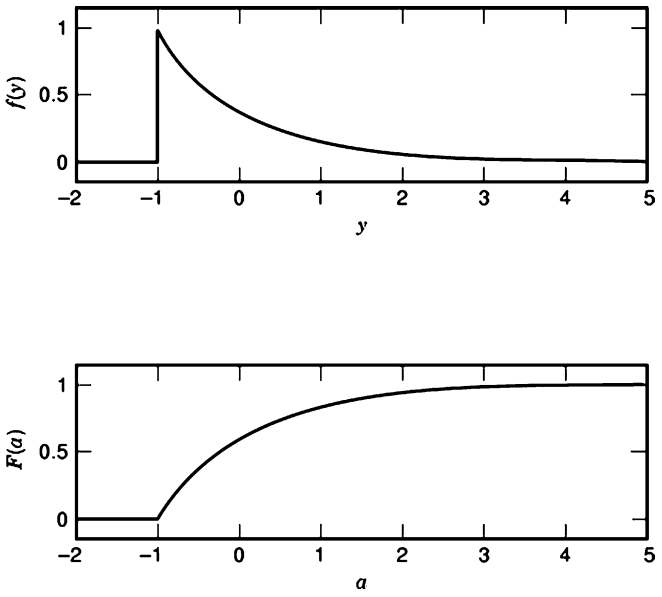


Figure 4. The probability density function $f(y)$ and cumulative distribution function $F(a)$ of the discrete random variable Y.

$F(a)$ by differentiating with respect to a and substituting y for a

$$f(y) = \frac{d}{da} F(a)\Big|_{a=y} \tag{23}$$

Joint Distribution Functions

The joint distribution of two continuous random variables is characterized by the joint pdf and joint CDF. The joint pdf, denoted by $f(x, y)$ assigns probability to regions in the xy plane,

$$P(X, Y \in A) = \iint_A f(x, y)\, dx\, dy \tag{24}$$

The probability that X and Y are within region A is the volume under the pdf over the region A. The joint pdf has the following properties:

1. $f(x, y) \geq 0$.
2. $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx\, dy = 1$.

The joint CDF for continuous random variables is defined for all real numbers a and b by

$$F(a, b) = P(X \leq a, Y \leq b) = \int_{-\infty}^{b} \int_{-\infty}^{a} f(x, y)\, dx\, dy \tag{25}$$

and the joint pdf can be found from the joint CDF from

$$f(x, y) = \frac{\partial^2}{\partial a\, \partial b} F(a, b)\Big|_{a=x, b=y} \tag{26}$$

For example, suppose that check-out times for two check-out lines have the following joint pdf:

$$f(x, y) = \begin{cases} 2e^{-(x+2y)} & x > 0, y > 0 \\ 0 & \text{otherwise} \end{cases}$$

The joint CDF is

$$P(X < a, Y < b) = \int_0^b \int_0^a 2e^{-(x+2y)}\, dx\, dy = (1 - e^{-a})(1 - e^{-2b})$$

The probability that both times are less than 2 minutes is

$$\begin{aligned} P(X < 2, Y < 2) &= \int_0^2 \int_0^2 2e^{-(x+2y)}\, dx\, dy \\ &= (1 - e^{-2})(1 - e^{-4}) = .8488 \end{aligned}$$

Marginal Distributions

Analogous to the discrete case, the marginal pdfs for X alone and Y alone are found from

$$f_X(x) = \int_{-\infty}^{\infty} f(x, y)\, dy \tag{27}$$

$$f_Y(y) = \int_{-\infty}^{\infty} f(x, y)\, dx \tag{28}$$

The marginal pdfs for our example are

$$f_X(x) = \int_0^{\infty} 2e^{-(x+2y)}\, dy = e^{-x} \qquad \text{for } x > 0$$

$$f_Y(y) = \int_0^{\infty} 2e^{-(x+2y)}\, dx = 2e^{-2y} \qquad \text{for } y > 0$$

Conditional Distributions

The conditional pdfs of X given $Y = y$ and Y given $X = x$ are defined as

$$f(x|Y = y) = \frac{f(x, y)}{f_Y(y)} \qquad \text{provided } f_Y(y) > 0 \tag{29}$$

$$f(y|X = x) = \frac{f(x, y)}{f_X(x)} \qquad \text{provided } f_X(x) > 0 \tag{30}$$

In this example, the conditional pdfs are

$$f(x|Y = y) = \frac{2e^{-(x+2y)}}{2e^{-2y}} = e^{-x} \qquad \text{for } x > 0$$

$$f(y|X = x) = \frac{2e^{-(x+2y)}}{e^{-x}} = 2e^{-2y} \qquad \text{for } y > 0$$

In this case, the conditional pdfs are the same as the marginal pdfs.

Independence

Two continuous random variables are said to be independent if and only if

$$f(x, y) = f_X(x) f_Y(y) \qquad \text{for all } x \text{ and } y \tag{31}$$

In this example, $f(x, y) = f_X(x) f_Y(y) = 2e^{-(x+2y)}$ for $x > 0$ and $y > 0$; therefore, X and Y are independent.

Transformations

Suppose that it costs the store \$5.00 to process each customer plus \$2.00 for each minute spent at check-out. Let Z be the cost to check out through the second line (i.e., $Z = 5 + 2X$). This is an example of a transformation of a continuous random variable. In general, if $Z = g(X)$ and $g(X)$ is an invertible function, then $X = g^{-1}(Z)$ and the pdf of Z is given by

$$f_Z(z) = f_X(g^{-1}(z)) \left| \frac{d}{dz} g^{-1}(z) \right| \tag{32}$$

In our example $Z = g(X) = 5 + 2X$; therefore, $X = g^{-1}(Z) = (Z - 5)/2$ and

$$f_Z(z) = 2e^{-2(z-5)/2} \left| \frac{1}{2} \right| = e^{-(z-5)} \qquad \text{for } z > 5$$

This result can be generalized to transformations of multiple random variables.

EXPECTATION

Expected Values

The statistical average or *expected value* of a random variable is denoted by $E[X]$ or μ. For discrete random variables, it is defined as

$$\mu \equiv E[X] = \sum_x x p(x) \tag{33}$$

The expected value of X is a weighted average of the possible values of the random variable, with the weighting determined by the probability of each value. In the coin tossing example, $E[X] = 0(\frac{1}{8}) + 1(\frac{3}{8}) + 2(\frac{3}{8}) + 3(\frac{1}{8}) = 1.5$.

For continuous random variables, the expected value of X is defined as

$$\mu \equiv E[X] = \int_{-\infty}^{\infty} x f(x)\, dx \tag{34}$$

In the check-out time example,

$$E[Y] = \int_0^{\infty} y e^{-y}\, dy = 1$$

The expected value of X is also called the *mean* and has the interpretation as the center of mass of the pmf or pdf.

The expected value of a function of X, say $g(X)$, is given by

$$E[g(X)] = \sum_x g(x) p(x) \qquad \text{(discrete)} \tag{35}$$

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f(x)\, dx \qquad \text{(continuous)} \tag{36}$$

In particular, if $g(X) = X^k$, the expected value is known as the *kth moment* of X. Thus the mean is the first moment. If $g(X) = (X - \mu)^k$, this is called the *kth central moment* of X. When $n = 2$, the second central moment is called the *variance* and denoted by σ^2. The variance is related to the first and second moments by

$$\sigma^2 \equiv E[(X - \mu)^2] = E[X^2] - \mu^2 \tag{37}$$

The square root of the variance (σ) is called the *standard deviation*. The variance and standard deviation are measures of the spread of the distribution from the mean. A distribution concentrated close to the mean will have a small standard deviation, and a widely spread distribution will have a large standard deviation. *Chebychev's Inequality,* which is discussed at the end of the article, provides the general rule of thumb that most of the probability is found within two to three standard deviations from the mean.

In the coin tossing example, $E[X^2] = 0(\frac{1}{8}) + 1(\frac{3}{8}) + 4(\frac{3}{8}) + 9(\frac{1}{8}) = 3$, $\sigma^2 = 3 - (1.5)^2 = .75$ and $\sigma = \sqrt{.75} = .8660$. In the check-out time example, $\sigma^2 = 2 - (1)^2 = 1$ and $\sigma = \sqrt{1} = 1$.

Moment Generating Function

The *moment generating function* (*MGF*) of the random variable X is defined as

$$M(t) = E[e^{Xt}] \tag{38}$$

The kth moment of X can be obtained from $M(t)$ by differentiating k times with respect to t and setting $t = 0$,

$$E[X^k] = \frac{d^k}{dt^k} M(t) \bigg|_{t=0} \tag{39}$$

In finding the first two moments, we use the notation

$$M'(t) \equiv \frac{d}{dt} M(t) \tag{40}$$

$$M''(t) \equiv \frac{d^2}{dt^2} M(t) \tag{41}$$

therefore, $E[X] = M'_X(0)$ and $E[X^2] = M''_X(0)$.

In the coin-tossing example, $M_X(t) = \frac{1}{8} + (\frac{3}{8})e^t + (\frac{3}{8})e^{2t} + (\frac{1}{8})e^{3t}$, $M'_X(t) = (\frac{3}{8})e^t + (\frac{6}{8})e^{2t} + (\frac{3}{8})e^{3t}$, and $M''_X(t) = (\frac{3}{8})e^t + (\frac{12}{8})e^{2t} + (\frac{9}{8})e^{3t}$. Therefore, $M'_X(0) = 1.5 = E[X]$ and $M''_X(0) = 3 = E[X^2]$. In the check-out time example, $M_Y(t) = (1 - t)^{-1}$, $M'_Y(t) = (1 - t)^{-2}$, and $M''_Y(t) = 2(1 - t)^{-3}$. Therefore, $M'_Y(0) = 1 = E[Y]$ and $M''_Y(0) = 2 = E[Y^2]$.

Joint Expectation

Expected values for jointly distributed random variables are defined as

$$E[g(X, Y)] = \sum_{x,y} g(x, y) p(x, y) \qquad \text{(discrete)} \tag{42}$$

$$E[g(X, Y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) f(x, y)\, dx\, dy \qquad \text{(continuous)} \tag{43}$$

Joint moments are obtained when $g(X, Y) = X^k Y^m$, and joint central moments are obtained when $g(X, Y) = (X - \mu_X)^k (Y - \mu_Y)^m$. When $k = m = 1$, the joint central moment is called the *covariance* of X and Y. It is related to the joint and marginal moments by

$$\text{COV}(X, Y) \equiv E[(X - \mu_X)(Y - \mu_Y)] = E[XY] - \mu_X \mu_Y \tag{44}$$

Properties

Let $Z = aX + bY + c$, then

$$\mu_Z = a\mu_X + b\mu_Y + c \tag{45}$$

$$\sigma_Z^2 = a^2\sigma_X^2 + b^2\sigma_Y^2 + 2ab\ \mathrm{COV}(X,Y) \tag{46}$$

If the random variables X and Y are independent, then

$$E[g_1(X)g_2(Y)] = E[g_1(X)]E[g_2(Y)] \tag{47}$$

This means that $E[X, Y] = E[X]E[Y] = \mu_X\mu_Y$ and $\mathrm{COV}(X, Y) = 0$. (The converse is not necessarily true, i.e., if $\mathrm{COV}(X, Y) = 0$, X and Y are not necessarily independent.) Then the variance of Z reduces to

$$\sigma_Z^2 = a^2\sigma_X^2 + b^2\sigma_Y^2 \tag{48}$$

Another consequence of Eq. (47) is that the moment generating function of Z is

$$\begin{aligned} M_Z(t) &= E[e^{(aX+bY+c)t}] = e^{ct}E[e^{aXt}]E[e^{bYt}] \\ &= e^{ct}M_X(at)M_Y(bt) \end{aligned} \tag{49}$$

SUMS OF INDEPENDENT RANDOM VARIABLES

A random sample of size n is a sequence of random variables $X_1, X_2, \ldots, X_n$, which are *independent and identically distributed* (*i.i.d.*). Their *multivariate* joint pmf or pdf has the form

$$p(x_1, x_2, \ldots, x_n) = p(x_1)p(x_2)\cdots p(x_n) \quad \text{(discrete)} \tag{50}$$

$$f(x_1, x_2, \ldots, x_n) = f(x_1)f(x_1)\cdots f(x_n) \quad \text{(continuous)} \tag{51}$$

As a consequence, $\mathrm{COV}(X_iX_j) = 0$ for $i \neq j$.

We are often interested in sums of i.i.d. random variables. Let $Z = X_1 + X_2 + \cdots + X_n$. We have the following results:

$$\mu_Z = \mu_X + \mu_X + \cdots + \mu_X = n\mu_X \tag{52}$$

$$\sigma_Z^2 = \sigma_X^2 + \sigma_X^2 + \cdots + \sigma_X^2 = n\sigma_X^2 \tag{53}$$

$$M_Z(t) = M_X(t)M_X(t)\cdots M_X(t) = M_X(t)^n \tag{54}$$

Another case of interest is $\bar{X} = (1/n)(X_1 + X_2 + \cdots + X_n)$. This is known as the *sample mean*. The mean, variance, and MGF of $\bar{X}$ are

$$\mu_{\bar{X}} = \frac{1}{n}(\mu_X + \mu_X + \cdots + \mu_X) = \mu_X \tag{55}$$

$$\sigma_{\bar{X}}^2 = \frac{1}{n^2}(\sigma_X^2 + \sigma_X^2 + \cdots + \sigma_X^2) = \frac{\sigma_X^2}{n} \tag{56}$$

$$M_{\bar{X}}(t) = M_X(t/n)M_X(t/n)\cdots M_X(t/n) = M_X(t/n)^n \tag{57}$$

SPECIAL DISCRETE DISTRIBUTIONS

Certain discrete random variables arise frequently in modeling physical phenomena. Some special discrete random variables are described below and summarized in Table 2.

Bernoulli (p)

A Bernoulli random variable X is a discrete random variable which has two possible outcomes, 1 and 0, with probabilities p and $1-p$. An experiment whose outcome is a Bernoulli random variable is called a *Bernoulli trial*. The Bernoulli random variable is named after Swiss mathematician Jacques (Jakob, James) Bernoulli, who studied games involving many repetitions of a procedure with two possible outcomes. He derived many early results in combinatorics including the formulas for permutations and combinations as well as the binomial expansion. His work in this area was published in 1713, eight years after his death.

The Bernoulli random variable models things like data bits, operational status of equipment (on or off), test results (pass or fail), quality of manufactured items (defective or good), and so on. The result $X = 1$ is usually called a success, and the result $X = 0$ is called a failure. The pmf of X is

$$p(x) = \begin{cases} 1-p & x = 0 \\ p & x = 1 \end{cases} \tag{58}$$

The first two moments of X are

$$\begin{aligned} E[X] &= 0(1-p) + 1(p) = p \\ E[X^2] &= 0(1-p) + 1(p) = p \end{aligned}$$

Therefore, the mean and variance are

$$\begin{aligned} \mu &= p \\ \sigma^2 &= p(1-p) \end{aligned}$$

The MGF of X is

$$M_X(t) = e^{0t}(1-p) + e^{1t}p = 1 - p + pe^t$$

Taking derivatives

$$M_X'(t) = pe^t$$

$$M_X''(t) = pe^t$$

therefore, $E[X] = M_X'(0) = p$ and $E[X_2] = M_X''(0) = p$, as expected. The Bernoulli random variable is the basic building block for the Binomial, Geometric, and Negative Binomial random variables, which characterize different observations of repeated Bernoulli trials.

Binomial (n, p)

A Binomial random variable X is the number of successes obtained in n i.i.d. Bernoulli trials. The possible values for X are 0, 1, ..., n. The probability that $X = 0$ is the probability of n failures, which is $(1-p)^n$, because the trials are independent and the probabilities multiply. The probability that $X = 1$ is the probability of $n-1$ failures and 1 success, multiplied by the number of combinations of $r = 1$ successes out of n trials,

$$p(1) = \binom{n}{1} p(1-p)^{n-1}$$

In general, the probability that $X = x$ is the probability of x successes and $n-x$ failures multiplied by the number of

Table 2. Discrete Random Variables

Type	Parameters	$p(x)$	x	μ	σ^2	$M(t)$
Bernoulli	p	$p^x(1-p)^{1-x}$	0, 1	p	$p(1-p)$	$1-p+pe^t$
Binomial	n, p	$\binom{n}{x}p^x(1-p)^{n-x}$	$0, 1, \ldots, n$	np	$np(1-p)$	$(1-p-pe^t)^n$
Geometric	p	$p(1-p)^{x-1}$	$1, 2, \ldots$	$\frac{1}{p}$	$\frac{(1-p)}{p^2}$	$\frac{pe^t}{1-(1-p)e^t}$
Negative Binomial	k, p	$\binom{x-1}{k-1}p^k(1-p)^{x-k}$	$k, k+1, \ldots$	$\frac{k}{p}$	$\frac{k(1-p)}{p^2}$	$\left[\frac{pe^t}{1-(1-p)e^t}\right]^k$
Poisson	μ	$\frac{e^{-\mu}(\mu)^x}{x!}$	0, 1, 2, ...	μ	μ	$e^{\mu(e^t-1)}$

combinations of x successes in n trials. Therefore,

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x} \qquad x = 0, 1, \ldots, n \tag{59}$$

The factor

$$\binom{n}{x}$$

is called a *binomial coefficient* because it appears in the *binomial expansion* of the sum of two numbers raised to the power n:

$$(a+b)^n = \sum_{x=0}^{n} \binom{n}{x} a^x b^{n-x} \tag{60}$$

The binomial expansion can be used to show that

$$\sum_{x=0}^{n} p(x) = \sum_{x=0}^{n} \binom{n}{x} p^x (1-p)^{n-x} = (p+1-p)^n = 1$$

It is often convenient to express X as the sum of n i.i.d. Bernoulli random variables, $X_1, \ldots, X_n$. Therefore when $n=1$, a Binomial $(1, p)$ random variable is the same as a Bernoulli (p) random variable. Using the properties of sums of independent random variables, the mean and variance of X are $\mu = np$ and $\sigma^2 = np(1-p)$, and the MGF of X is $M(t) = (1-p+pe^t)^n$.

Geometric (p)

A Geometric random variable X is the number of Bernoulli trials until the first success is obtained. The possible values for X are $1, 2, \ldots$. The probability that $X=1$ is the probability that a success is obtained on the first trial, which is p. The probability that $X=2$ is the probability of a failure on the first trial and a success on the second trial, which is $(1-p)p$. In general, the probability that $X=x$ is the probability of $x-1$ failures and a success on the xth trial. Therefore,

$$p(x) = p(1-p)^{x-1} \qquad x = 1, 2, \ldots \tag{61}$$

The mean and variance of X are $\mu = 1/p$ and $\sigma^2 = 1/p^2$, and the MGF of X is $M(t) = pe^t/[1-(1-p)e^t]$.

These can be derived using the *geometric series*

$$\sum_{x=0}^{\infty} a^x = \frac{1}{1-a} \qquad |a| < 1 \tag{62}$$

Negative Binomial (k, p)

A Negative Binomial random variable X is the number of Bernoulli trials until the kth success is obtained. The possible values for X are $k, k+1, \ldots$. The probability that $X=k$ is the probability that k successes are obtained on the first k trials, which is p^k. The probability that $X=k+1$ is the probability of $k-1$ successes and one failure on the first k trials, and a success on the $(k+1)$th trial, multiplied by the number of combinations of $k-1$ successes in k trials. In general, the probability that $X=x$ is the probability of $k-1$ successes and $x-k$ failures on the first $x-1$ trials and a success on the xth trial, multiplied by the number of combinations of $k-1$ successes in $x-1$ trials. Therefore,

$$p(x) = \binom{x-1}{k-1} p^k (1-p)^{x-k} \qquad x = k, k+1, \ldots \tag{63}$$

X can be expressed as the sum of n i.i.d. Geometric random variables, $X_1, \ldots, X_n$. Using the properties of sums of independent random variables, the mean and variance of X are $\mu = k/p$ and $\sigma^2 = k/p^2$, and the MGF of X is $M(t) = \{pe^t/[1-(1-p)e^t]\}^k$. The Negative Binomial distribution gets its name because proving that the distribution sums to one requires use of the binomial expansion of $[1-(1-p)]$ raised to a negative power $(-k)$. It is also called the Pascal distribution after French mathematician Blaise Pascal.

Poisson (μ)

A Poisson random variable X is the number of successes observed in an interval of length $\mu = \lambda t$, where λ is the average rate of successes and t is an interval of observation. The interval may correspond to time, length, etc. The possible values for X are $0, 1, \ldots$, and the pdf is

$$p(x) = \frac{e^{-\mu}(\mu)^x}{x!} \qquad x = 0, 1, 2, \ldots \tag{64}$$

The mean and variance of X are $\mu = \mu$ and $\sigma^2 = \mu$, and the MGF of X is $M(t) = e^{\mu(e^t-1)}$. These can be derived using the series expansion of the exponential function,

$$\sum_{x=0}^{\infty} \frac{a^x}{x!} = e^a \tag{65}$$

The Poisson distribution gets its name from French mathematician Simeon-Denis Poisson who introduced it in 1837 as a limiting form of a binomial distribution when n be-

comes large and p becomes small while np remains constant.

SPECIAL CONTINUOUS DISTRIBUTIONS

Some special continuous random variables are described below and summarized in Table 3.

Uniform (a, b)

A uniform random variable X is a continuous random variable whose pdf is constant, or uniform, over the interval $[a, b]$. To ensure that the area under the pdf is one, the height of the pdf must be the inverse of the length of the interval $1/(b-a)$

$$f(x) = \frac{1}{b-a} \qquad a \le x \le b \tag{66}$$

The mean and variance are $\mu = (a+b)/2$ and $\sigma^2 = (b-a)^2/12$, and the MGF of X is $M(t) = (e^{tb} - e^{ta})/[t(b-a)]$. At $t = 0$, the MGF has the form $\frac{0}{0}$; however, it is easy to verify that the limit as $t \to 0$ exists and is equal to one. The derivatives of the MGF also have the form $\frac{0}{0}$, but the limits as $t \to 0$ exist and are equal to the moments.

Normal (μ, σ^2)

A normal random variable X has the pdf

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} \qquad -\infty < x < \infty \tag{67}$$

The mean and variance are μ and σ^2, and the MGF is $M_X(t) = e^{\mu t + \sigma^2 t^2/2}$. The *standard normal* distribution has $\mu = 0$ and $\sigma^2 = 1$. This distribution was first introduced by French–English mathematician Abraham DeMoivre in 1733 as an approximation to the binomial distribution. He called it the exponential bell-shaped curve. The normal distribution is also called the Gaussian distribution after German mathematician and scientist Karl Friedrich Gauss, who used it to model errors in scientific experiments in 1809. It was referred to as the normal distribution by British statistician Karl Pearson in the late 19th century, who observed that it was "normal" for data sets to have this distribution. These observations are consequences of the Central Limit Theorem, which is discussed at the end of the article. It states that the distribution of a sum of a large number of independent random variables is approximately normal. Because of this result, the normal distribution models many phenomena.

Gamma (α, β)

The gamma random variable has pdf

$$f(x) = \frac{x^{\alpha-1} e^{-x/\beta}}{\beta^\alpha \Gamma(\alpha)} \qquad x \ge 0 \tag{68}$$

where $\Gamma(\alpha)$ is the gamma function, defined as

$$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x}\, dx \tag{69}$$

For $\alpha > 1$, the gamma function has the property

$$\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1)$$

When α is a positive integer (i.e., $\alpha = n$), $\Gamma(n) = (n-1)!$. The mean and variance are of X are $\mu = \alpha\beta$ and $\sigma^2 = \alpha\beta^2$, and the MGF is $M(t) = (1 - \beta t)^{-\alpha}$. Some special cases of the Gamma distribution include the Exponential, Erlang, and Chi-Square distributions.

Exponential (λ)

The Exponential distribution is obtained when $\alpha = 1$ and $\beta = 1/\lambda$. The pdf is

$$f(x) = \lambda e^{-\lambda x} \qquad x \ge 0 \tag{70}$$

The mean and variance are $1/\lambda$ and $1/\lambda^2$, and the MGF is $M(t) = [1 - (t/\lambda)]^{-1}$. The exponential distribution models the time between successes when the number of successes has a Poisson distribution with $\mu = \lambda t$. The parameter λ is the average rate of success in both distributions.

Erlang (n, λ)

The Erlang distribution is obtained when $\alpha = n$ and $\beta = 1/\lambda$. The pdf is

$$f(x) = \frac{(\lambda x)^{n-1} \lambda e^{-\lambda x}}{(n-1)!} \qquad x \ge 0 \tag{71}$$

The mean and variance are n/λ and n/λ^2, and the MGF is $M(t) = [1 - (t/\lambda)]^{-n}$. The Erlang distribution models the time until n successes occur when the number of successes has a Poisson distribution with $\mu = \lambda t$. An Erlang random variable is the sum of n i.i.d. Exponential random variables. It is named after Danish engineer and mathematician A. K. Erlang, who studied call traffic in telephone systems.

Chi-Square (n)

The Chi-Square distribution is obtained when $\alpha = n/2$ and $\beta = 2$. The pdf is

$$f(x) = \frac{x^{\frac{n}{2}-1} e^{-x/2}}{2^{n/2}\Gamma(n/2)} \qquad x \ge 0 \tag{72}$$

The mean and variance are n and $2n$, and the MGF is $M(t) = (1 - 2t)^{-n/2}$. The Chi-Square distribution is obtained from the sum of the squares of n standard normal random variables.

Rayleigh (σ_r^2)

The pdf of the Rayleigh distribution is

$$f(x) = \frac{x e^{-x^2/2\sigma_r^2}}{\sigma_r^2} \qquad x \ge 0 \tag{73}$$

The mean and variance are $\mu = \sigma_r\sqrt{\pi/2}$ and $\sigma^2 = \sigma^2{}_r(2 - \pi/2)$, and the MGF is

$$M(t) = 1 + \sqrt{\pi}\sigma_r t e^{\sigma_r^2 t^2/2}\{1/\sqrt{2} + \operatorname{erf}(\sigma t/\sqrt{2})\} \tag{74}$$

Table 3. Continuous Random Variables

Type	Parameters	$f(x)$	x	μ	σ^2	$M(t)$
Uniform	a, b	$\frac{1}{b-a}$	[a, b]	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$	$\frac{e^{tb}-e^{ta}}{t(b-a)}$
Normal	μ, σ^2	$\frac{1}{\sqrt{2\pi}\sigma}e^{-(x-\mu)^2/2\sigma^2}$	$(-\infty, \infty)$	μ	σ^2	$e^{\mu t+\sigma^2t^2/2}$
Gamma	α, β	$\frac{x^{\alpha-1}e^{-x/\beta}}{\beta^\alpha\Gamma(\alpha)}$	$[0, \infty)$	$\alpha\beta$	$\alpha\beta^2$	$(1-\beta t)^{-\alpha}$
Exponential	λ	$\lambda e^{-\lambda x}$	$[0, \infty)$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	$(1-t/\lambda)^{-1}$
Erlang	n, λ	$\frac{(\lambda x)^{n-1}\lambda e^{-\lambda x}}{(n-1)!}$	$[0, \infty)$	$\frac{n}{\lambda}$	$\frac{n}{\lambda^2}$	$(1-t/\lambda)^{-n}$
Chi-Square	n	$\frac{x^{(n/2)-1}e^{-x/2}}{2^{n/2}\Gamma(n/2)}$	$[0, \infty)$	n	$2n$	$(1-2t)^{-n/2}$
Rayleigh	σ_r^2	$\frac{xe^{-x^2/2\sigma_r^2}}{\sigma_r^2}$	$[0, \infty)$	$\sigma_r\sqrt{\frac{\pi}{2}}$	$\sigma_r^2\left(2-\frac{\pi}{2}\right)$	Eq. (74)

where the *error function* is defined as

$$\text{erf}(z) = \sqrt{\frac{\pi}{2}}\int_0^z e^{v^2}\,dv \qquad (75)$$

The Rayleigh distribution is obtained from the square root of sum of the squares of two normal $(0, \sigma_r^2)$ random variables.

LIMIT THEOREMS

Two of the most important theorems in probability theory are the *Law of Large Numbers* and the *Central Limit Theorem.* Another important result is *Chebychev's Inequality,* upon which the Weak Law of Large Numbers is based. Here we state these theorems without giving proofs.

Chebychev's Inequality

If X is a random variable with mean μ and variance σ^2, then for any $k > 0$,

$$P(|X-\mu| \geq k\sigma) \leq \frac{1}{k^2} \qquad (76)$$

For example, let $k = 2$. This inequality says that the probability that X has a value more than two standard deviations from its mean is less than .25. The probability that X is more than three standard deviations from its mean is less than .10. This theorem has many important theoretical implications, but it also has practical ones. A good rule of thumb for both discrete and continuous random variables is that most of the probability is within a few standard deviations of the mean.

Weak Law of Large Numbers

Let $X_1, X_2, \ldots, X_n$ be i.i.d. random variables with mean μ and variance σ^2. Then for any $\epsilon > 0$

$$P(|\overline{X}-\mu| \geq \epsilon) \to 0 \text{ as } n \to \infty \qquad (77)$$

From Eqs. (52) and (53), the mean and variance of $\bar{X}$ are μ and σ^2/n. Using Chebychev's Inequality with $\epsilon = k\sigma/\sqrt{n}$ and $k^2 = \epsilon^2 n/\sigma^2$,

$$P(|\overline{X}-\mu| \geq \epsilon) = P\left(|\overline{X}-\mu| \geq k\frac{\sigma}{\sqrt{n}}\right) \leq \frac{\sigma^2}{\epsilon^2 n}$$

As $n \to \infty$, this probability goes to zero. The Law of Large Numbers ensures that the average of a set of i.i.d. random variables converges to their mean as the number of samples increases.

Central Limit Theorem

Let $X_1, X_2, \ldots, X_n$ be i.i.d. random variables with mean μ and variance σ^2. The distribution of

$$Z = \frac{X_1 + X_2 + \cdots + X_n - n\mu}{\sigma\sqrt{n}}$$

tends to the standard normal distribution as $n \to \infty$.

Applying the properties of sums of independent random variables, the mean of Z is zero and its variance is one. The Central Limit Theorem says that the distribution of Z approaches the standard normal distribution for large n, regardless of the distribution of the sample. This theorem proves what is often observed in practice, namely that the sum of a large number of i.i.d. random variables has a distribution which is approximately normal.

SUMMARY

The concepts of random experiments, sample spaces, and events provide the framework to mathematically describe the principles of probability and to develop the concepts of conditional probability, independence, and expectation. Random variables and probability distributions provide the additional tools necessary to analyze a wide range of random phenomena. In this article, we have provided an introduction to discrete and continuous random variables, probability distributions, and expectation; developed prop-

erties for sums of independent random variables; and introduced two important probability theorems, the Law of Large Numbers and the Central Limit Theorem. We also summarized some important discrete and continuous probability distributions.

BIBLIOGRAPHY

1. S. Ross, *A First Course in Probability*, 6th ed., Upper Saddle River, NJ: Prentice-Hall, 2001.
2. J. Pitman, *Probability*, New York: Springer-Verlag, 1999.
3. J. J. Higgins and S. Keller-McNulty, *Concepts in Probability and Stochastic Modeling*, Belmont, CA: Duxbury Press, 1995.
4. R. E. Walpole, R. H. Myers, and S. L. Myers, *Probability and Statistical for Engineers and Scientists*, 7th ed., Upper Saddle River, NJ: Prentice Hall, 2002.
5. P. L. Meyer, *Introductory Probability and Statistical Applications*, 2nd ed., Reading, MA: Addison-Wesley, 1970.
6. R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, 6th ed., New York: Macmillan, 2004.
7. W. B. Davenport, Jr. and W. L. Root *An Introduction to the Theory of Random Signals and Noise*, New York: IEEE Press, 1987. Originally published by McGraw-Hill in 1958.
8. E. Cinlar, *Introduction to Stochastic Processes*, Englewood, NJ: Prentice-Hall, 1974.
9. H. Stark and J. W. Woods *Probability, Random Processes, and Estimation Theory for Engineers*, 3rd ed., Englewood Cliffs, NJ: Prentice-Hall, 2001.
10. A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed., New York: McGraw-Hill, 2002.
11. C. W. Helstrom, *Probability and Stochastic Processes for Engineers*, 2nd ed., New York: Macmillan, 1991.
12. A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd ed., Reading, MA: Addison-Wesley, 1994.
13. P. Z. Peebles Jr., *Probability, Random Variables, and Random Signal Principles*, 4th ed., New York: McGraw-Hill, 2000.
14. M. Loeve, *Probability Theory*, 3rd ed., Princeton, NJ: Van Nostrand, 1963.
15. W. Feller, *An Introduction to Probability Theory and Its Application, Vol. I*, 3rd ed., New York: Wiley, 1968.
16. W. Feller, *An Introduction to Probability Theory and Its Application, Vol. II*, 2nd ed., New York: Wiley, 1971.
17. L. Breiman, *Probability*, Philadelphia: SIAM, 1992. Originally published by Addison-Wesley in 1968.
18. P. Billingsley, *Probability and Measure*, 2nd ed., New York: Wiley, 1986.
19. R. M. Gray, *Probability, Random Processes, and Ergodic Properties*, New York: Springer-Verlag, 1988.
20. L. Motz and J. H. Weaver, *The Story of Mathematics*, New York: Plenum, 1993.
21. C. Boyer and U. C. Merzbach, *A History of Mathematics*, rev., 2nd ed., New York: Wiley, 1991.

KRISTINE L. BELL
George Mason University,
Fairfax, VA

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Process Algebra
Standard Article

Rance Cleaveland[1] and Scott A. Smolka[1]
[1]State University of New York at Stony Brook, Stony Brook, NY

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2446
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (264K)

Abstract

The sections in this article are

A Calculus of Communicating Systems

Behavioral Congruences for CCS

Equational Reasoning in CCS

Refinement Orderings for CCS

Computing Behavioral Relations for Finite-State Systems

Other Process Algebras

Conclusion

Keywords: process algebra; equational reasoning; verification; verification tools; bisimulation; failures/testing relations

| | |

Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

PROCESS ALGEBRA

The term *process algebra* encompasses a collection of theories that support mathematically rigorous (*in*)*equational* reasoning about systems consisting of concurrent, interacting processes. The field grew out of a seminal book of Milner (1) in 1980 and has been an active area of research since then. Over the past decade and a half researchers have developed a number of different process-algebraic theories in order to capture different aspects of system behavior; however, each such formalism generally includes the following characteristics.

(1) A language, or *algebra,* is defined for describing systems.
(2) A *behavioral equivalence* is introduced that is intended to relate systems whose behavior is indistinguishable to an external observer.
(3) Equational rules, or *axioms,* are developed that permit proofs of equivalences between systems to be conducted in a syntax-driven manner.

Some formalisms include a *refinement ordering* in lieu of an equivalence; in this case, the theories allow one to determine if a system is "greater than or equal to" (i.e., refines) another. The literature typically refers to each theory as a process algebra; so the field of process algebra contains many process algebras.

Process algebras derive their motivation from the fact that a system design often consists of several different descriptions of the system involving different levels of detail. The behavioral equivalence or refinement relation provided by a process algebra may be used to determine whether these different descriptions conform to one another. More specifically, higher-level descriptions of system behavior may be related to lower-level ones using the equivalence or refinement ordering supplied by the algebra. These relations are typically *substitutive,* meaning that related systems may be used interchangeably inside larger system descriptions; this facilitates compositional system verification, since low-level designs of system components may be checked in isolation against their high-level designs.

This article surveys some of the main features of process algebra, and it develops along the following lines. The next section introduces CCS, the process algebra that we use throught the article to illustrate the principles we cover. The section following then introduces behavioral equivalences based on the notion of *bisimulation,* a fundamental concept due to Milner and Park. We then show how two of these equivalences may be given equational axiomatizations. The section following that introduces the failures/testing refinement relations and provides inequational axiomatizations for them for CCS. The next section shows how these relations may be computed for finite-state systems. The penultimate section surveys related work, and the final one summarizes the contents of the article.

A Calculus of Communicating Systems

This section introduces the syntax and semantics of the process algebra A Calculus of Communicating Systems (*CCS*). CCS will serve as a vehicle for illustrating the different ingredients that make up a process algebra throughout the remainder of this article. Other process algebras are briefly discussed in the next to last section.

The Syntactic Form of CCS Processes. CCS provides a small set of operators that may be used to construct system descriptions from definitions of subsystems. The basic building blocks of these descriptions, and indeed of system definitions in all existing process algebras, are *actions.* Intuitively, actions represent atomic, uninterruptible execution steps; some actions denote internal execution, and others represent potential interactions with the environment that the system may engage in.

Actions in CCS. A binary, synchronous model of process communication underlies CCS, and the structure of the set of actions reflects this design decision. Actions represent either inputs/outputs on *ports* or internal computation steps. The former are sometimes called *external,* as they require interaction from the environment in order to take place.

To formalize these intuitions, let Λ represent a countably infinite set of labels, or ports, not containing the distinguished symbol τ. Then an action in CCS has one of the following three forms.

- α, where $\alpha \in \Lambda$, represents the act of receiving a signal on port α.
- $\bar{\alpha}$, where $\alpha \in \Lambda$, represents the act of emitting a signal on port α.
- τ represents an internal computation step.

In what follows we use A_{CCS} to stand for the set of all CCS actions; that is,

$$A_{\text{CCS}} = \Lambda \cup \{\overline{\alpha} | \alpha \in \Lambda\} \cup \{\tau\}$$

We also abuse notation by defining $\bar{\alpha} = \alpha$; note that $\bar{\tau}$ is not a valid action. We refer to the actions α and $\bar{\alpha}$, where $\alpha \in \Lambda$, as *complementary,* as they represent the input and output action on the same channel. The set $A_{\text{CCS}} - \{\tau\}$ then contains the set of external, or visible, actions; the only internal action is τ.

CCS Operators. Having defined the set A_{CCS} of CCS actions, we now introduce the operators the process algebra provides for assembling actions into systems. In what follows, we assume that p, p_1, and p_2 denote CCS system descriptions that have previously been constructed, and we also assume a countably infinite set C of *process variables.* CCS then includes six different mechanisms for building systems.

- *nil* represent the terminated process that has finished execution.
- Given $a \in A_{\text{CCS}}$, the *prefixing operator* $a.$ allows an action to be "prepended" onto an existing system description. Intuitively, $a.p$ is capable first of an a and then behaves like p.
- $+$ represents a choice construct. The system $p_1 + p_2$ has the potential of behaving like either p_1 or p_2, depending on the interactions offered by the environment.
- $|$ denotes parallel composition. The system $p_1|p_2$ interleaves the execution of p_1 and p_2 while also permitting complementary actions of p_1 and p_2 to synchronize; in this case, the resulting composite action is a τ.
- If $L \subseteq A_{\text{CCS}} - \{\tau\}$ then the *restriction* operator $\backslash L$ permits actions to be localized within a system. Intuitively, $p\backslash L$ behaves like p except that it is disallowed from interacting with its environment using actions mentioned in L. Note that τ can never be restricted.

- The operator $[f]$ allows actions in a process to be *renamed.* Here f is a function from A_{CCS} to A_{CCS} that is required to satisfy the following two restrictions:

$$f(\tau) = \tau$$
$$f(\overline{a}) = \overline{f(a)}$$

  When this is the case, f is called a *renaming.* The system $p[f]$ behaves exactly like p except that f is applied to each action that p wishes to engage in.
- If $C \in C$, then C represents a valid system provided that a *defining equation* of the form $C \triangleq p$ has been given. Intuitively, C represents an "invocation" that behaves like p. This construct allows systems to be defined recursively.

In process-algebraic parlance, system descriptions built using the above operators are often referred to as *terms* or *processes.* We use P_{CCS} to represent the set of all CCS processes. As examples, consider the following, where we assume that Λ contains `send`, `recv`, `msg`, `ack`, `get`, `put`, `get_ack` and `put_ack`.

- The term $\texttt{send}.\overline{\texttt{recv}}\ .nil$ represents a system that engages in a sequence of two actions: an "input" on the *send* channel, followed by an "output" on the `recv` channel.
- Consider the definition
  $\texttt{M} \triangleq \texttt{put}.\overline{\texttt{get}}\ .M + \texttt{put_ack}.\texttt{M}\overline{\texttt{get_ack}}$
  This defines a system `M` that may be thought of as a one-place communication buffer: given a "message" on its `put` channel, it delivers it on its `get` channel, and similarly for acknowledgments. This example illustrates how, although the version of CCS considered here does not explicitly support value passing, a limited form of data exchange can be implemented by encoding values in port names. Here `M` can handle two kinds of "data": messages and acknowledgments.
- Now consider the following definitions, where M is as defined previously.
  $\texttt{S} \triangleq \texttt{send}.\overline{\texttt{msg}}\ \underline{\texttt{.ack}}.\texttt{S}$
  $\texttt{R} \triangleq \texttt{msg}.\overline{\texttt{recv}}\ \texttt{.ack}\ .\texttt{R}$
  `P` $\triangleq$ `(S[put/msg,get_ack/ack]|M|R[get/msg,put_ack/ack])\{get,put,get_ack,put_ack}`
  Prepresents the CCS term for a simple communications protocol consisting of a sender `S`, a receiver `R`, and a medium `M`, a graphical depiction of which may be found in Fig. 1. The sender repeatedly accepts "messages" on its `send` channel, outputs them on its `msg` channel, and then awaits an acknowledgment on its `ack` channel. The receiver behaves similarly: it awaits a message on its `msg` channel, delivers it on its `recv` channel, and then sends an acknowledgment via its `ack` channel. The relabeling operators are given in the form *a/b*, *c/d*, ...; intuitively, such a relabeling changes b (and its inverse) to a, d to c, and so on. Actions not mentioned are unaffected. In this example the relabelings effect the "wiring" given in the figure. The restriction operator ensures that only the sender and receive may interact directly with the medium.

The Operational Semantics of CCS Terms. In the account so far we have relied on the reader's intuition to understand the meaning of the CCS operators. To make these meanings precise, CCS and other process algebras usually include an *operational semantics* that is intended precisely to define the "execution steps" that processes may engage in. This semantics is usually specified in the form of a ternary relation, $\rightarrow$; intuitively, $p \xrightarrow{a} p'$ holds if system p is capable of engaging in action a and then behaving like p'. Process algebras such as CCS typically define $\rightarrow$ inductively using a collection of *inference rules* for each operator.

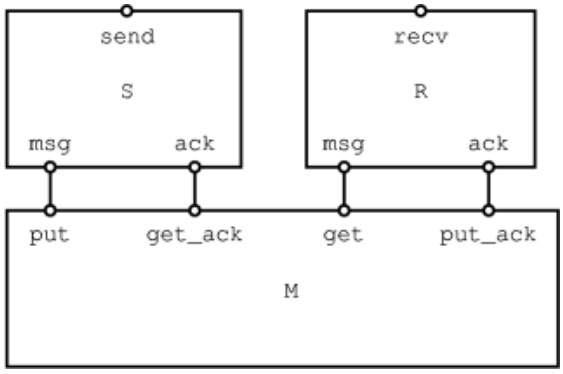


Fig. 1. The architecture of a sample communications protocol.

These rules have the following form.

$$\frac{\text{premises}}{\text{conclusion}}\ (\text{side condition})$$

A rule states that, if one has established the premises, and the side condition holds, then one may infer the conclusion. This presentation style for operational semantics is often called *SOS*, for *structural operational semantics,* and was devised by Plotkin (2).

The remainder of this section covers the SOS rules for CCS and shows how they may be used rigorously to characterize the behavior of CCS system descriptions. We group the rules on the basis of the CCS operators to which they apply.

nil. The CCS process *nil* has no rules; consequently, it is incapable of any transitions.

Prefixing. The prefixing operator contains one rule:

$$\frac{}{a.p \xrightarrow{a} p}$$

This rule has no premises, and the conclusion states that processes of the form $a.p$ may engage in a and thereafter behave like p. Note that the side condition is omitted; in such cases it is assumed to be "true".

Choice. The choice operator has two symmetric rules:

$$\frac{p \xrightarrow{a} p'}{p+q \xrightarrow{a} p'} \qquad \frac{q \xrightarrow{a} q'}{p+q \xrightarrow{a} q'}$$

These rules in essence state that a system of the form $p + q$ "inherits" the transitions of its subsystems p and q.

Parallel Composition. The parallel composition operator has three rules, the first two of which are symmetric:

$$\frac{p \xrightarrow{a} p'}{p|q \xrightarrow{a} p'|q} \qquad \frac{q \xrightarrow{a} q'}{p|q \xrightarrow{a} p|q'}$$

These rules indicate that | interleaves the transitions of its subsystems. The next rule allows processes connected by | to interact:

$$\frac{p \xrightarrow{a} p',\ q \xrightarrow{\bar{a}} q'}{p|q \xrightarrow{\tau} p'|q'}$$

According to this rule, subsystems may *synchronize* on complementary actions (i.e. inputs and outputs on the same port). Note that the action produced as the result of the synchronization is a τ; since $\bar{\tau}$ is undefined, this ensures that synchronizations involve only two partners.

Restriction. The restriction operator has one rule:

$$\frac{p \xrightarrow{a} p'}{p \setminus L \xrightarrow{a} p' \setminus L}\ (a, \bar{a} \notin L)$$

This rule, which includes a side condition, only allows actions not mentioned in L (or whose complements are not in L) to be performed by $p\setminus L$. Restriction in effect "localizes" actions in L, since the operator forbids the system's environment to interact with the system using them.

Relabeling. The relabeling operation has one rule:

$$\frac{p \xrightarrow{a} p'}{p[f] \xrightarrow{f(a)} p'[f]}$$

As the intuitive account above suggests, $p[f]$ engages in the same transitions as p, the difference being that the actions are relabeled via f.

Process Variables. The behavior of process variables is given by one rule:

$$\frac{p \xrightarrow{a} p'}{C \xrightarrow{a} p'}\ (C \triangleq p)$$

This rule states that a system C behaves like its "body," p, provided that C has been provided with a definition of the form $C \triangleq p$.

Example: As stated above, the SOS rules for CCS define the single-step transitions that CCS processes may engage in. As one example, consider the system M defined above. Using the prefixing rule, one may infer the transition

$\texttt{put}.\overline{\texttt{get}}\ .\texttt{M} \xrightarrow{\texttt{put}} \overline{\texttt{get}}\ .\texttt{M}$

Using this fact and one of the rules for +, one may therefore infer that

$\texttt{put}.\overline{\texttt{get}}\ .\texttt{M} + \texttt{put_ack}.\overline{\texttt{get_ack}}\ \ .\texttt{M} \xrightarrow{\texttt{put}} \overline{\texttt{get}}\ .\texttt{M}$

This observation and the rule for constants then permit the following transition to be inferred: $\texttt{M} \xrightarrow{\texttt{put}} \overline{\texttt{get}}\ .\texttt{M}$

Using similar lines of reasoning, one may also deduce that

$\texttt{P} \xrightarrow{\texttt{send}} ((\overline{\texttt{msg}}\ \ .\texttt{ack.S})\texttt{[put/msg,get_ack/ack]}|\texttt{M}|\texttt{R[get/msg,put_ack/ack]}\ \setminus\{\texttt{get,get_ack,put,put_ack}\}$

Note that this is the only transition available to `P`, since the transitions of `M` and `R` all involve actions in the restriction set.

CCS, Processes, and Labeled Transition Systems. The definition of $\rightarrow$ just given allows CCS processes to be viewed as state machines of a certain type. To begin with, we show how CCS may be viewed as a structure called a *labeled transition system* consisting of a collection of possible system states and transitions.

Definition 1. *A labeled transition system (LTS) is a triple $\langle Q, A, \rightarrow \rangle$, where Q is a set of states, A is a set of actions, and $\rightarrow \subseteq Q \times A \times Q$ is a transition relation.*

Some definitions of LTS also designate a start state. We refer to labeled transitions of this form (i.e. quadruples of the form $\langle Q, A, \rightarrow, q_S \rangle$ where $q_S \in Q$ is the start state) as *rooted* labeled transition systems.

Perhaps surprisingly, the definitions of this chapter show that CCS may be viewed as single LTS. Recall that P_{CCS} represents the (infinite) set of syntactically valid CCS system definitions, and let $\rightarrow_{\text{CCS}}$ be the transition relation defined in the previous subsection. Then $\langle P_{\text{CCS}}, A_{\text{CCS}}, \rightarrow_{\text{CCS}} \rangle$ satisfies the definition of LTS. This observation also holds for other process algebras and has two consequences. The first is that certain definitions, such as those for behavioral equivalences and refinement orderings, may be given in a language-independent manner by defining them with respect to LTSs. The second consequence is that that individual system descriptions may be "converted" into rooted LTSs. Mathematically, for any CCS system p the quadruple $\langle P_{\text{CCS}}, A_{\text{CCS}}, \rightarrow_{\text{CCS}}, p \rangle$ constitutes a rooted LTS. As P_{CCS} is infinite this observation is only of theoretical interest until one observes that not every state in P_{CCS} is "reachable" from p via $\rightarrow_{\text{CCS}}$. Consequently, we may instead define another LTS, M_p, consisting only of CCS terms reachable from p via sequences of transitions. If M_p contains only finitely many states, then it may be analyzed using algorithms for manipulating finite-state machines. As an example, Fig. 2 contains the finite-state rooted LTS corresponding to the communication protocol P described above.

Behavioral Congruences for CCS

Process algebras usually use a notion of behavioral congruence as a basis for system analysis. A *congruence* for an algebra is an equivalence relation (i.e. a relation that is reflexive, symmetric, and transitive) that also has the substitution property: equivalent systems may be used interchangeably inside any larger system. Formally, define a *context* $C[\ ]$ to be a system description with a "hole," $[\ ]$; given a system description p, then, $C[p]$ represents the system obtained by "filling" the hole with p. Then an equivalence $\approx$ is a congruence for a language if, whenever $p \approx q$, then $C[p] \approx C[q]$ for any context $C[\ ]$ built using operators in the language. It should be noted that relations that are congruences for some languages are not congruences for others.

In this section we study congruences for CCS with a view toward defining a relation that relates systems with respect to their "observable" behavior. In each case we first define an equivalence relation on states in an arbitrary LTS; since CCS may be viewed as an LTS, these relations may then be used to relate CCS system descriptions. We then consider the suitability of the equivalence from the standpoint of the observable behavior to which it is sensitive and study whether or not the relation is a congruence for CCS. In the first part of the section we make no special allowance for the "unobservability" of the action τ, deferring its treatment to later.

The Inadequacy of Trace Equivalence. State machines have a well-studied equivalence, *language equivalence,* that stipulates that two machines are equivalent if they accept the same sequences of symbols. Rooted labeled transition systems do not contain "accepting states" per se, and consequently the notion of language equivalence from finite-state machine theory cannot be directly applied. However, if we identify every state in a rooted LTS as being accepting, then the "language" of the machine contains the execution sequences, or *traces,* that a machine may engage in. Consequently, a reasonable first attempt at defining a

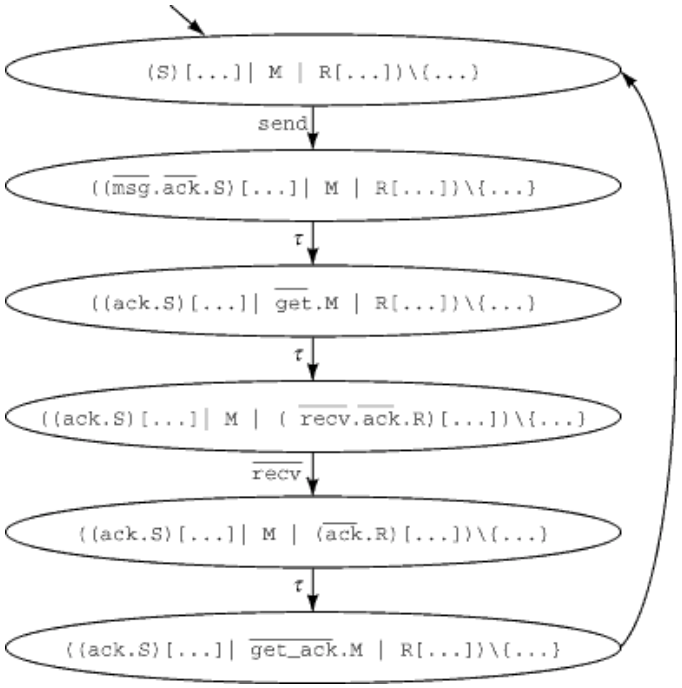


Fig. 2. The state machine for P.

behavioral equivalence for CCS and other process algebras might be to relate two system descriptions (i.e. states in the LTS $\langle Q, A, \rightarrow \rangle$) exactly when the machines for them have exactly the same traces.

Before formalizing these notions, we first review some concepts from the theory of finite sequences. If A is a set, then $A*$ consists of the set of (possibly empty) finite sequences of elements of A. We use ε to represent the empty sequence. One may now define traces, and trace equivalence, as follows.

Definition 2. *Let $\langle Q, A, \rightarrow \rangle$ be a labeled transition system.*

(1) Let $s = a_1 \ldots a_n \in A*$ be a sequence of actions. Then $q \stackrel{s}{\rightarrow} q'$ if there are states $q_0, \ldots, q_n$ such that $q = q_0$, $q_i \stackrel{a_{i+1}}{\rightarrow} q_{i+1}$, and $q' = q_n$.

(2) s is a strong trace of q if there exists q' such that $q \stackrel{s}{\rightarrow} q'$. We use $S(q)$ to represent the set of all strong traces of q.

(3) $p \approx_S q$ exactly when $S(p) = S(q)$.

We use the term *strong traces* because the definition given above does not distinguish between internal and external actions; all may appear in a strong trace. In contrast, the traditional definition of traces treats τ actions in a special manner.

Since CCS is a labeled transition system whose states are system descriptions, we may apply the definition of $\approx_S$ to CCS systems. Unfortunately, $\approx_S$ suffers from severe deficiencies for CCS and other languages that permit the definition of nondeterministic systems, as the following examples illustrate.

(1) Let p be $a.b.nil + a.c.nil$, and q be $a.(b.nil + c.nil)$. Then $p \approx_S q$, as $S(p) = S(q) = \{\varepsilon, a, ab, ac\}$. However, after an a transition q_1 can perform both a b and a c, whereas p_1 must reject one or the other of these possibilities after each of its (two) a transitions.
(2) Let $C_1 \triangleq a.C_1$ and $C_2 \triangleq a.C_2 + a.nil$. Then $C_1 \approx_S C_2$, and yet C_2 can reach a "deadlocked" state after an a-transition (i.e. a state that is incapable of any transitions) while C_1 cannot.

The trouble with trace equivalence and nondeterministic systems is that even though two systems have the same traces, they may go through inequivalent states in performing them. (This situation cannot occur in deterministic systems.) In particular, trace-equivalent systems can have different deadlocking behavior.

Bisimulation Equivalence. The last observation in the previous section suggests that an appropriate equivalence for CCS, and indeed for any language permitting the definition of nondeterministic systems, ought to have a recursive flavor: execution sequences for equivalent systems ought to "pass through" equivalent states. This intuition underlies the definition of *bisimulation,* or *strong* equivalence. The name of the equivalence stems from the fact that it is defined in terms of special relations called bisimulations.

Definition 3. *Let $\langle Q, A, \rightarrow \rangle$ be a LTS. A relation $R \subseteq Q \times Q$ is a bisimulation if, whenever $\langle p, q \rangle \in R$, the following conditions hold for any a, p', and q':*

(1) $p \xrightarrow{a} p'$ implies $q \xrightarrow{a} q'$ for some q' such that $\langle p', q' \rangle \in R$.
(2) $q \xrightarrow{a} q'$ implies $p \xrightarrow{a} p'$ for some p' such that $\langle p', q' \rangle \in R$.

Intuitively, if two systems are related by a bisimulation, then it is possible for each to simulate, or "track," the other's behavior: hence the term *bi*simulation. More specifically, for a relation to be a bisimulation, related states must be able to "match" transitions of each other by moving to related states. Two states are then *bisimulation-equivalent* exactly when a bisimulation may be found relating them.

Definition 4. *Systems p and q are bisimulation-equivalent, or bisimilar, if there exists a bisimulation R containing $\langle p, q \rangle$. We write $p \sim q$ whenever p and q are bisimilar.*

Since CCS may be viewed as a LTS description, one may use $\sim$ to relate CCS processes. As examples, we have the following.

(1) $a.b.nil + a.b.nil \sim a.b.nil$
(2) $a.b.nil + a.c.nil \nsim a.(b.nil + c.nil)$
(3) $C_1 \nsim C_2$.

Bisimulation equivalence has a number of pleasing properties. Firstly, for any labeled transition system it is indeed an equivalence; that is, the relation $\sim$ is reflexive, symmetric, and transitive. Secondly, it can be shown in a precise sense that two equivalent systems must have the same "deadlock potential"; this point is addressed in more detail below. Thirdly, $\sim$ implies $\approx_S$ and coincides with it if the LTS is *deterministic* in the sense that every state has at most one outgoing transition per action. Finally, $\sim$ is a congruence for CCS; if $p \sim q$, then p and q may be used interchangeably inside any larger system.

However, $\sim$ does suffer from a major flaw from the perspective of CCS and other process algebras allowing asynchronous execution: it is too sensitive to internal computation. In particular, the definition does not take account of the special status that τ has vis-à-vis other actions: the systems $a.\tau.b.nil$ and $a.b.nil$ are not bisimulation-equivalent, even though an external observer cannot detect the difference between them. Nevertheless, $\sim$ has been studied extensively in the literature, and for process algebras in which internal

computation in one component can indeed affect the behavior of other components, it is a reasonable basis for verification.

Deadlock, Logical Characterizations, and $\sim$. The preceding discussion states that $\sim$ relates systems on the basis of their relative "deadlock potentials." The remainder of this subsection makes this statement precise by defining a logic, called the Hennessy–Milner logic (*HML*) (3), that permits the formulation of simple system properties, including potentials for deadlock. The logic also characterizes $\sim$ in the following sense: two systems are bisimilar if and only if they satisfy exactly the same formulas in the logic.

Syntax of HML. The definition of HML is parametrized with respect to a set A of actions. Given such a set, the syntax of HML formulas can be given via the following grammar:

$$\begin{aligned}\phi ::= & \ tt \\ | & \ ff \\ | & \ \phi \wedge \phi \\ | & \ \phi \vee \phi \\ | & \ \langle a \rangle \phi \\ | & \ [a]\phi\end{aligned}$$

We use Φ for the set of all well-formed HML formulas.

The constructs in the logic may be understood as follows. First, it should be noted that formulas are intended to be interpreted with respect to states in a labeled transition system. Then tt and ff represent the constants "true" and "false" that hold of any state and no state, respectively, while $\wedge$ and $\vee$ denote conjunction ("and") and disjunction ("or"), respectively. The final two operators are referred to as *modalities,* as they permit statements to be made about the transitions emanating from a state; thus HML is a *modal logic.* A state satisfies $\langle a \rangle \phi$ if a target state of one of its a-transitions satisfies ϕ, while $[a]\phi$ holds of a state if the target states of all of its a-transitions satisfy ϕ.

Semantics of HML. In order to formalize the previous informal discussion, we first fix a labeled transition system $\mathscr{S} = \langle Q, A, \rightarrow \rangle$ having the same action set as HML. We then define a relation $\models_{\mathscr{S}} \subseteq Q \times \Phi$; intuitively, $q \models_{\mathscr{S}} \phi$ should hold if state q "satisfies" ϕ. The formal definition is given inductively as follows:

- $q \models_{\mathscr{S}} tt$ for any $q \in Q$.
- $q \models_{\mathscr{S}} ff$ for no $q \in Q$.
- $q \models_{\mathscr{S}} \phi_1 \wedge \phi_2$ if and only if $q \models_{\mathscr{S}} \phi_1$ and $q \models_{\mathscr{S}} \phi_2$.
- $q \models_{\mathscr{S}} \phi_1 \vee \phi_2$ if and only if $q \models_{\mathscr{S}} \phi_1$ or $q \models_{\mathscr{S}} \phi_2$.
- $q \models_{\mathscr{S}} \langle a \rangle \phi$ if and only if $q \stackrel{a}{\rightarrow} q'$ and $q' \models_{\mathscr{S}} \phi$ for some $q' \in Q$.
- $q \models_{\mathscr{S}} [a]\phi$ if and only if for every q' such that $q \stackrel{a}{\rightarrow} q'$, one has $q' \models_{\mathscr{S}} \phi$.

This definition includes some subtleties that deserve comment. To begin with, the formula $[a]ff$ is satisfied by any state *not* having an a-transition; such states vacuously fulfill the requirement imposed by $[a]$. Indeed, a state with no a-transitions satisfies $[a]\phi$ for any ϕ. These facts also imply that a state incapable of any action in the set $\{a_1, \ldots, a_n\}$ will satisfy the formula $[a_1]ff \wedge \cdots \wedge [a_n]ff$. If such a state occurs in an environment that requires one of these actions, then a deadlock results. In a related vein, a state satisfies $\langle b \rangle tt$ if and only if it has an b-transition; more generally, given a (nonempty) sequence of actions $b_1 \ldots b_m$, a state includes $b_1 \ldots b_m$ as one of its strong traces if and only if the state satisfies the formula $\langle b_1 \rangle \cdots \langle b_m \rangle tt$. Finally, consider a state satisfying a formula of the form

$$\langle b_1 \rangle \cdots \langle b_m \rangle ([a_1]ff \wedge \cdots \wedge [a_n]ff)$$

Such a state satisfies this formula if it can engage in the sequence $b_1 \ldots b_m$ and arrive at a state that rejects offers for interaction involving any of $a_1, \ldots, a_n$. In an environment capable of exercising the sequence $b_1 \ldots b_m$ and then requiring an interaction involving one of $a_1, \ldots, a_n$, the given state could deadlock. It is in this sense that HML permits the formulation of properties expressing potentials for deadlock.

HML and $\sim$. The relationship between HML and $\sim$ is captured by the following theorem, which states that HML characterizes $\sim$ for labeled transition systems that are *image-finite.* A LTS is image-finite if every state in the LTS has at most finitely many transitions sharing the same action label. In practice almost all labeled transition systems satisfy this requirement; in particular, CCS does, provided the definitions of process variables obey a small restriction.

Theorem 5. Let $\mathscr{L} = \langle Q, A, \rightarrow \rangle$ be an image-finite LTS, and let $p, q \in Q$. Then $p \sim q$ if and only if for all HML formulas ϕ, either $p \models_{\mathscr{L}} \phi$ and $q \models_{\mathscr{L}} \phi$ or $p \not\models_{\mathscr{L}} \phi$ and $q \not\models_{\mathscr{L}} \phi$.

On the one hand, this result and the previous discussion substantiate the claim that bisimulation equivalence requires equivalent systems to have the same "deadlock potentials." On the other hand, the theorem provides a useful mechanism for explaining why two systems fail to be equivalent: one need only present a formula satisfied by one system and not the other. The following provides examples illustrating this latter point in the context of CCS.

- Consider the system p given by $a.b.nil + a.c.nil$ and the system q given by $a.(b.nil + c.nil)$. Since $p \not\sim q$, there must be a formula satisfied by one and not the other. One such formula is $\langle a \rangle [b] ff$, which is satisfied by p but not by q.
- Consider C_1 and C_2 given above. The formula $\langle a \rangle [a] ff$ distinguishes them, as C_2 satisfies it and C_1 does not.

Observational Equivalence and Congruence for CCS. This subsection presents a coarsening of bisimulation equivalence that is intended to relax the sensitivity of the former to internal computation. The definition of this relation relies on the introduction of so-called "weak" transitions.

Definition 6. *Let* $\langle Q, A, \rightarrow \rangle$ *be a LTS with* $\tau \in A$*, and let* $q \in Q$*.*

(1) If $s \in A*$*, then* $\hat{s} \in (A - \{\tau\})*$ *is the action sequence obtained by deleting all occurrences of* τ *from* s*.*

(2) Let $s \in (A - \{\tau\})*$*. Then* $q \stackrel{s}{\Rightarrow} q'$ *if there exists* s' *such that* $q \stackrel{s'}{\rightarrow} q'$ *and* $s = \hat{s'}$*.*

Intuitively, $\hat{s}$ returns the "visible content" (i.e. non-τ elements) of the sequence s; in particular, if $a \in A$, then $\hat{a} = \varepsilon$ if $a = \tau$, while $\hat{a} = a$ if $a \neq \tau$. In addition, $q \stackrel{s}{\Rightarrow} q'$ if q can perform a sequence of transitions with the same visible content as s and evolve to q'. In this case note that the sequence of transitions that is performed is the same as s except that it potentially includes an arbitrary number of τ transitions in between the visible actions of s. In particular, $q \stackrel{\varepsilon}{\Rightarrow} q'$ if a sequence of τ transitions leads from q to q', while for a single visible action a, $q \stackrel{a}{\Rightarrow} q'$ if q can perform an a, possibly "surrounded" by some internal computation, in order to arrive at q'.

We may now define *weak bisimulations* as follows.

Definition 7. *Let* $\langle Q, A, \rightarrow \rangle$ *be a LTS, with* $\tau \in A$*. Then a relation* $R \subseteq Q \times Q$ *is a weak bisimulation if, whenever* $\langle p, q \rangle \in R$*, the following hold for all* $a \in A$ *and* $p', q' \in Q$*:*

(1) If $p \stackrel{a}{\rightarrow} p'$ *then* $q \stackrel{\hat{a}}{\Rightarrow} q'$ *for some* q' *such that* $\langle p', q' \rangle \in R$*.*

(2) If $q \stackrel{a}{\rightarrow} q'$ *then* $p \stackrel{\hat{a}}{\Rightarrow} p'$ *for some* p' *such that* $\langle p', q' \rangle \in R$*.*

States p and q are observationally equivalent, or weakly equivalent, or weakly bisimilar, if there exists a weak bisimulation R containing $\langle p, q \rangle$. *When this is the case, we write* $p \approx q$.

A weak bisimulation closely resembles a regular bisimulation; the only difference lies in the fact that systems may use weak transitions to simulate normal transitions in the other system.

As CCS is a labeled transition system whose action set contains τ, the definition of $\approx$ may be used to related CCS system descriptions. Doing so leads to the following observations:

- $a.\tau.bnil \approx a.b.nil$.
- For any p, $\tau.p \approx p$.
- Let Svc $\stackrel{\Delta}{=}$ send.$\overline{\texttt{recv}}$ .Svc. Then P $\approx$ Svc, where P is the simple communications protocol described in the previous section.

The last example illustrates the power of equivalences in relating system designs at different levels of abstraction, since Svc could be thought of as a "high-level" design that P is intended to conform to.

Even though it ignores internal computation, observational equivalence still enjoys a similar degree of deadlock sensitivity to bisimulation equivalence: a variant of HML can be defined that characterizes $\approx$ in the same way that HML characterizes $\sim$. (This logic replaces the $\langle a \rangle$ and *[a]* modalities of HML by two new operators, $\langle\langle a \rangle\rangle$ and $[[a]]$; a state $q \models_{\mathscr{L}} \langle\langle a \rangle\rangle \phi$ if there exists a q' such that $q \hat{} q'$ and $q' \models_{\mathscr{L}} \phi$, and similarly for $[[a]]$.) Consequently it would appear to be a viable candidate for relating CCS system descriptions. Unfortunately, however, it is *not* a congruence for CCS. To see why, consider the context $C[\ ]$ given by $[\ ] + b.nil$. It is easy to establish that $p \approx q$, where p is given by $\tau.a.nil$ and q by $a.nil$. However, $C[p] \not\approx C[q]$. To see this, note that $C[p] \Rightarrow a.nil$. This transition must be matched by a weak ε-labeled transition from $C[q]$. The only such transition $C[q]$ has is $C[\mathrm{q}] \rightarrow C[q]$. However, $a.nil \not\approx C[q]$, since the latter can engage in a b-labeled transition that cannot be matched by the former.

This defect of $\approx$ arises from the interplay between $+$ and the initial internal computation that a system might engage in; in particular, the only CCS operator that "breaks" the congruence-hood of $\approx$ is $+$. Some researchers reasonably suggest that this is an argument against including $+$ in the language. Milner (1,4) adopts another point of view, which we pursue in the remainder of this section, and that is to focus on finding the *largest* CCS congruence $\approx^C$ that implies $\approx$. Such a largest congruence is guaranteed to exist (3).

Definition 8. *Let* $\langle Q, A, \rightarrow \rangle$ *be a LTS with* $\tau \in A$, *and let* $p, q \in Q$. *Then* $p \approx^C q$ *if the following hold for all* $a \in A$ *and* $p', q' \in Q$*:*

(1) If $p \stackrel{a}{\rightarrow} p'$ *then* $q \stackrel{a}{\Rightarrow} q'$ *for some* q' *such that* $p' \approx q'$.
(2) If $q \stackrel{a}{\rightarrow} q'$ *then* $p \stackrel{a}{\Rightarrow} p'$ *for some* p' *such that* $p' \approx q'$.

Some remarks about this relation are in order. Firstly, it should be noted that for $p \approx^C q$ to hold, any τ-transition of p must be matched by a $\stackrel{\tau}{\Rightarrow}$ transition of q; in particular, this weak transition must consist of a *nonempty* sequence of τ transitions. Secondly, the definition is not recursive: the targets of initial matching transitions need only be related by $\approx$. Finally, it indeed turns out that $\approx^C$ is a congruence for CCS and that it is the largest CCS congruence entailing $\approx$. That is, $p \approx^C q$ implies $p \approx q$, and for any other congruence R such that $p \ R \ q$ implies $p \approx q$, $p \ R \ q$ also implies $p \approx^C q$. As examples, we have the following:

(1) $a.\tau.b.nil \approx^C a.b.nil$.
(2) $\tau.a.nil \not\approx^C a.nil$, since the $\stackrel{\tau}{\rightarrow}$ transition of the former cannot be matched by a $\stackrel{\tau}{\Rightarrow}$ transition of the latter.
(3) For any p, q, if $p \approx q$ then $\tau.p \approx^C \tau.q$.

Table 1. Axiomatizing $\sim$ for Basic CCS: Rule Set E_1

(A1)	$x + y = y + x$
(A2)	$x + (y + z) = (x + y) + z$
(A3)	$x + nil = x$
(A4)	$x + x = x$

(4) Svc $\approx^C$ P, where Svc and P are as defined above.

Equational Reasoning in CCS

In addition to definitions of behavioral congruences, process algebras traditionally provide *equational axiomatizations* that permit equivalences to be established by means of simple syntactic manipulations. This section presents such axiomatizations for CCS for both $\sim$ and $\approx^C$.

Axiomatizing $\sim$. We present the axiomatization of $\sim$ for CCS in stages by considering successively larger fragments of CCS. The first, and most basic, subset of CCS we investigate we term *basic CCS*.

Axiomatizing Basic CCS. Basic CCS contains only the *nil,* prefixing and $+$ operators of CCS, and hence it only allows the definition of "sequential" (i.e. no parallelism) terminating systems. The axiomatization of $\sim$ for basic CCS consists of the four rules given in Table 1.

Some words of explanation about these axioms are in order. Firstly, and for convenience, each rule we present has a name; in this case, the rules are named (A1)–(A4). Secondly, each rule contains variables that are intended to be arbitrary terms in the language under consideration. In (A2), for example, x, y, and z are variables, and the rule should be read as asserting that regardless of the basic CCS terms substituted for these variables, the indicated equivalence holds. Finally, axioms are used to construct equational proofs as illustrated by the following example:

$$
\begin{aligned}
&a.(b.nil + nil) + (a.nil + a.b.nil) \\
&\qquad = a.b.nil + (a.nil + a.b.nil) && \text{by } (A3) \\
&\qquad = a.b.nil + (a.b.nil + a.nil) && \text{by } (A1) \\
&\qquad = (a.b.nil + a.b.nil) + a.nil && \text{by } (A2) \\
&\qquad = a.b.nil + a.nil && \text{by } (A4)
\end{aligned}
$$

This proof establishes that $a.(b.nil + nil) + (a.nil + a.b.nil) = a.b.nil + a.nil$ in four steps, where each step represents the "application" of a rule to a subterm, yielding a new term. The development of such equational proofs typically relies on four rules of inference reflecting the fact that $=$ is reflexive, symmetric, and transitive and that equal terms may be used interchangeably; these rules implicitly support the construction of proofs such as the one above. We will not say more about this matter.

When a proof that $t_1 = t_2$ may be derived using axioms in set E, we write $E \vdash t_1 = t_2$. Thus,

$$E_1 \vdash a.(b.nil + nil) + (a.nil + a.b.nil) = a.b.nil + a.nil$$

where E_1 contains the four rules in Table 1.

Returning to the rules in Table 1, rules (A1) and (A2) assert that $+$ is commutative and associative, respectively. Rule (A3) indicates that *nil* is an *identity element* for $+$; these first three rules are sometimes

Table 2. Axiomatizing $\sim$ for Basic Parallel CCS: Rule Set E_2

(A1)–(A4) from Table 1
(Exp) $(\sum_{i\in I} a_i.x_i) \mid (\sum_{j\in J} b_j.y_j) =$ $\sum_{i\in I} a_i.(x_i \mid (\sum_{j\in J} b_j.y_j)) + \sum_{j\in J} b_j.((\sum_{i\in I} a_i.x_i) \mid y_j) + \sum_{\{(i,j)\mid a_i=\overline{b_j}\}} \tau.(x_i \mid y_j)$

refered to as the *monoid laws,* a monoid being any mathematical structure obeying these axioms. The final rule is often called the *absorption law,* as it allows multiple copies of the same summand to be "absorbed" into one.

Metatheory. Given a proposed axiomatization for an equivalence relation, one may ask two questions:

(1) Is the axiomatization *sound*? That is, are all proved equalities true?
(2) Is the axiomatization *complete*? That is, are all true equalities provable?

Soundness is an absolute necessity; an unsound proof system is worse than useless, since it allows the derivation of untrue information. Completeness is highly desirable, since once a proof system is shown to be complete, one knows that there can be no "missing" axioms.

The following results establish the soundness and completeness of the axioms in Table 1 for $\sim$ over basic CCS.

Theorem 9 (Soundness). Let t_1 and t_2 be terms in basic CCS, and suppose that $E_1 \vdash t_1 = t_2$. Then $t_1 \sim t_2$.

Theorem 10 (Completeness). Let t_1 and t_2 be terms in basic CCS such that $t_1 \sim t_2$. Then $E_1 \vdash t_1 = t_2$.

Axiomatizing Basic Parallel CCS. The next fragment of CCS we present an axiomatization for extends basic CCS with the inclusion of the parallel composition operator, $|$. In what follows we call this fragment *basic parallel CCS*.

As it turns out, rules (A1)–(A4) remain sound for basic parallel CCS, but they are obviously not complete, since none of the rules mentions $|$. In order to devise a complete axiomatization for this subset of CCS we therefore must add axioms for $|$. The new axiomatization is presented in Table 2.

The single new axiom, (Exp), is often referred to as the *expansion law,* as it shows how terms involving $|$ at the top level may be "expanded" into ones involving prefixing and summation. This axiom is the most complicated rule for CCS, and it deserves further commentary. Firstly, the Σ notation needs explanation. Rules (A1) and (A2) indicate that $+$ is commutative and associative. This means that expressions of the form $t_1 + \cdots + t_n$, while not strictly speaking expressions (since they are not fully parenthesized), nevertheless have a precise meaning, since all parenthesizations of such expressions are equivalent. More generally, given a finite index set I and an I-indexed set of terms of the form t_i, we may define $\Sigma_{i\in I}\ t_i$ as *nil* if I is empty and as the summation of all the t_i's otherwise.

The second feature of (Exp) is that it may only be applied to a term $t_1|t_2$ if both t_1 and t_2 have a special form: namely, each must be a summation of terms whose outermost operator involves prefixing. Technically speaking, (Exp) is not a single axiom but an axiom schema, with each different value of I and J yielding a different axiom.

Finally, the right-hand side of (Exp) consists of three summands, each corresponding to a different SOS rule for $|$. The first summand allows the left subterm to "move"autonomously, and the second permits the same behavior from the right subterm. The third summand handles possible synchronizations.

Table 3. Axiomatizing $\sim$ for Finite CCS: Rule Set E_3

	(A1)–(A4) from Table 1; (Exp) from Table 2
(Res1)	$nil\backslash L = nil$
(Res2)	$(a.x)\backslash L = \begin{cases} nil & \text{if } a, \overline{a} \in L \\ a.(x\backslash L) & \text{otherwise} \end{cases}$
(Res3)	$(x + y)\backslash L = x\backslash L + y\backslash L$
(Rel1)	$nil[f] = nil$
(Rel2)	$(a.x)[f] = f(a).(x[f])$
(Rel3)	$(x + y)[f] = x[f] + y[f]$

To see how (Exp) is used in equation proofs, consider the following example, showing that $E_2 \vdash nil|b.nil = b.nil$; recall that nil is the same as $\Sigma_{i\in\phi}\, t_i$:

$$\begin{aligned} nil|b.nil &= nil + b.(nil|nil) + nil && \text{by (Exp)} \\ &= b.(nil|nil) && \text{by (A3) twice} \\ &= b.(nil + nil + nil) && \text{by (Exp)} \\ &= b.nil && \text{by (A3) twice} \end{aligned}$$

Indeed, for any term t in basic parallel CCS it follows that $E_2 \vdash nil|t = t$. It may also be shown that for any terms t_1, t_2 and t_3,$E_2 \vdash t_1|t_2 = t_2|t_1$ and $E_2 \vdash t_1|(t_2|t_3) = (t_1|t_2)|t_3$; consequently, $|$ is commutative and associative. Finally, as the strict application of (Exp) results in many occurrences of nil as a summand, these nil's are suppressed in practice, since they may be removed by applying (A3) appropriately.

It may be shown that E_2 is a sound and complete axiomatization of $\sim$ for basic parallel CCS.

Axiomatizing $\sim$ for Finite CCS: Rule Set E_3. The next fragment of CCS we axiomatize includes all operators except for process variables; the literature refers to this fragment as *finite CCS*. Finite CCS extends basic parallel CCS with the restriction and relabeling operators; the axioms for this subset of CCS appear in Table 3.

The axioms for $\backslash L$ and $[f]$ only explain how these operators interact with nil, prefixing, and $+$. That no rules are needed defining the interaction between $|$ and $\backslash L$, or $\backslash L$ and $[f]$, is a consequence of the fact that the innermost occurrences of these so-called *static* operators (with nil, prefixing, and $+$ being the *dynamic* ones) can be eliminated by repeated use of the laws for the operator in conjunction with (A1)–(A4). This argument may be formalized and used to show that rule set E_3 constitutes a sound and complete axiomatization of $\sim$ for finite CCS.

Rules for Recursive Processes. In order to axiomatize full CCS, we need rules for reasoning about terms that include process variables. Unfortunately, results from computability theory imply that no complete axiomatization can exist for $\sim$ for full CCS. (The set of equalities one can prove using any axiomatization can only be recursively enumerable; however, $\sim$ for full CCS is known not to be recursively enumerable.) However, two useful heuristics have been developed for handling process variables, and we review these here.

Both techniques take the form of inference rules and are therefore similar in form to the SOS rules used to define the operational semantics of CCS. The first rule, called the *unrolling rule,* states that a process

invocation is equivalent to the body of the invocation.

$$(\mathbf{U}\mathrm{nr})\ \frac{C \triangleq p}{C = p}$$

The second inference rule is often called the *unique fixpoint induction* principle, and stating it relies on introducing the notion of *equation* and *solution*. Given a variable X and a CCS term t potentially containing X, and only X, free, we call the expression $X = t$ an *equation*. A CCS process p is a *solution* to $X = t$ if and only if $p \sim t[p/X]$, where $t[p/X]$ is the CCS term obtained by replacing all occurrences of variable X by p. An equation has a *unique solution up to* $\sim$ if for any two solutions p and q to the equation, $p \sim q$. We may now formulate the unique fixpoint induction rule as follows.

$$(\mathbf{UFI})\ \frac{p = t[p/X] q = t[q/X]}{p = q}\ (X = t \text{ has a unique solution})$$

This rule allows one to conclude that two terms are equal, provided one can prove that they are both solutions to the same equation and the equation has a unique solution.

A couple of comments about (UFI) are in order. Firstly, every equation $X = t$ has a solution: given definition $X \triangleq t$, it is easy to see that process X is a solution of $X = t$. Secondly, (UFI) is only useful insofar as one may readily identify when equations have a unique solution. One large class of equations can be defined as follows.

Definition 11. *Let X be a variable, and t be a CCS involving X. Then X is guarded in t if every occurrence of X in t falls within the scope of a prefix operator.*

For example, X is guarded in $a.X$ and $a.X|(b.(X + c.nil))$, but it is not guarded in $X + b.X$. We now have the following result.

Theorem 12. Let X be guarded in t. Then the equation $X = t$ has a unique solution up to $\sim$.

As an application of (Unr) and (UFI), suppose we wish to prove that A and B are bisimilar, where $A \triangleq a.A$ and $B \triangleq a.a.B$. Consider the equation $X = a.a.X$. We can show that both A and B are solutions to this equation:

$$\begin{aligned} A &= a.A && \text{by } (\mathbf{U}\mathrm{nr}) \\ &= a.a.A && \text{by } (\mathbf{U}\mathrm{nr}) \\ B &= a.a.B && \text{by } (\mathbf{U}\mathrm{nr}) \end{aligned}$$

Since X is guarded in $a.a.X$, $X = a.a.X$ has a unique solution, and consequently, using (UFI), one may conclude that $A = B$.

Axiomatizing $\approx^C$. This section presents an axiomatization for $\approx^C$ and CCS. Following the development in the previous subsection, we first consider the finite-CCS fragment and then full CCS.

Axiomatizing Finite CCS. To begin with, it should be noted that the axioms in rule set E_3 of Table 3 are also sound for $\approx^C$, since whenever $p \sim q$ it immediately follows that $p \approx^C q$. In order to obtain a full axiomatization for $\approx^C$, then, we need only add axioms reflecting the special status of the action τ in this congruence.

One tempting axiom to add would be $x = \tau.x$; however, this is not sound for $\approx^C$, since it would allow one to prove that $\tau.a.nil = a.nil$, which is not valid. The correct rules are listed in Table 4 and are often called the τ *laws*.

Rule (τ1) allows for the "absorption" of τ actions that immediately follows prefixing operations. Rule (τ2) is more subtle, and may be understood as follows. First, note that any strong transition of $\tau.x$ is also a strong

Table 4. Axiomatizing $\approx^C$ for Finite CCS: Rule Set E_4

(A1)–(A4) from Table 1;	
(Exp) from Table 2;	
(Res1)–(Res3), (Rel1)–(Rel3) from Table 3	
(τ1)	$a.\tau.x = a.x$
(τ2)	$x + \tau.x = \tau.x$
(τ3)	$a.(x + \tau.y) = a.(x + \tau.y) + a.y$

transition of $x + \tau.x$. Secondly, any strong transition of $x + \tau.x$, including any τ-transition, may be matched by an appropriate weak transition in $\tau.x$. The final rule, (τ3) is perhaps the most difficult to interpret; note that the strong transition

$$a.(x + \tau.y) + a.y \xrightarrow{a} y$$

of the right-hand side may however be matched by the weak transition

$$a.(x + \tau.y) \stackrel{a}{\Longrightarrow} y$$

of the left-hand side.

Somewhat surprisingly, these rules suffice; the axiomatization E_4 is sound and complete for $\approx^C$ and finite CCS.

Axiomatizing Full CCS. The same observations for $\sim$ also hold for $\approx^C$ vis-à-vis sound and complete axiomatizations: none can exist. The (Unr) and (UFI) rules nevertheless still hold, although the characterization of which equations have unique fixpoints becomes somewhat more complex; guardedness no longer suffices. To see this, consider the equation $X = \tau.X$. X is guarded in $\tau.X$, and yet any process capable of an initial τ action is a solution to this equation up to $\approx^C$. In particular, $\tau.a.nil \approx^C \tau.\tau.a.nil$ and $\tau.b.nil \approx^C \tau.\tau.b.nil$, and yet $\tau.a.nil \not\approx^C \tau.b.nil$.

One potential solution to this problem is to require a stronger condition than guardedness in equations.

Definition 13. *Let X be a variable and t a CCS term involving X. Then X is strongly guarded in t if every occurrence of X falls within the scope of a prefixing operator a where $a \neq \tau$.*

That is, X is strongly guarded in t if a prefix operator involving a visible action "guards" each occurrence of X in t. Note that X is not strongly guarded in $\tau.X$. However, even if X is strongly guarded in t, it does not follow that $X = t$ has a unique solution up to $\approx^C$. To see this, consider the equation

$$X = (a.X|\overline{a}.nil) \setminus \{a\}$$

X is strongly guarded in the right-hand side of the equation, and yet it can be shown that e.g. $\tau.b.nil$ and $\tau.c.nil$ are both solutions. We may nevertheless solve this problem by requiring the following.

Definition 14. *Let X be a variable, and t a CCS term involving X. Then X is sequential in t if no occurrence of X in t falls within the scope of a parallel composition operator.*

As examples, X is sequential in $a.X$ and $\tau.X + (b.nil|c.nil)$ but not sequential in $a.X|b.nil$. The following can now be proved.

(S[put/ack,get_ack/ack]|M|R[get/msg,put_ack/ack])\{get,put,get_ack,put_ack}
= send.
((msg.ack.S)[put/ack,get_act/ack]|M|R[get/msg,put_ack/ack])/{get,put,get_ack,put_ack})
by (Exp), (Rel1)-(Rel3), (Res1)-(Res3)
= send.τ.
((ack.S)[put/ack,get_ack/ack]|get.M)|R[get/msg,put_ack/ack])\{get,put,get_ack,put_ack})
by (Exp), (Rel1)-(Rel3), (Res1)-(Res3)
= send.
((ack.S)[put/ack,get_ack/ack]|get.M)|R[get/msg,put_ack/ack])\{get,put,get_ack,put_ack})
by (τ1)
= send.τ.
((ack.S)[put/ack,get_ack/ack]|M|recv.ack.R)[get/msg,put_ack/ack])\{get,put,get_ack,put_ack})
by (Exp), (Rel1)-(Rel3), (Res1)-(Res3)
= send.
((ack.S)[put/ack,get_ack/ack]|M|recv.ack.R)[get/msg,put_ack/ack])\{get,put,get_ack,put_ack})
by (τ1)
= send.recv.
((ack.S)[put/ack,get_ack/ack]|M|ack.R)[get/msg,put_ack/ack])\{get,put,get_ack,put_ack})
by (Exp), (Rel1)-(Rel3), (Res1)-(Res3)
= send.recv.τ.
((ack.S)[put/ack,get_ack/ack]|get_ack.M|R[get/msg,put_ack/ack])\{get,put,get_ack,put_ack})
by (Exp), (Rel1)-(Rel3), (Res1)-(Res3)
= send.recv.
((ack.S)[put/ack,get_ack/ack]|get_ack.M|R[get/msg,put_ack/ack])\{get,put,get_ack,put_ack})
by (τ1)
= send.recv.τ.
(S[put/ack,get_ack/ack]|M|R[get/msg,put_ack/ack])\{get,put,get_ack,put_ack})
by (Exp), (Rel1)-(Rel3), (Res1)-(Res3)
= send.recv.
(S[put/ack,get_ack/ack]|M|R[get/msg,put_ack/ack])\{get,put,get_ack,put_ack})
by (τ1)

Fig. 3. Proving that P = Svc.

Theorem 15. Let $X = t$ be an equation with X strongly guarded and sequential in t. Then $X = t$ has a unique solution up to $\approx^C$.

We conclude this section with an extended example illustrating the use of the axioms. Recall the simple communications protocol P given in the subsection "The Syntactic Form of CCS Processes" and the specification Svc given below Definition 7. We may establish that $E_4 \cup \{(\text{Unr}), (\text{UFI})\} \vdash \texttt{P} = \texttt{Svc}$ as follows. First note that X is strongly guarded and sequential in send.recv.X and consequently has a unique solution up to $\approx^C$. Therefore, we need only show that both P and Svc are solutions to this equation; then, by (UFI), P = Svc. Now, Svc = send.recv .Svc by (Unr)

so Svc is a solution. As for P, we can prove that

P = (S[put/msg,get_ack/ack] |M|R[get/msg,put_ack/ack])
\{get,put,get_ack,put_ack}

using (Unr), so it suffices to prove that the right-hand side is a solution to the given equation. The proof of this may be found in Fig. 3.

Refinement Orderings for CCS

This article has so far concentrated on the role of behavioral equivalences in process algebra in general, and CCS in particular. We now shift our attention to refinement orderings, and to a particular class of refinement orderings that are often referred to as the failures/testing orderings. This section presents a definition of these orderings and gives axiomatizations for them for CCS.

The Failures/Testing Orderings. The motivation for the failures/testing orderings arises from two sources. On the one hand, equivalences sometimes impose overly severe restrictions on a designer defining a lower-level design that is intended to implement a higher-level one. In particular, equivalences require that the behaviors of the designs be identical; this precludes a higher-level design offering several possibilities for behavior or including "don't-care points." This suggests that an ordering in which a more deterministic system is larger, or better, than a less deterministic one would be desirable. On the other hand, while $\approx$ and $\approx^C$ abstract from internal computation and are sensitive to deadlock, it can be argued that they are overly sensitive to unobservable differences in the branching structure of systems. As an example, consider the two CCS definitions $P \triangleq a.b.c.nil + a.b.d.nil$ and $Q \triangleq a.(b.c.nil + b.d.nil)$. These two systems are not related by $\approx$; the formula $[[a]]\langle\langle b\rangle\rangle\langle\langle c\rangle\rangle tt$ is satisfied by the latter and not the former. However, a user ought not to be able to distinguish them, since to a user it does not matter when the nondeterministic choice that ultimately eliminates the possibility of c or d is made.

The failures (5,6) and testing (7,8) orderings differ substantially in their approaches to addressing these issues, and yet the resulting orderings turn out to coincide. In this section we follow the failures presentation given in Ref. 9 because it requires the introduction of less notation given the machinery we have already developed. We need the following definitions.

Definition 15. *Let $\langle Q, A, \rightarrow\rangle$ be a LTS with $\tau \in A$, let $q \in Q$, and let $s \in (A - \{\tau\})*$ be a sequence of visible actions.*

(1) $q \stackrel{s}{\Rightarrow}$ holds if there exists q' such that $q \stackrel{s}{\Rightarrow} q'$. In this case we say s is a trace of q. $L(q)$ denotes the set of all traces of q.

(2) q refuses $B \subseteq A - \{\tau\}$ if $|B| < \infty$ and for all $b \in B$, there exists no q' such that $q \stackrel{b}{\Rightarrow} q'$.

(3) q is divergent, written $q \Uparrow$, if and only if there exists an infinite sequence $q_0, q_1, \ldots$ such that $q = q_0$ and $q_i \stackrel{\tau}{\rightarrow} q_{i+1}$ for all $i \geq 0$. $q \Uparrow s$ if and only if there exists a prefix s' of s and state q' such that $q \stackrel{s'}{\Rightarrow} q'$ and $q' \Uparrow$. When this is the case we say q diverges on s. We write $q \Downarrow s$ if $q \Uparrow s$ is not true and say that q converges on s in this case.

(4) A state q is totally convergent if $q \Downarrow s$ holds for all sequences s.

(5) Let s be a sequence of visible actions and $B \subseteq A$ be finite. Then $\langle s, B\rangle$ is a failure for q if either $q \Uparrow s$ or there is a q' such that $q \stackrel{s}{\Rightarrow} q'$ and q' refuses B. We use $F(q)$ to represent the set of all failures of q.

The failures/testing ordering rely on the notions of *trace, refusal, divergence,* and *failure.* Intuitively, a trace of a state consists of a sequence of visible actions the state can perform, with arbitrary amounts of internal computation allowed in between. A refusal consists of a finite set of visible actions that a state is incapable of engaging in, no matter how much internal computation is performed. A state is divergent if it can engage in an infinite sequence of internal transitions, thereby ignoring its environment; $q \Uparrow s$ holds if, in the course of "executing" s, q could enter a divergent state. Finally, a failure consists of a sequence of actions and a set of "offered actions" that a state can fail to complete, either by diverging in the course of performing the sequence or completing the sequence and arriving at a state that is incapable of responding to the offered actions. As examples, consider the following.

- The pair $\langle a, \{b\}\rangle$ is a failure of $a.b.nil + a.c.nil$ and of $a.(\tau.b.nil + \tau.c.nil)$ but not of $a.(b.nil + c.nil)$. Both of the former processes have $\stackrel{a}{\Rightarrow}$ transitions to *c.nil,* which refuses $\{b\}$; the last process has no such transition.
- Consider $\mathtt{D} \triangleq \tau.\mathtt{D}$; then $\mathtt{D} \Uparrow s$ for any sequence s of visible actions, and consequently $\langle s, B\rangle$ is a failure for any D for any sequence s and finite set of actions B.

The sets $L(q)$ and F(q) satisfy a number of properties. For example, the empty sequence ε is in $L(q)$ for any q. In addition, if $q \Downarrow s$ then $s \in L(q)$ if and only if there is a B such that $\langle s, B\rangle \in F(q)$. It should also be noted that if $\langle s, B\rangle \in F(q)$ and $B' \subseteq B$ then $\langle s, B'\rangle \in F(q)$. Readers are referred to Ref. 9 for other such properties.

We now introduce the following orderings and equivalences.

Definition 16. *Let $\langle Q, A, \rightarrow\rangle$ be a LTS, with $p, q \in Q$.*

(1) $p \sqsubseteq_L q$ if $L(p) \subseteq L(q)$; $p \approx_L q$ if $p \sqsubseteq_L q$ and $q \sqsubseteq_L p$.
(2) $p \sqsubseteq_F q$ if $F(p) \supseteq F(q)$; $p \approx_F q$ if $p \sqsubseteq_F q$ and $q \sqsubseteq_F p$.

The ordering $\sqsubseteq_L$ and $\sqsubseteq_F$ capture different aspects of system behavior. The former relates systems on the basis of their execution sequences; a "lesser" system has fewer execution possibilities. The latter identifies failure as undesirable; consequently, a "lesser process" has *more* possibilities for failure than a "greater one." In this case failure can either be the result of nondeterminism or of divergence; the more nondeterministic or divergent a system is, the more failures it has.

Both orderings are *preorders* on Q; that is, they are reflexive and transitive relations. The relation $\sqsubseteq_L$ is also referred to as the *may preorder* in Refs. 7 and 8, while $\sqsubseteq_F$ is called the *must preorder.* This terminology derives from connections with process testing: $p \sqsubseteq_L q$ holds if and only if every test that p may pass may also be passed by q, in a precisely defined sense, while $p \sqsubseteq_F q$ holds if and only if every test that p must pass must also be passed by q. In addition, if q is totally convergent, then $p \sqsubseteq_F q$ implies that $q \sqsubseteq_L p$. This follows because for any failure $\langle s, B\rangle$ of q, one has $s \in L(q)$.

Finally, it should be noted that in CCS, the system Div given by $\mathtt{Div} \triangleq \tau.\mathtt{Div}$ is a least element for both $\sqsubseteq_L$ and $\sqsubseteq_F$. That is, $\mathtt{Div} \sqsubseteq_L p$ and $\mathtt{Div} \sqsubseteq_F p$ for any p.

For many process algebras $\sqsubseteq_L$ and $\sqsubseteq_F$ are *precongruences:* "larger" systems may be substituted for "smaller" ones inside any context, with the resulting overall system being larger after the substitution. For CCS, $\sqsubseteq_L$ is a precongruence, but $\sqsubseteq_F$ is not, owing the effect that initial internal computation can have on the $+$ operator. As was the case with $\approx$, one may identify the largest precongruence $\sqsubseteq^C{}_F$ contained within $\sqsubseteq_F$ for CCS; it turns out that for CCS systems p and q, $p \sqsubseteq^C{}_F q$ if and only if the following hold: $p \sqsubseteq_F q$, and $p\, p//_{_}{}^{\tau<}\!\rightarrow$ implies $q\, p//_{_}{}^{\tau}\!\rightarrow$.

The relations $\sqsubseteq_F$ and $\sqsubseteq^C{}_F$ have attracted much more attention in the literature than $\sqsubseteq_L$, because of certain *full-abstractness* results than have been established for the former. In particular, for a number of languages it turns out that $\sqsubseteq_F/\sqsubseteq^C{}_F$ are the coarsest (i.e. most permissive) preorders that preserve deadlock information, in a precisely defined sense. Accordingly, the remainder of this section is devoted to a study of $\sqsubseteq^C{}_F$.

Axiomatizing $\sqsubseteq^C{}_F$ for CCS. As was the case for $\sim$ and $\approx^C$, $\sqsubseteq^C{}_F$ has been axiomatized for (fragments of) CCS. We present the axiomatization for finite CCS below and talk briefly about mechanisms for handling recursive processes.

Finite CCS. The axiomatization for finite CCS appears in Table 5. Unlike the other axiomatizations we have seen, it is an *inequational* axiomatization: it is used to prove statements of the form $p \leq q$ rather than $p = q$. The axioms therefore include inequalities; equalities such as rule (F1) should be interpreted as shorthand for two inequalities, one in each direction.

Table 5. Axiomatizing $\sqsubseteq_F^C$ for Finite CCS: Rule Set E_4

(A1)–(A4) from Table 1; (Exp) from Table 2;	
(F1)	$a.x + a.y = a.(\tau.x + \tau.y)$
(F2)	$x + \tau.y \leq \tau.(x + y)$
(F3)	$a.x + \tau.(a.y + z) = \tau.(a.x + a.y + z)$
(F4)	$\tau.x \leq x$
(F5)	$\tau.x + \tau.y \leq x$

To see how these rules may be used to derive results, we give a sample proof of $E_4 \vdash a.b.nil + a.c.nil \leq a.b.nil$:

$$\begin{aligned} a.b.nil + a.c.nil &= a.(\tau.b.nil + \tau.c.nil) && \text{by } (\mathbf{F1}) \\ &\leq a.b.nil && \text{by } (\mathbf{F5}) \end{aligned}$$

The rules in Table 5 are sound and complete for $\sqsubseteq^C{}_F$ for finite CCS.

Reasoning about Recursive Processes. To handle recursive processes, one may use rules (Unr) and (UFI) as given in the subsection "Rules for Recursive Processes." A sufficient condition for the existence of unique solutions to equations includes a requirement of *divergence-freedom* along with the strong-guardedness and sequentiality requirements needed for $\approx^C$.

Interpreting systems as sets of failures also permits the use of reasoning techniques from fixpoint theory in denotational semantics (8). This is because the collection of sets of failures can be turned into a *domain.* We do not pursue this topic further, however.

Computing Behavioral Relations for Finite-State Systems

The previous sections have developed several semantic equivalences and refinement orderings in the context of CCS, and (in)equational axiomatizations have been presented for determining when two systems are related. However, the equational reasoning supported by these axiomatizations is tedious to undertake by hand. When the systems in question are *finite-state,* meaning that the rooted labeled transition systems for them contain only finitely many distinct states, these relations can be computed algorithmically. This section discusses some of the ideas underlying these decision procedures.

Computing Behavioral Equivalences. Most behavioral equivalences can be computed by combining appropriate LTS transformations with an algorithm for calculating bisimulation equivalence (10). Accordingly, we first discuss techniques for deciding $\sim$ and then show how these methods may be used in the computation of other equivalences as well.

Calculating $\sim$. Algorithms for $\sim$ come in two basic varieties. *Global* algorithms require the *a priori* construction of the state spaces of the systems in question before any analysis can be undertaken. *On-the-fly* approaches, on the other hand, combine analysis with state-space construction. The latter algorithms offer obvious potential benefits: when systems are inequivalent, this may be determined by examining only a subset of their states. These approaches are relatively new, however, and have not proven themselves in practice. Global approaches also enjoy better asymptotic efficiency than existing on-the-fly methods. Consequently, we only discuss the former.

Global approaches to calculating $\sim$ over a finite-state LTS (11–13) compute the *equivalence classes* of $\sim$ using approximation-refinement techniques. Typically, these algorithms begin with a very coarse approxima-

tion to $\sim$: they assume that every state is related to every other state, meaning that there is one equivalence class. Existing classes that are found to contain inequivalent states are then split; the determination of inequivalence relies on examining the transitions emanating from states and the equivalence classes containing the targets of these transitions. When no more splitting is possible, the final equivalence classes indeed represent the equivalence classes of $\sim$ over the given LTS. These algorithms are sometimes called *partition-refinement* algorithms, as the collections of equivalence classes are maintained as partitions (i.e. lists of disjoint sets of states). The best algorithm has complexity $O(m \log n)$, where m represents the number of transitions in the LTS and n the number of states (11,13).

In order to use a partition-refinement algorithm to determine whether two CCS expressions are bisimilar, one would first construct the labeled transition system whose states consist of all CCS expressions reachable from the two in question. A partition-refinement algorithm may then be applied to this LTS, and if the two expressions in question ever wind up in different equivalence classes, they are inequivalent. Otherwise, if the refinement procedure terminates with them in the same class, then they are equivalent.

Partition-refinement algorithms may also be used to *minimize* LTSs with respect to $\sim$. This is done by replacing states by equivalence classes; the resulting LTS contains exactly one state per equivalence class.

Computing Other Equivalences. As the introduction to this section indicates, a variety of other behavioral equivalences may be computed by first applying a transformation to the underlying LTS and then using an algorithm for $\sim$. Here we present two examples of this approach.

Calculating $\approx$. To calculate the $\approx$-equivalence classes of a LTS, one may alter the LTS by replacing the $\xrightarrow{a}$-transitions by $\stackrel{\hat{a}}{\Rightarrow}$-transitions and then computing $\sim$ over the transformed LTS. A similar approach works for $\approx^C$, although one must first transform the LTS to ensure that the start state contains no incoming transitions and then replace $\xrightarrow{a}$-transitions from the start state by $\stackrel{a}{\Rightarrow}$-transitions (and not $\stackrel{\hat{a}}{\Rightarrow}$-transitions).

Computing $\approx_S$. To determine whether two states in a given finite-state LTS are strong trace equivalent, one may apply the well-known subset construction to *determinize* the LTS (14) and then compute the equivalence classes of $\sim$. The two states in question will have the same strong traces if and only if the subsets containing only these states are bisimilar in the transformed LTS.

Computing Refinement Orderings. The calculation of refinement orderings follows a similar pattern to that of equivalences: a given ordering can be computed by combining an LTS transformation with a procedure for a certain generic ordering (10,15). The generic ordering is somewhat less standard than $\sim$, but in many cases the *simulation ordering* may be used. In the remainder of this section we define this ordering and indicate very briefly how it is used as a basis for computing other relations.

The Simulation Ordering. Given an LTS $\langle Q, A, \rightarrow \rangle$, a *simulation* is a relation $R \subseteq Q \times Q$ with the property that when $\langle p, q \rangle \in R$, then the following holds for all $a \in A$:

$$p \xrightarrow{a} p' \quad \text{implies} \quad q \xrightarrow{a} q' \quad \text{for some } q' \text{ with } \langle p', q' \rangle \in R$$

So if p is related to q in a simulation, then q can "simulate" the behavior of p by "matching" its transitions. The simulation ordering then may be defined as follows: $p \sqsubseteq q$ if and only if there exists a simulation R with $\langle p, q \rangle \in R$.

Algorithms for computing $\sqsubseteq$ on finite-state LTSs follow a similar strategy to that for $\sim$ in that they use approximation refinement. Initially, every state is assumed to be related to every other state; then, as pairs of states are found not to be related because the first has a transition that cannot be "simulated" by the second, they are removed. When no more pairs can be removed, the remaining pairs constitute $\sqsubseteq$ for this LTS.

Since $\sqsubseteq$ is not an equivalence, partitions cannot be used as data structures, and the resulting algorithms exhibit somewhat worse worst-case performance: the best algorithms use $O(mn)$ time, where m is the number of transitions and n the number of states (15).

Computing Other Orderings. As an example of how an algorithm for $\sqsubseteq$ may be used in the calculation of other relations, consider the trace-containment relation: $p \sqsubseteq_L q$ if and only if $L(p) \subseteq L(q)$. This relation may be computed by first replacing $\stackrel{a}{\rightarrow}$ transitions by $\stackrel{\hat{a}}{\Rightarrow}$ ones, determinizing the resulting LTS using the subset construction, and then applying a $\sqsubseteq$ algorithm to the result. Other relations, including the failures/testing ordering, may be computed similarly (10).

Tool Support. Several tools have been implemented that include implementations of algorithms for different behavioral relations. Noteworthy examples include Aldébaran (16), the Concurrency Workbench (17), and FDR (18).

Other Process Algebras

The presentation in this article has focused on a particular process algebra, CCS, and on semantic relations for CCS. In this section we discuss other process algebras and process-algebra-oriented results. Since 1980 over 1000 journal and conference papers have been published in the area; as a result, the discussion here will necessarily be incomplete. Interested readers are referred to the forthcoming *Handbook of Process Algebra,* to be published by Elsevier, for a more complete account of the state of the art.

Schools of Process Algebras. The discussion in this article has followed the approach advocated by the *Edinburgh school* of process algebra, so named because CCS was invented at the University of Edinburgh. The Edinburgh school places primacy on operational semantics, with equivalences and refinement relations then defined on labeled transition systems resulting from these operational definitions. The chief virtue of this approach lies in its insistence on understanding language constructs operationally; this emphasis accords well with intuitions about system behavior. The drawback of this approach arises from the fact that since operational equivalences and refinement orderings are defined on language-independent structures (i.e. labeled transition systems), determining which relations are congruences becomes nontrivial.

Two other schools of process algebra have also arisen. The *Amsterdam school* focuses on equational axioms as the basis for defining the semantics of languages (19). In this approach one defines the syntax of an algebra and then provides a set of axioms that one uses to deduce equivalences. Traditional techniques from universal algebra may then be used to construct models of these equational theories. These constructions ensure that the model-theoretic notions of equivalence are congruences for the language in question; the drawback is that equations obscure the operational intuitions underlying operators in the algebra.

The *Oxford school* focuses on *denotational semantics* as the basis for defining process algebras (6). The Oxford approach relies on defining a mathematical space of *system meanings* and then interpreting algebraic operators as functions in this space. The space most studied by Oxford adherents consists of *failure sets* as presented in the section "Refinement Orderings for CCS" above; process constructors then become functions mapping sets of failures to sets of failures. As with the Amsterdam approach, the virtue of this methodology is that the semantic equivalence inherited from the semantic space is guaranteed to be a congruence for the language; additionally, traditional techniques from denotation semantics may be used to define the semantics of recursive processes in a mathematically elegant fashion. The drawback arises from the paucity of operational insight the semantics provides for the operators.

Operators in Traditional Algebras. The different schools just mentioned have also traditionally focused on including somewhat different operators in their algebras. CCS includes a parallel composition operator that supports binary synchronous communication. The *Algebra of Communicating Processes* (*ACP*) algebra developed by the Amsterdam school, on the other hand, allows the specific communication mechanism to be parametrized; by including different axioms one obtains different synchronization behavior. ACP also includes a traditional sequential composition operator that generalizes the prefixing construct of CCS. Theoretical CSP (*TCSP*), the process algebra studied by the Oxford school, features multiway rendezvous as its model of in-

teraction; a hiding operator allows actions to be converted into internal actions. Another novel feature of this language is its separation of choice (i.e. +) into two constructs, external and internal. The former can only be resolved by visible actions, while the latter is always resolved autonomously, without interaction from the environment.

These algebras have also inspired the development of LOTOS, a process algebra with explicit data passing that is an ISO standard protocol specification notation (20). LOTOS combines CSP-like operators with a facility for user-defined data types; as in CCS actions may be categorized as inputs or outputs, with the former extended with a capability for binding incoming values to variables and the latter including specific values to be output.

Algebras for Synchrony. Traditional process algebras, including those mentioned above, typically include a synchronous model of communication but an asynchronous model of execution. That is, processes interact by synchronizing, but not every process in a system need execute in order for a system transition to take place. This makes traditional algebras useful for modeling loosely coupled systems, but it renders them problematic as vehicles for describing synchronous, globally clocked systems such as traditional digital circuits. To overcome this difficulty, several researchers have proposed algebras whose parallel composition operator requires all subsystems to engage in transitions in order the system to perform an execution step. The best-known of these is *synchronous CCS* (21), whose action set forms a commutative group whose product operator is interpreted as "simultaneous execution." Other synchronous process algebras of note include Meije (22) and CIRCAL (23); the latter was specifically developed for reasoning about circuits. All three algebras use equivalences based on strong bisimulation; weak equivalences such as observational equivalence will necessarily not be congruences for such languages, since the internal computation a subsystem may engage in will directly affect the transitions available to the surrounding system.

Metaalgebraic Results. The algebras just described feature a variety of different operators; in each case, the Edinburgh approach (which has become dominant) requires the proof of congruence results for bisimulation. Some researchers have addressed this problem by proving that, provided the SOS rules defining a langauge's operators satisfy a certain format, bisimulation is guaranteed to be a congruence (24,25,26). Other results show how equational axiomatizations for languages satisfying these requirements may be automatically derived (27).

Other Semantic Relations. Researchers have also investigated relations other than the ones presented here. Branching bisimulation (28) aims to remedy a perceived defect of observational congruence that allows transitions in one process to be matched by weak transitions in the other that permit inequivalent states to "transitioned through." This equivalence is somewhat finer (i.e. relates fewer systems) than observational equivalence, and a sound and complete axiomatization for finite ACP terms, and algorithms for finite-state systems, have been developed. Ready simulation (24) represents a refinement ordering that is fully abstract for deadlock when the language considered includes all operators definable using SOS rules of a certain format. A number of other relations have also been proposed; the interested reader is referred to Ref. 29 for a thorough survey and taxonomy.

Capturing Other Features of System Behavior. Traditional process algebras have focused on nondeterminism and synchronization as the essential behavioral features distinguishing concurrent systems from sequential ones. Inspired by the elegance of the resulting theories, researchers have attempted to develop operational theories that allow other aspects of system behavior to be captured (in)equationally. One strand of inquiry has focused on so-called *true concurrency.* One criticism of traditional process algebras is that they "reduce" concurrency to nondeterminism by interpreting parallelism as interleaving. Truly concurrent models instead attempt to capture "true" notions of simultaneity. A number of different theories have been developed, and a full account is beyond the scope of this chapter. A good starting point, however, may be found in Ref. 30, which introduces the notion of location of a transition explicitly into the operational semantics of CCS and develops a bisimulation-based theory of equivalence based on this.

Other work has focused on including notions of *priority* into the operational semantics of process algebras. The first such work (31) extends ACP action with priority and and operator for "enforcing" priorities. In Ref. 32

CCS actions are enriched with a two-level priority structure, with high-priority actions intuitively being thought of as "interrupts." Camilleri and Winskel (43) opt instead for a prioritized choice operator that gives precedence to one choice over another when both are enabled. Also worthy of note is the resource-oriented process algebra ACSR (33), which allows the modeling of resource contention in which different resource requests may be given different priorities. In all of these cases, semantic equivalences based on strong bisimulation are defined and axiomatizations developed.

Process algebras for real-time systems have also been developed. Generally speaking, these theories introduce special "time-passing" actions, with all other actions being viewed as instantaneous. The *Algebra of Timed Pocesses* (34) pioneered this approach, with useful variants being proposed in Ref. 35.

Another area of ongoing research involves the incorporation of probabilistic behavior into systems, with a view toward providing a theory in which quality-of-service statements can be made. One strand of this research augments traditional process algebra with notions of probabilistic choice in which nondeterminism is resolved probabilistically (36,37,38). Other pieces of work incorporate notions of time and probability in order to model stochastic systems, in which the time needed to perform a given action is drawn from a continuous probability distribution. Noteworthy examples include Refs. 39 and 40.

Conclusion

This article has surveyed results in the area of process algebra. It has presented several behavioral equivalences and refinement orderings, and it has shown how they may be axiomatized in the setting of CCS (4). Decision procedures for finite-state systems have also been touched on. The treatment has necessarily been sketchy, and much interesting material has been omitted, including a variety of case studies illustrating different applications of process algebra. Interested readers may turn to Refs. 18, 41, and 42 as a starting point for investigating this topic.

BIBLIOGRAPHY

1. R. Milner *A Calculus of Communicating Systems*, Berlin: Springer-Verlag, 1980.
2. G. D. Plotkin A structural approach to operational semantics, Technical Report DAIMI-FN-19, Computer Science Department, Aarhus University, Aarhus, Denmark, 1981.
3. M. C. B. Hennessy R. Milner Algebraic laws for nondeterminism and concurrency, *J. Assoc. Comput. Mach.*, **32** (1): 137–161, 1985.
4. R. Milner *Communication and Concurrency*, London: Prentice-Hall, 1989.
5. S. D. Brookes C. A. R. Hoare A. W. Roscoe A theory of communicating sequential processes, *J. Assoc. Comput. Mach.*, **31** (3): 560–599, 1984.
6. C. A. R. Hoare *Communicating Sequential Processes*, London: Prentice-Hall, 1985.
7. R. De Nicola M. C. B. Hennessy Testing equivalences for processes, *Theor. Comput. Sci.*, **34**: 83–133, 1983.
8. M. C. B. Hennessy *Algebraic Theory of Processes*, Boston: MIT Press, 1988.
9. M. Main Trace, failure and testing equivalences for communicating processes, *Int. J. Parallel Program.*, **16** (5): 383–400, 1987.
10. R. Cleaveland M. C. B. Hennessy Testing equivalence as a bisimulation equivalence, *Formal Aspects Comput.*, **5**: 1–20, 1993.
11. J.-C. Fernandez An implementation of an efficient algorithm for bisimulation equivalence, *Comput. Program.*, **13**: 219–236, 1989/1990.
12. P. Kanellakis S. A. Smolka CCS expressions, finite state processes, and three problems of equivalence, *Inf. Comput.*, **86** (1): 43–68, 1990.
13. R. Paige R. E. Tarjan Three partition refinement algorithms, *SIAM J. Comput.*, **16** (6): 973–989, 1987.

14. J. Hopcroft J. D. Ullman *Introduction to Automata Theory, Languages, and Computation*, Reading, MA: Addison-Wesley, 1979.
15. U. Celikkan R. Cleaveland Generating diagnostic information for behavioral preorders, *Distributed Comput.*, **9**: 61–75, 1995.
16. A. Bozga *et al.* Protocol verification with the Aldébaran toolset. *Softw. Tools Technol. Transf.*, **1** (1+2): 166–183, 1997.
17. R. Cleaveland J. Parrow B. Steffen The Concurrency Workbench: A semantics-based tool for the verification of finite-state systems, *ACM Trans. Program. Lang. Syst.*, **15** (1): 36–72, 1993.
18. A. W. Roscoe *The Theory and Practice of Concurrency*, Upper Saddle River, NJ: Prentice-Hall, 1997.
19. J. C. M. Baeten W. P. Weijland *Process Algebra*, Cambridge: UK: Cambridge University Press, 1990.
20. T. Bolognesi E. Brinksma Introduction to the ISO specification language LOTOS, *Comput. Networks ISDN Syst.*, **14**: 25–59, 1987.
21. R. Milner Calculi for synchrony and asynchrony, *Theor. Comput. Sci.*, **25**: 267–310, 1983.
22. D. Austry G. Boudol Algèbre de processus et synchronisation. *Theor. Comput. Sci.*, **30**: 91–131, 1984.
23. G. Milne CIRCAL and the representation of communication, concurrency and time, *ACM Trans. Program. Lang. Syst.*, **7** (2): 270–298, 1985.
24. B. Bloom S. Istrail A. Meyer Bisimulation can't be traced, *J. Assoc. Comput. Mach.*, **42** (1): 232–268, 1995.
25. R. N. Bol J. F. Groote The meaning of negative premises in transition system specifications, *J. Assoc. Comput. Mach.*, **43** (5): 863–914, 1996.
26. J. F. Groote F. Vaandrage Structured operational semantics and bisimulation as a congruence, *Inf. Comput.*, **100** (2): 202–260, 1992.
27. L. Aceto B. Bloom F. Vaandrager Turning SOS rules into equations, *Inf. Comput.*, **111** (1): 1–52, 1994.
28. R. van Glabbeek P. Weijland Branching time and abstraction in bisimulation semantics, *J. Assoc. Comput. Mach.*, **43** (3): 555–600, 1996.
29. R. J. van Glabbeek *Comparative concurrency semantics, with refinement of actions*, Ph.D. Thesis, Free University, Amsterdam, 1990.
30. G. Boudol *et al.* Observing localities, *Theor. Comput. Sci.*, **114** (1): 31–61, 1993.
31. J. C. M. Baeten J. A. Bergstra J. W. Klop Syntax and defining equations for an interrupt mechanism in process algebra, *Fundam. Inf.*, **9**: 127–168, 1986.
32. R. Cleaveland M. C. B. Hennessy Priorities in process algebra, *Inf. Comput.*, **87** (1/2): 58–77, 1990.
33. R. Gerber I. Lee A resource-based prioritized bisimulation for real-time systems, *Inf. Comput.*, **113** (1): 102–142, 1994.
34. X. Nicollin J. Sifakis The algebra of timed processes ATP: Theory and application, *Inf. Comput.*, **114** (1): 131–178, 1994.
35. K. Larsen W. Yi Time-abstracted bisimulation: Implicit specifications and decidability, *Inf. Comput.*, **134** (2): 75–101, 1997.
36. J. C. M. Baeten J. A. Bergstra S. A. Smolka Axiomatizing probabilistic processes: ACP with generative probabilities, *Inf. Comput.*, **121** (2): 234–255, 1995.
37. K. G. Larsen A. Skou Bisimulation through probabilistic testing, *Inf. Comput.*, **94** (1): 1–28, 1991.
38. R. J. van Glabbeek S. A. Smolka B. Steffen Reactive, generative and stratified models of probabilistic processes, *Inf. Comput.*, **121** (1): 59–80, 1995.
39. R. Gorrieri M. Roccetti E. Stancampiano A theory of processes with durational actions, *Theor. Comput. Sci.*, **140** (1): 73–94, 1995.
40. P. Harrison J. Hillston Process algebras and their application to performance modelling, *Comput. J.*, **38** (7): 489–491, 1995.
41. J. C. M. Baeten (ed.) *Applications of Process Algebra*, Cambridge, UK: Cambridge University Press, 1990.
42. G. Bruns *Distributed Systems Analysis with CCS*, London: Prentice-Hall, 1997.
43. J. Camilleri G. Winskel CCS with prioritized choice, *Inf. Comput*, **116** (1): 26–37, 1995.

RANCE CLEAVELAND
SCOTT A. SMOLKA
State University of New York at Stony Brook

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Random Matrices
Standard Article

M. A. Stephanov[1], J. J. M. Verbaarschot[2], T. Wettig[3]
[1]State University of New York at Stony Brook
[2]Yale University
[3]State University of New York at Stony Brook

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2447
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (385K)

Abstract

The sections in this article are

Mathematical Methods I: Hermitian Matrices

Mathematical Methods II: Non-Hermitian Matrices

APplications and Advanced Topics

| | |

Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

RANDOM MATRICES

In general, random matrices are matrices whose matrix elements are stochastic variables. The main goal of random matrix theory (*RMT*) is to calculate the statistical properties of eigenvalues for very large matrices, which are important in many applications. Ensembles of random matrices first appeared in the mathematics literature as a p-dimensional generalization of the χ^2-distribution (1). Ensembles of real symmetric random matrices with independently distributed Gaussian matrix elements were introduced in the physics literature to describe the spacing distribution of nuclear levels (2). The theory of Hermitian random matrices was first worked out in a series of seminal papers by Dyson (3). Since then, *RMT* has had applications in many branches of physics, ranging from sound waves in aluminum blocks to quantum gravity. For an overview of the early history of *RMT* see the book by Porter (4). Another authoritative source on *RMT* is the book by Mehta (5). For a comprehensive review including the most recent developments see Ref. 6

Generally speaking, random matrix ensembles provide a statistical description of a complex interacting system. Depending on the hermiticity properties of the interactions, one can distinguish two essentially different classes of random matrices: Hermitian matrices with real eigenvalues, and matrices without hermiticity properties with eigenvalues scattered in the complex plane. This article provides an overview of the 10 different classes of Hermitian random matrices and then briefly covers non-Hermitian random matrix ensembles.

The best known random matrix ensembles are the Wigner–Dyson ensembles, which are ensembles of Hermitian matrices with matrix elements distributed according to

$$P(H)DH = \mathcal{N} e^{-[(N\beta)/4]\mathrm{Tr}H^{\dagger}H} DH \qquad (1)$$

Here, H is a Hermitian $N \times N$ matrix with real, complex, or quaternion real matrix elements. The corresponding random matrix ensemble is characterized by the Dyson index $\beta = 1$, 2, and 4, respectively. The measure DH is the Haar measure, which is given by the product over the independent differentials. The normalization constant of the probability distribution is denoted by N. The probability distribution in Eq. (1) is invariant under the transformation

$$H \to UHU^{-1} \qquad (2)$$

where U is an orthogonal matrix for $\beta = 1$, a unitary matrix for $\beta = 2$, and a symplectic matrix for $\beta = 4$. This is the reason that these ensembles are known as the Gaussian orthogonal ensemble (GOE), the Gaussian unitary ensemble (GUE), and the Gaussian symplectic ensemble (GSE), respectively. The GOE is also known as the Wishart distribution. Since both the eigenvalues of H and the Haar measure DH are invariant with respect to Eq. (2), the eigenvectors and the eigenvalues are independent, with the distribution of the eigenvectors given by the invariant measure of the corresponding orthogonal, unitary, or symplectic group.

There are two ways of arriving at the probability distribution in Eq. (1): first, from the requirement that the matrix elements are independent and are distributed with the same average and variance for an ensemble

invariant under Eq. (2); and second, by requiring that the probability distribution maximizes the information entropy subject to the constraint that the average and the variance of the matrix elements are fixed.

A second class of random matrices are the chiral ensembles (7) with the chiral symmetries of the quantum chromodynamics (*QCD*) Dirac operator. They are defined as the ensembles of $N \times N$ Hermitian matrices with block structure

$$H = \begin{pmatrix} 0 & C \\ C^\dagger & 0 \end{pmatrix} \tag{3}$$

and probability distribution given by

$$P(C)DC = \mathcal{N} \det{}^{N_f} \begin{pmatrix} 0 & C \\ C^\dagger & 0 \end{pmatrix} e^{-[(N\beta)/4]\mathrm{Tr}C^\dagger C} DC \tag{4}$$

Again, DC is the Haar measure, and N_f is a real parameter (corresponding to the number of quark flavors in *QCD*). The matrix C is a rectangular $n \times (n + \nu)$ matrix. Generically, the matrix H in Eq. (3) has exactly $|\nu|$ zero eigenvalues. Also generically, the *QCD* Dirac operator corresponding to a field configuration with the topological charge ν has exactly $|\nu|$ zero eigenvalues, in accordance with the Atiyah–Singer index theorem. For this reason, ν is identified as the topological quantum number. The normalization constant of the probability distribution is denoted by N. Also, in this case one can distinguish ensembles with real, complex, or quaternion real matrix elements. They are denoted by $\beta = 1$, $\beta = 2$, and $\beta = 4$, respectively. The invariance property of the chiral ensembles is given by

$$C \rightarrow UCV^{-1} \tag{5}$$

where U and V are orthogonal, unitary, and symplectic matrices, respectively. For this reason, the corresponding ensembles are known as the chiral Gaussian orthogonal ensemble (chGOE), the chiral Gaussian unitary ensemble (chGUE), and the chiral Gaussian symplectic ensemble (chGSE), respectively. A two sublattice model with diagonal disorder in the chGUE class was first considered in Ref. 8

A third class of random matrix ensembles occurs in the description of disordered superconductors. Such ensembles with the symmetries of the Bogoliubov-de Gennes Hamiltonian have the block structure

$$H = \begin{pmatrix} A & B \\ B^\dagger & -A^T \end{pmatrix} \tag{6}$$

where A is Hermitian, and, depending on the underlying symmetries, the matrix B is symmetric or antisymmetric. The probability distribution is given by

$$P(H)DH = \mathcal{N} \exp\left(-\frac{N\beta}{4}\mathrm{Tr}H^\dagger H\right) DH \tag{7}$$

where DH is the Haar measure and N is a normalization constant. For symmetric B the matrix elements of H can be either complex (C) or real (CI). For antisymmetric B the matrix elements of H can be either complex (D) or quaternion real (DIII). The name of the ensembles (in parentheses) refers to the symmetric space to which they are tangent. Since they were first introduced by Altland and Zirnbauer (9,10), we call them the Altland–Zirnbauer ensembles. A hopping model based on the class CI first entered in Ref. 11

A key ingredient in the classification of a Hamiltonian in terms of one of the preceding random matrix ensembles is its antiunitary symmetries. An antiunitary operator can be written as

$$U = AK \tag{8}$$

where A is unitary and K is the complex conjugation operator. For the classification according to the antiunitary symmetries, we can restrict ourselves to the following three possibilities: (1) the Hamiltonian does not have any antiunitary symmetries; (2) the Hamiltonian commutes with AK and $(AK)^2 = 1$; and (3) $[H, AK] = 0$ but $(AK)^2 = -1$. In the first case, the matrix elements of the Hamiltonian are complex; in the second case, it is always possible to find a basis in which the Hamiltonian is real; and in the third case, it can be shown that it is possible to organize the matrix elements of the Hamiltonian in quaternion real elements. These three possibilities are denoted by the number of degrees of freedom per matrix element, $\beta = 2$, $\beta = 1$, and $\beta = 4$, respectively. This triality characterizes the Wigner–Dyson ensembles, the chiral ensembles, and the Altland–Zirnbauer ensembles. In most cases, the antiunitary operator is the time-reversal symmetry operator. For systems without spin, this is just the complex conjugation operator. For systems with spin, the time reversal operator can be represented as $i\sigma_2 K$, where σ_2 is one of the Pauli matrices.

We have introduced ten random matrix ensembles. Each of these ensembles can be identified as the tangent space of one of the large families of symmetric spaces as classified by Cartan (see Table 1). The matrices in each of these ten ensembles can be diagonalized by a unitary transformation, with the unitary matrix distributed according to the group measure. For all ensembles, the Jacobian for the transformation to eigenvalues as new integration variables depends only on the eigenvalues. For an extensive discussion of the calculation of this type of Jacobian see Ref. 12 For the Wigner Dyson ensembles, the joint probability distribution of the eigenvalues is given by

$$P(\{\lambda\})d\{\lambda\} = \mathcal{N}|\Delta(\{\lambda\})|^\beta \prod_k e^{-N\beta\lambda_k^2/4} d\lambda_k \tag{9}$$

where the Vandermonde determinant is defined by

$$\Delta(\{\lambda\}) = \prod_{k>1} (\lambda_k - \lambda_l) \tag{10}$$

This factor results in correlations of eigenvalues that are characteristic for the random matrix ensembles. For example, one finds repulsion of eigenvalues at small distances.

For the remaining ensembles, the eigenvalues occur in pairs $\pm\lambda_k$. This results in the distribution

$$P(\{\lambda\})d\{\lambda\} = \mathcal{N}|\Delta(\{\lambda^2\})|^\beta \prod_k \lambda_k^\alpha e^{-N\beta\lambda_k^2/4} d\lambda_k \tag{11}$$

The values of β and α are given in Table 1.

Another well-known random matrix ensemble that is not in the preceding classification is the Poisson ensemble, defined as an ensemble of uncorrelated eigenvalues. Its properties are very different from the preceding *RMTs*, where the diagonalization of the matrices leads to strong correlations between the eigenvalues.

The physical applications of *RMT* have naturally biased the interest of researchers to Hermitian matrices (e.g., the Hamiltonian of a quantum system is a Hermitian operator and should be represented by a Hermitian matrix). A variety of methods, described in this article, have been developed to treat ensembles of Hermitian

Table 1. Random Matrix Ensemble, Corresponding Symmetric Space, and the Values for α and β

RMT	Symmetric Space	β	α
GOE	AI	1	—
GUE	A	2	—
GSE	AII	4	—
chGOE	BDI	1	$\lvert\nu\rvert + 2N_f$
chGUE	AIII	2	$1 + 2\lvert\nu\rvert + 2N_f$
chGSE	CII	4	$3 + 4\lvert\nu\rvert + 2N_f$
AZ-CI	CI	1	1
AZ-D	D	2	0
AZ-C	C	2	2
AZ-DIII	DIII	4	1

matrices. In contrast, non-Hermitian random matrices received less attention. Apart from the intrinsic mathematical interest of such a problem, a number of physically important applications that warrant the study of non-Hermitian random matrices exist.

The simplest three classes of non-Hermitian random matrices, introduced by Ginib 13, are direct generalizations of the GOE, GUE, and GSE. They are given by an ensemble of matrices C without any Hermiticity properties and a Gaussian probability distribution given by

$$P(C)DC = \mathcal{N} e^{-[N\beta/2]\mathrm{Tr}C^{\dagger}C} DC \qquad (12)$$

where DC is the product of the differentials of the real and imaginary parts of the matrix elements of C. Such matrices can be diagonalized by a similarity transformation, with eigenvalues scattered in the complex plane. The probability distribution is not invariant under this transformation, and therefore the eigenvalues and the eigenvectors are not distributed independently. Similarly to the Hermitian ensembles, the matrix elements can be chosen real, arbitrary complex, or quaternion real.

The case of the arbitrary complex non-Hermitian random matrix ensemble Eq. (12) with $\beta = 2$ is the simplest. The joint probability distribution of eigenvalues $\{\lambda\} = \{\lambda_1, \ldots, \lambda_N\}$ is given by a formula similar to Eq. (9):

$$P(\{\lambda\})d\{\lambda\} = \mathcal{N}|\Delta(\{\lambda\})|^2 \prod_k e^{-N|\lambda_k|^2} dx_k dy_k \qquad (13)$$

where $x_k = \mathrm{Re}\ \lambda_k$, $y_k = \mathrm{Im}\ \lambda_k$. In the quaternion-real case, the joint probability distribution can also be written explicitly. In the case of real matrices, the joint probability distribution is not known in closed analytical form.

It is also possible to introduce non-Hermitian ensembles with a chiral structure, but such ensembles have received very little attention in the literature and are not discussed. What has received a great deal of attention in the literature are non-Hermitian deformations of the Hermitian random matrix ensembles. Among others, they enter in the statistical theory of S-matrix fluctuations (14), models of directed quantum chaos (15,16), and chiral random matrix models at nonzero chemical potential (17). The last class of ensembles is obtained from Eqs. (13) and (14) by making the replacement

$$\begin{pmatrix} 0 & C \\ C^{\dagger} & 0 \end{pmatrix} \to \begin{pmatrix} 0 & C + i\mu \\ C^{\dagger} + i\mu & 0 \end{pmatrix} \qquad (14)$$

This chRMT is a model for the *QCD* partition function at nonzero chemical potential μ and will be discussed in more detail later.

Random matrix theory describes the correlations of the eigenvalues of a differential operator. The correlation functions can be derived from the joint probability distribution. The simplest object is the spectral density

$$\rho(\lambda) = \sum_k \delta(\lambda - \lambda_k) \tag{15}$$

The average spectral density, denoted by

$$R_1(\lambda) = \langle \rho(\lambda) \rangle \tag{16}$$

is obtained from the joint probability distribution by integration over all eigenvalues except one. The connected two-point correlation function is defined by

$$\rho_c(\lambda_1, \lambda_2) = \langle \rho(\lambda_1)\rho(\lambda_2) \rangle - \langle \rho(\lambda_1) \rangle \langle \rho(\lambda_2) \rangle \tag{17}$$

In *RMT* it is customary to subtract the diagonal term from the correlation function and to introduce the two-point correlation function $R_2(\lambda_1, \lambda_2)$ defined by

$$R_2(\lambda_1, \lambda_2) = \langle \rho(\lambda_1)\rho(\lambda_2) \rangle - \delta(\lambda_1 - \lambda_2) \langle \rho(\lambda) \rangle \tag{18}$$

and the two-point cluster function

$$T_2(\lambda_1, \lambda_2) = R_1(\lambda_1)R_1(\lambda_2) - R_2(\lambda_1, \lambda_2) \tag{19}$$

In general, the k-point correlation function can be expressed in terms of the joint probability distribution P_N as

$$R_k(\lambda_1, \ldots, \lambda_k) = \frac{N!}{(N-k)!} \int_{-\infty}^{\infty} d\lambda_{k+1} \ldots d\lambda_N P_N(\lambda_1, \ldots, \lambda_N) \tag{20}$$

where we have included a combinatorial factor to account for the fact that spectral correlation functions do not distinguish the ordering of the eigenvalues. Similarly, one can define higher-order connected correlation functions and cluster functions with all lower-order correlations subtracted out. For details we refer to Mehta (5).

Instead of the spectral density, one often studies the resolvent defined by

$$G(z) = \frac{1}{N} \mathrm{Tr} \frac{1}{z - H} = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{z - \lambda_k} \tag{21}$$

which is related to the spectral density by

$$\langle \rho(\lambda) \rangle = -\lim_{\epsilon \to 0} \frac{N}{\pi} \operatorname{Im} \langle G(\lambda + i\epsilon) \rangle \tag{22}$$

In the analysis of spectra of complex systems and the study of random matrix theories, it has been found that the average spectral density is generally not given by the result for the Gaussian random matrix ensembles which have a semicircular shape. What is given by *RMT* are the correlations of the eigenvalues expressed in units of the average level spacing. For this reason, one introduces the cluster function

$$Y_2(r_1, r_2) = \frac{T_2(r_1/R_1(\lambda_1), r_2/R_1(\lambda_2))}{R_1(\lambda_1) R_1(\lambda_2)} \tag{23}$$

In general, correlations of eigenvalues in units of the average level spacing are called microscopic correlations. These are the correlations that can be described by the $N \to \infty$ limit of *RMT*.

The cluster function in Eq. (23) has universal properties. In the limit $N \to \infty$, it is invariant with respect to modifications of the probability distribution of the random matrix ensemble. For example, for the GUE and the chGUE, it has been shown that replacing the Gaussian probability distribution by a distribution given by the exponent of an arbitrary even polynomial results in the same microscopic correlation functions (18,19).

For ensembles in which the eigenvalues occur in pairs $\pm\lambda_k$, an additional important correlation function with universal properties is the microscopic spectral density (20) defined by

$$\rho_s(u) = \lim_{N \to \infty} \frac{1}{\pi \rho(0)} \left\langle \rho\left(\frac{u}{\pi \rho(0)} \right) \right\rangle \tag{24}$$

Related to this observable is the distribution of the smallest eigenvalue, which was shown to be universal as well (21). For this class of ensembles, the point $\lambda = 0$ is a special point. Therefore, all correlation functions near $\lambda = 0$ must be studied separately. However, the microscopic correlations of these ensembles in the bulk of the spectrum are the same as those of the Wigner–Dyson ensemble with the same value of β.

There are two different applications of *RMT*. First, it is used as an exact theory of spectral correlations of a differential operator. As an important application we mention the study of universal properties in transport phenomena in nuclei (14) and disordered mesoscopic systems. In particular, the latter topic has received a great deal of attention recently (see Refs. 6 and 42). This is the original application of *RMT*. Second, it is used as a schematic model for a complex system. One famous example in the second class is the Anderson model (22) for Anderson localization. The properties of this model depend in a critical way on the spatial dimensionality of the lattice. Other examples that are discussed in more detail later are models for the *QCD* partition function at nonzero temperature and nonzero chemical potential.

Random matrix theory eigenvalue correlations are not found in all systems. Obviously, integrable systems, for example a harmonic oscillator, have very different spectral properties. Originally, in the application to nuclear levels, it was believed that the complexity of the system is the main ingredient for the validity of *RMT*. Much later it was realized that the condition for the presence of *RMT* correlations is that the corresponding classical system is completely chaotic. This so-called Bohigas–Giannoni–Schmit conjecture (23) was first shown convincingly for chaotic quantum billiards with two degrees of freedom. By now, this conjecture has been checked for many different systems, and, with some well-understood exceptions, it has been found to be correct. However, a real proof is still absent, and it cannot be excluded that additional conditions may be required for its validity. In particular, the appearance of collective motion in complex many-body systems deserves more attention in this respect.

In general, the average spectral density is not given by *RMT*. Therefore, the standard procedure is to unfold the spectrum (i.e., to rescale the spacing between the eigenvalues according to the local average eigenvalue density). In practice, this unfolding procedure is done as follows. Given a sequence of eigenvalues $\{\lambda_k\}$ with average spectral density $\langle\rho(\lambda)\rangle$, the unfolded sequence is given by

$$\lambda_k^u = \int_{-\infty}^{\lambda_k} \langle\rho(\lambda)\rangle d\lambda \tag{25}$$

The underlying assumption is that the average spectral density and the eigenvalue correlations factorize. The eigenvalue correlations of the unfolded eigenvalues can be investigated by means of suitable statistics. The best-known statistics are the nearest-neighbor spacing distribution $P(S)$, the number variance $\Sigma^2(r)$, and the Δ_3 statistic. The number variance is defined as the variance of the number of eigenvalues in an interval of length r. The Δ_3 statistic is related to the number variance by

$$\Delta_3(L) = \frac{2}{L^4}\int_0^\infty (L^3 - 2L^2 r + r^3)\Sigma^2(r)dr \tag{26}$$

In the analysis of spectra, it is essential to include only eigenstates with the same exact quantum numbers. Spectra with different exact quantum numbers are statistically independent.

The exact analytical expression of the *RMT* result for the nearest-neighbor spacing distribution is rather complicated. However, it is well approximated by the Wigner surmise, which is the spacing distribution for an ensemble of 2×2 matrices. It is given by

$$P(S) = \alpha_\beta S^\beta e^{-b_\beta S^2} \tag{27}$$

where the constants a_β and b_β can be fixed by the conditions that $P(S)$ is normalized to unity and that the average level spacing is one. The level repulsion at short distances is characteristic for interacting systems. For uncorrelated eigenvalues, one finds $P(S) = \exp(-S)$.

Another characteristic feature of *RMT* spectra is the spectral stiffness. This is expressed by the number variance, which, asymptotically for large r, is given by

$$\Sigma^2(r) \sim \frac{2}{\beta\pi^2}\log r \tag{28}$$

This should be contrasted with the result for uncorrelated eigenvalues given by $\Sigma^2(r) = r$.

In the analysis of spectra one often relies on spectral ergodicity, defined as the equivalence of spectral averaging and ensemble averaging. This method cannot be used for the distribution of the smallest eigenvalues, and one must rely on ensemble averaging.

Before proceeding to the discussion of mathematical methods of random matrix theory, a comment about the notations should be made. Different conventions for normalizing the variance of the probability distribution appear in the literature. This simply amounts to a rescaling of the eigenvalues. For example, in the discussion of orthogonal polynomials and the Selberg integral later, the introduction of rescaled eigenvalues such as $\lambda_k\sqrt{N/2}$ or $\lambda_k\sqrt{N}$ simplifies the expressions.

Mathematical Methods I: Hermitian Matrices

Orthogonal Polynomials. One of the oldest and perhaps most widely used methods in *RMT* is based on orthogonal polynomials. A comprehensive presentation of this method is given in Mehta (5). Here, we summarize the most important ingredients, concentrating on the GUE for mathematical simplicity.

We have already seen that the spectral correlation functions can be obtained by integrating the joint probability distribution. The mathematical problem consists in performing these integrations in the limit $N \to \infty$. It is convenient to rescale λ_k and introduce $x_k = \lambda_k \sqrt{N/2}$. The main point of the orthogonal-polynomial method is the observation that the Vandermonde determinant can be rewritten in terms of orthogonal polynomials $p_n(x)$ by adding to a given row appropriate linear combinations of other rows

$$\Delta(\{x\}) = \det\left[(x_j)^{i-1}\right]_{ij} = \text{const} \times \det[p_{i-1}(x_j)]_{ij} \tag{29}$$

Including the Gaussian factor in Eq. (9), this yields

$$\Delta(\{x\}) \prod_{i=1}^{N} e^{-x^2/2} = \text{const} \times \det[\varphi_{i-1}(x_j)]_{i,j=1,\ldots,N} \tag{30}$$

with functions $\varphi_n(x)$ satisfying

$$\int_{-\infty}^{\infty} dx \varphi_m(x)\varphi_n(x) = \int_{-\infty}^{\infty} dx\, e^{-x^2} p_m(x) p_n(x) = \delta_{nm} \tag{31}$$

In this case, the orthogonal polynomials are essentially the Hermite polynomials, and the φ_n are the oscillator wave functions,

$$\varphi_n(x) = \frac{e^{x^2/2}}{\sqrt{2^n n! \sqrt{\pi}}} \left(\frac{d}{-dx}\right)^n e^{-x^2} \tag{32}$$

The integrals in Eq. (20) can now be performed row by row. The k-point functions are then given by determinants of a two-point kernel

$$R_k(x_1, \ldots, x_k) = \det[K_N(x_i, x_j)]_{i,j=1,\ldots,k} \tag{33}$$

The kernel $K_n(x, y)$ is given by

$$K_N(x, y) = \sum_{n=0}^{N-1} \varphi_n(x)\varphi_n(y) \tag{34}$$

which can be evaluated using the Christoffel–Darboux formula. In the large-N limit, the spectral density becomes the famous Wigner semicircle

$$R_1(x) = \lim_{N\to\infty} K_N(x, x) = \frac{1}{\pi}\sqrt{2N - x^2} \tag{35}$$

if $x^2 < 2N$ and zero otherwise. The mean level spacing $D(x) = 1/R_1(x)$ in the bulk of the semicircle is thus proportional to $1/\sqrt{N}$. The R_k are universal if the spacing $|x-y|$ is on the order of the local mean level spacing [i.e., we require $|x-y| = rD(x)$ with r of order unity]. In this limit, we obtain

$$D(x) \lim_{N\to\infty} K_N(x,y) = \frac{\sin(\pi r)}{\pi r} \tag{36}$$

which is the famous sine kernel. The various functions appearing in a typical *RMT* analysis [e.g., $P(s)$, $\Sigma^2(n)$, or $\Delta_3(n)$] can all be expressed in terms of the R_k.

Selberg's Integral. In 1944, Selberg computed an integral that turned out to have significant applications in *RMT* (24). His result (5) reads

$$\begin{aligned} &I(\alpha,\beta,\gamma,n) \\ &\quad = \int_0^1 dx_1 \cdots \int_0^1 dx_n |\Delta(x)|^{2\gamma} \prod_{j=1}^{n} x_j^{\alpha-1}(1-x_j)^{\beta-1} \\ &\quad = \prod_{j=0}^{n-1} \frac{\Gamma(1+\gamma+j\gamma)\Gamma(\alpha+j\gamma)\Gamma(\beta+j\gamma)}{\Gamma(1+\gamma)\Gamma(\alpha+\beta+(n+j-1)\gamma)} \end{aligned} \tag{37}$$

where $\Delta(x)$ is the Vandermonde determinant, n is an integer, and α, β, and γ are complex numbers satisfying Re $\alpha > 0$, Re $\beta > 0$, Re $\gamma > -\min\{1/n, \text{Re}\,\alpha/(n-1), \text{Re}\,\beta/(n-1)\}$. Choosing the parameters in Eq. (37) appropriately, one can derive special forms of Selberg's integral related to specific orthogonal polynomials (5). For example, choosing $x_i = y_i/L$, $\alpha = \beta = aL^2 + 1$ and taking the limit $L \to \infty$, one obtains the integrals of the joint probability density function of the GUE, which are related to Hermite polynomials. Selberg's integral is also very useful in the derivation of spectral sum rules (25).

Aomoto derived the following generalization of Selberg's integral (26)

$$\begin{aligned} &\int_0^1 dx_1 \cdots \int_0^1 dx_n\, x_1 \cdots x_m |\Delta(x)|^{2\gamma} \prod_{j=1}^{n} x_j^{\alpha-1}(1-x_j)^{\beta-1} \\ &\qquad = \prod_{j=1}^{m} \frac{\alpha+(n-j)\gamma}{\alpha+\beta+(2n-j-1)\gamma} I(\alpha,\beta,\gamma,n) \end{aligned} \tag{38}$$

where $1 \le m \le n$. A further extension of Selberg's integral was considered by Kaneko 27, who related it to a system of partial differential equations whose solution can be given in terms of Jack polynomials.

Supersymmetric Method. The supersymmetric method has been applied successfully to problems where the orthogonal polynomial method has failed (14,28,29). It relies on the observation that the average resolvent can be written as

$$\langle G(z)\rangle = \frac{1}{N}\left\langle \text{Tr}\frac{1}{z-H}\right\rangle = \frac{1}{N}\frac{\partial}{\partial J}\bigg|_{J=0} Z(J) \tag{39}$$

where the generating function is defined by

$$Z(J) = \int DHP(H)\frac{\det(z-H+J)}{\det(z-H)} \tag{40}$$

and the integral is over the probability distribution of one of the random matrix ensembles defined earlier. The determinant can be expressed in terms of Gaussian integrals,

$$\frac{\det(z-H+J)}{\det(z-H)} = \int d\{\psi\} \exp\left(\sum_{kl} [i\phi_k^*(z-H)_{kl}\phi_l + i\chi_k^*(z+J-H)_{kl}\chi_l] \right) \tag{41}$$

where the measure is defined by

$$d\{\psi\} = \prod_{j=1}^{N} \frac{d\phi_j d\phi_j^* d\chi_j d\chi_j^*}{2\pi} \tag{42}$$

For convergence, the imaginary part of z must be positive. The integrations over the real and imaginary parts of ϕ_i range over the real axis (the usual commuting, or bosonic variables), whereas χ_i and $\chi_i{}^*$ are Grassmann variables (i.e., anticommuting, or fermionic variables) with integration defined according to the convention that

$$\int d\chi = 0 \quad \text{and} \quad \int \chi d\chi = 1 \tag{43}$$

With this normalization, $Z(0) = 1$.

For simplicity, we consider the GUE [$\beta = 2$ in Eq. (1)], which mathematically is the simplest ensemble. The Gaussian integrals over H can be performed trivially, resulting in the generating function

$$Z(J) = \int d\{\psi\} \exp\left[-\frac{1}{2N} \mathrm{Trg} \begin{pmatrix} \sum_j \phi_j^* \phi_j & \sum_j \chi_j^* \phi_j \\ \sum_j \chi_j \phi_j^* & \sum_j \chi_j^* \chi_j \end{pmatrix}^2 + i \sum_j (\phi_j^* z \phi_j + \chi_j^*(z+J)\chi_j) \right] \tag{44}$$

where the sums over j run from 1 to N. The symbol Trg denotes the graded trace (or supertrace), defined as the difference of the trace of the boson–boson block (upper left) and the trace of the fermion–fermion (lower right) block. For example, in terms of the 2×2 matrix in Eq. (46), $\mathrm{Trg}\sigma = \sigma_{BB} - i\sigma_{FF}$. The quartic terms in ϕ and χ can be expressed as Gaussian integrals by means of a Hubbard–Stratonovitch transformation. This results in

$$Z(J) = \int d\{\psi\} d\sigma \exp\left[-\frac{N}{2} \mathrm{Tr} g\sigma^2 + i \sum_j \begin{pmatrix} \phi_j^* \\ \chi_j^* \end{pmatrix} (\sigma + \zeta) \begin{pmatrix} \phi_j \\ \chi_j \end{pmatrix} \right] \tag{45}$$

where

$$\sigma = \begin{pmatrix} \sigma_{BB} & \sigma_{BF} \\ \sigma_{FB} & i\sigma_{FF} \end{pmatrix} \tag{46}$$

and

$$\zeta = \begin{pmatrix} z & 0 \\ 0 & z+J \end{pmatrix} \tag{47}$$

The variables σ_{BB} and σ_{FF} are commuting (bosonic) variables that range over the full real axis. Both σ_{BF} and σ_{FB} are Grassmann (fermionic) variables.

The integrals over the ϕ and the χ variables are now Gaussian and can be performed trivially. This results in the σ-model

$$Z(J) = \int d\sigma \exp\left[-\frac{N}{2}\mathrm{Trg}\,\sigma^2 + N\mathrm{Trg}\,\log(\sigma+\zeta)\right] \tag{48}$$

By shifting the integration variables according to $\sigma \to \sigma - \zeta$ and carrying out the differentiation with respect to J, one easily finds that

$$\langle G(z)\rangle = \langle z - i\sigma_{FF}\rangle \tag{49}$$

In the large N limit, the expectation value of σ_{FF} follows from a saddle-point analysis. The saddle-point equation for σ_{FF} is given by

$$\sigma_{FF} + iz = 1/\sigma_{FF} \tag{50}$$

resulting in the resolvent

$$\langle G(z)\rangle = \frac{z}{2} - \frac{i}{2}\sqrt{4-z^2} \tag{51}$$

Using the relation in Eq. (22), we find that the average spectral density is a semicircle.

The supersymmetric method can also be used to calculate spectral correlation functions. They follow from the average of the advanced and the retarded resolvent. In that case, we do not have a saddle-point but rather a saddle-point manifold related to the hyperbolic symmetry of the retarded and advanced parts of the generating function. The supersymmetric method provides us with more than alternative derivations of known results. As an example, the analytical result for S-matrix fluctuations at different energies was first derived by means of this method (14).

Alternatively, it is possible to perform the σ integrations by a supersymmetric version of the Itzykson–Zuber integral (30) rather than a saddle-point approximation. The final result is an exact expression for the kernel of the correlation functions. The advantage of this method is that it exploits the determinantal structure of the correlation functions [see Eq. (33)], and all correlations functions are obtained at the same time. Moreover, the results are exact at finite N.

Replica Trick. The replica trick, which was first introduced in the theory of spin glasses (31), is based on the observation that

$$\left\langle Tr\frac{1}{z-H}\right\rangle = \lim_{n\to 0}\frac{1}{n}\frac{\partial}{\partial J}\bigg|_{J=0} Z_n(J) \tag{52}$$

where the generating function is defined by

$$Z_n(J) = \int DHP(H)\det{}^n(z-H+J) \tag{53}$$

The determinant can then be expressed as a Grassmann integral, where the χ-variables now have an additional flavor index

$$\det{}^n(z-H+J) = \int d\{\chi\}\exp\sum_{ij}\chi_i^{f*}(z-H+J)_{ij}\chi_j^f \tag{54}$$

The sum over $_f$ ranges from 1 to n, and the measure is defined by

$$d\{\chi\} = \prod_{f=1}^{n}\prod_{i=1}^{N} d\chi_i^f d\chi_i^{f*} \tag{55}$$

After averaging over the matrix elements of H and a Hubbard–Stratonovitch transformation, one can again proceed to the σ variables. In this case, we have only a σ_{FF} block, which is now an $n \times n$ matrix. The average resolvent then follows by making a saddle-point approximation and taking the replica limit with the same final result as given in Eq. (51).

Because the replica trick relies on an analytical continuation in n, it is not guaranteed to work. Several explicit examples for its failure have been constructed (17,32). In general, it cannot be used to obtain nonperturbative results for eigenvalue correlations on the microscopic scale, which decreases as $1/N$ in the limit $N \to \infty$.

Resolvent Expansion Methods. The Gaussian averages can also be performed easily by expanding the resolvent in a geometric series in $1/z$

$$G(z) = \frac{1}{z} + \frac{1}{N}\mathrm{Tr}\frac{1}{z}H\frac{1}{z} + \frac{1}{N}\mathrm{Tr}\frac{1}{z}H\frac{1}{z}H\frac{1}{z} + \cdots \tag{56}$$

The Gaussian integral over the probability distribution of the matrix elements is given by the sum over all pairwise contractions. For the GUE, a contraction is defined as

$$\langle H_{ij}H_{kl}^{\dagger}\rangle = \frac{1}{N}\delta_{il}\delta_{jk} \tag{57}$$

To the leading order in $1/N$, the contributions are given by the nested contractions. One easily derives that the average resolvent satisfies the equation

$$\langle G(z)\rangle = \frac{1}{z}(1 + \langle G(z)\rangle^2) \tag{58}$$

again resulting in the same expression for the average resolvent.

This method is valid only if the geometric series is convergent. For this reason, the final result is valid only for the region that can be reached from large values of z by analytical continuation. For non-Hermitian matrices, this leads to the failure of the method, and instead one must rely on the so-called Hermitization.

As is the case with the replica trick, this method does not work to obtain nonperturbative results for microscopic spectral correlations. This method has been used widely in the literature. We mention as one of the earlier references the application to the statistical theory of nuclear reactions (33).

Dyson Gas. The formula in Eq. (9) suggests a very powerful analogy between the Wigner–Dyson random matrix ensembles and the statistical properties of a gas of charged particles restricted to move in one dimension, the Dyson gas 3.

Let λ_k be a coordinate of a classical particle that moves in the potential $V_1(\lambda_k) = N\lambda^2{}_k/4$. Furthermore, let two such particles repel each other so that the potential of the pairwise interaction is $V_2(\lambda_k, \lambda_l) = -\ln|\lambda_k - \lambda_l|$. If one considers a gas of N such particles in thermal equilibrium at temperature T, then the probability distribution for the coordinates of the particles $\lambda = \{\lambda_1, \ldots, \lambda_N\}$ will be proportional to $\exp(-V(\lambda)/T)\ \Pi_k d\lambda_k$, where the potential energy V is given by

$$V(\lambda) = \sum_{k<l} V_2(\lambda_k, \lambda_l) + \sum_i V_1(\lambda_i) \tag{59}$$

If the temperature T of the gas is chosen to be equal to $1/\beta$, the probability distribution of the coordinates of the particles becomes identical to the probability distribution in Eq. (9) of the eigenvalues.

This analogy allows one to apply methods of statistical mechanics to calculate distributions and correlations of the eigenvalues (3). It also helps to grasp certain aspects of universality in the statistical properties of the eigenvalues. In particular, it is understandable that the correlations of the relative positions of particles are determined by the interactions between them (i.e., by V_2) and are generally insensitive to the form of the single-particle potential V_1. On the other hand, the overall density will depend on the form of the potential V_1.

The logarithmic potential V_2 is the Coulomb potential in two-dimensional space (i.e., it satisfies the two-dimensional Laplace equation $\Delta V_2 = 0$). Therefore, the Dyson gas can be viewed as a two-dimensional Coulomb gas, with the kinematic restriction that the particles move along a straight line only. This restriction is absent in the case of non-Hermitian matrices.

Mathematical Methods II: Non-Hermitian Matrices

The eigenvalues of non-Hermitian matrices are not constrained to lie on the real axis. Instead, they occupy the two-dimensional complex plane. This fact requires nontrivial modifications of some of the methods developed for Hermitian matrices. Surprisingly or not, the required formalism is sometimes simpler and sheds more light on the properties of Hermitian random matrices.

Orthogonal Polynomials. The method of orthogonal polynomials can also be applied to treat non-Hermitian random matrices. The simplest example is the Ginibre ensemble of arbitrary complex matrices in Eq. (12), with $\beta = 2$ (13). It is convenient to rescale λ_k and introduce $w_k = \lambda_k\sqrt{N}$. The orthogonal polynomials

with respect to the weight given by $\exp(-|w|^2)$ are simply the monomials w^n. Indeed

$$\int dudv e^{-|w|^2} w^n (w^*)^m = \pi n! \delta_{mn} \tag{60}$$

where $w = u + iv$. The orthonormal functions, $f\, dudv \phi_n(w)\phi_m(w*) = \delta_{nm}$, are therefore

$$\phi_n(w) = \frac{1}{\sqrt{\pi n!}} e^{-|w|^2/2} w^n \tag{61}$$

Following the same steps as in the case of the Hermitian GUE, one obtains all correlation functions in the form of the determinant

$$R_k(w_1, \ldots, w_k) = \det[K_N(w_i, w_j)]_{i,j=1,\ldots,k} \tag{62}$$

with a kernel K_N given by

$$K_N(w_1, w_2) = \sum_{n=0}^{N-1} \phi_n(w_1)\phi_m(w_2^*) \tag{63}$$

By a careful analysis of the large-N limit of the kernel, one finds that $R_1(w)$ is $1/\pi$ inside the complex disk $|w| < \sqrt{N}$ and vanishes outside this domain.

Coulomb Gas. The probability distribution in Eq. (13) is the same as for a Coulomb gas in two dimensions placed in the harmonic potential $V_1 = N|z|^2/2 \equiv N(x^2 + y^2)/2$ at a temperature $1/\beta = 1/2$. Unlike in the Hermitian case, the particles of the gas are now allowed to move in both dimensions.

The analogy with the Coulomb gas can be used to calculate the density of eigenvalues of the ensemble of complex non-Hermitian matrices [Eq. (12) with $\beta = 2$] in the limit $N \to \infty$. In this limit, the typical energy per particle, $O(N)$, is infinitely larger than the temperature, 1/2. Therefore, the system is assuming an equilibrium configuration with the minimal energy, as it would at zero temperature. Each particle is subject to a linear force $-dV_1/d|z| = -N|z|$ directed to the origin, $z = 0$. This force must be balanced by the Coulomb forces created by the distribution of all other particles. Thus, the electric field created by this distribution must be directed along the radius and be equal to $|\boldsymbol{E}| = N|z|$. The Gauss law, $\nabla \cdot \boldsymbol{E} = 2\pi\rho$, tells us that such a field is created by charges distributed uniformly (with density $\rho = N/\pi$) inside a circle around $z = 0$, known as the Ginibre circle. The radius of this circle R is fixed by the total number of the particles, $\pi R^2 \rho = N$, so that $R = 1$.

Electrostatic Analogy and Analyticity of Resolvent. In general, the mapping of the random matrix model onto the Coulomb gas is not possible because the pairwise interaction is not always given simply by the logarithm of the distance between the particles. However, a more generic electrostatic analogy exists, relating the two-dimensional density of eigenvalues ρ

$$\rho(x, y) = \sum_k \delta(x - x_k)\delta(y - y_k) \tag{64}$$

where x_k and y_k are real and imaginary parts of λ_k, and the resolvent G

$$G(x, y) = \frac{1}{N} Tr \frac{1}{z - C} = \frac{1}{N} \sum_k \frac{1}{z - \lambda_k} \tag{65}$$

Since the electric field created by a point charge in two dimensions is inversely proportional to the distance from the charge, one can see that the two-component field (NRe G, $-N$Im G) coincides with the electric field $\boldsymbol{E}$, created by the charges located at the points $\{\lambda_1, \ldots, \lambda_N\}$ in the complex plane.

The Gauss law, relating the density of charges and the resulting electric field, $\nabla \cdot \boldsymbol{E} = 2\pi\rho$, gives the following relation between the density of the eigenvalues and the resolvent:

$$\rho = \frac{N}{2\pi}\left\{\frac{\partial \mathrm{Re}\, G}{\partial x} - \frac{\partial \mathrm{Im}\, G}{\partial y}\right\} \tag{66}$$

This relation is the basis of methods for the calculation of the average density of the eigenvalues, $\langle\rho\rangle$. The right-hand side of this equation vanishes if G obeys the Cauchy–Riemann conditions (i.e., if it is an analytic function of the complex variable $z = x + iy$). Conversely, ρ describes the location and the amount of nonanalyticity in G.

In the case of Hermitian matrices, $C = H$, the eigenvalues lie on a line (real axis), and, after ensemble averaging, they fill a continuous interval. This means that the average resolvent $\langle G(x, y)\rangle$ has a cut along this interval on the real axis. The discontinuity along this cut is related to the linear density of the eigenvalues by Eq. (22). In the case of a non- Hermitian matrix C, the eigenvalues may and, in general, do fill two-dimensional regions. In this case, the function G is not analytic in such regions.

This is best illustrated by the ensemble of arbitrary complex matrices C in Eq. (12). In the $N \to \infty$ limit, the resolvent is given by

$$G(x, y) = \begin{cases} z^*, & |z| < 1 \\ 1/z, & |z| > 1 \end{cases} \tag{67}$$

One observes that G is nonanalytic inside the Ginibre circle (13).

Replica Trick. The generalization of the replica trick to the case of non-Hermitian matrices is based on the relation

$$\left\langle \mathrm{Tr}\frac{1}{z - C}\right\rangle = \lim_{n\to 0}\frac{1}{n}\frac{\partial}{\partial z}\ln Z_n(z) \tag{68}$$

where now

$$Z_n(z) = \int DC P(C) |\det(z - C)|^n \tag{69}$$

The absolute value of the determinant can be also written as $\det^{n/2}(z - C)\det^{n/2}(z* - C^\dagger)$. Following Eq. (54), one introduces $n/2$ Grassmann variables χ_i to represent $\det^{n/2}(z - C)$ and another $n/2$ to represent $\det^{n/2}(z* - C^\dagger)$. If the measure $P(C)$ is Gaussian, the integral over C can now be performed, resulting in terms quartic in the Grassmann variables. These can be rewritten with the help of an auxiliary $n \times n$ variable σ as bilinears in χ, after which the χ integration can be done. The resulting integral, in the limit $N \to \infty$, is given by the saddle point (maximum) of its integrand. In the case of the Ginibre ensembles, one arrives at the following expression:

$$\ln Z_n(z) = nN \max_{\sigma}[-|\sigma|^2 + \ln(|z|^2 + |\sigma|^2)] \tag{70}$$

There are two possible maxima, $\sigma = 0$ and $|\sigma|^2 = 1 - |z|^2$, which give two branches for $\ln Z_n$, $\ln Z_n/(nN) = \ln |z|^2$ and $\ln Z_n/(nN) = |z|^2 - 1$. The former dominates when $|z| > 1$, and the latter, when $|z| < 1$. Using Eq. (68), one obtains the average resolvent given by Eq. (67).

It is important that the absolute value of the determinant is taken in Eq. (69). Without taking the absolute value, one would obtain the incorrect result $G = 1/z$ everywhere in the complex plane.

Hermitization. The method of Hermitization, as well as the replica trick in Eqs. (68) and (69), is based on the observation that $\Delta \ln |z|^2 = 4\pi\delta(x)\delta(y)$, where Δ is the Laplacian in the coordinates x and y. One can therefore write for the eigenvalue density $\rho(x,y)$

$$4\pi \langle \rho(x,y) \rangle = \Delta \langle \ln \det(z - C)(z^* - C^\dagger) \rangle \qquad (71)$$

The determinant on the right-hand side can be written as the determinant of a matrix (up to a sign)

$$H(z,z^*) = \begin{pmatrix} 0 & z - C \\ z^* - C^\dagger & 0 \end{pmatrix} \qquad (72)$$

This matrix is Hermitian, and one can apply methods of Hermitian *RMT* (e.g., the supersymmetric method or the replica trick) to determine its resolvent $G(\eta)$. Integrating over η one obtains the quantity

$$\langle \ln \det(\eta - H) \rangle \qquad (73)$$

which in the limit $\eta \to 0$ reduces to the expression on the right-hand side of Eq. (71) (15,34,35,36).

APplications and Advanced Topics

In this section, we briefly review a variety of different subfields of physics where *RMT* has been applied successfully. Most of the examples can be found in the comprehensive presentation of Ref. 6, which also contains a wealth of useful references.

Nuclear Level Spacings. Historically, the first application of *RMT* in physics arose in the study of nuclear energy levels. The problem of computing highly excited energy levels of large nuclei is so complicated that it is impossible to make detailed predictions based on microscopic models. Therefore, as discussed in the introduction, it is interesting to ask whether the statistical fluctuations of the nuclear energy levels are universal and described by the predictions from *RMT*. The nuclear Hamiltonian is time-reversal invariant so that the data should be compared with GOE results. Figure 1 shows the nearest-neighbor spacing distribution of nuclear energy levels of the same spin and parity, averaged over 1726 spacings from 32 different nuclei (37). Clearly, the data are described by *RMT*, indicating that the energy levels are strongly correlated. The parameter-free agreement seen in Fig. 1 gave strong support to the ideas underlying *RMT*.

Hydrogen Atom in a Magnetic Field. The Hamiltonian of this system is given by

$$H = \frac{\mathbf{p}^2}{2m} - \frac{e^2}{r} + \omega L_z + \frac{1}{2} m\omega^2 (x^2 + y^2) \qquad (74)$$

where m is the reduced mass, e is the unit charge, $r = (x^2+y^2+z^2)^{1/2}$ is the separation of proton and electron, $\omega = eB/(2mc)$ is the Larmor frequency, B is a constant magnetic field in the z-direction, and L_z is the third component of the angular momentum. At $B = 0$, the system is integrable. This property is lost when the magnetic field is turned on, and large parts of the classical phase space become chaotic. For an efficient numerical

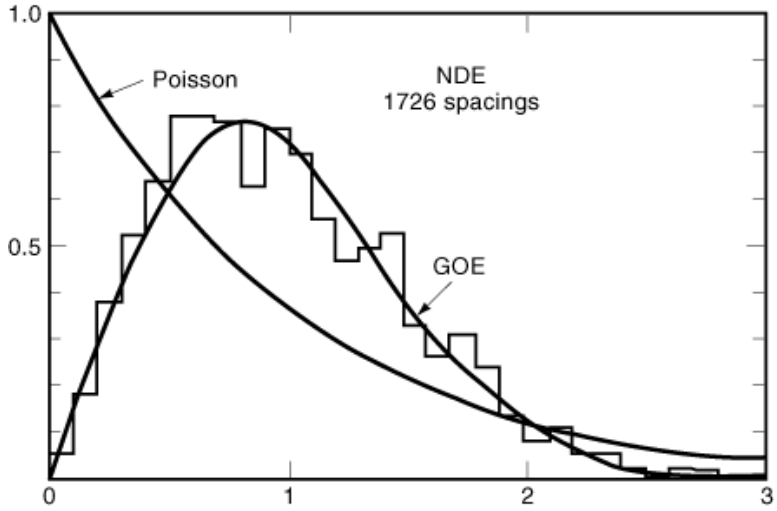


Fig. 1. The histogram represents the nearest-neighbor spacing distribution of the nuclear data ensemble (*NDE*). The curve labeled GOE is the random-matrix prediction, and the Poisson distribution, representing uncorrelated eigenvalues, is shown for comparison. Taken from Ref. 37 with kind permission from Kluwer Academic Publishers.

computation of the eigenvalues, it was important to realize that the Hamiltonian has a scaling property that simplifies the calculations considerably: The spectrum depends only on the combination $\epsilon = \gamma^{-2/3}E$, where γ is a dimensionless variable proportional to B and E is the energy of the system (38,39). This variable increases if the magnetic field is increased and/or the ionization threshold is approached from below. Thus, as a function of ϵ, one should observe a transition from Poisson to *RMT* behavior in the spectral correlations. The numerical results are in agreement with experimental data and clearly show a Poisson to *RMT* transition, see Fig. 2.

Billiards and Quantum Chaos. These are the prototypical systems used in the study of quantum chaos. A billiard is a dynamical system consisting of a point particle that can move around freely in a bounded region of space. For simplicity, we assume that the space is two-dimensional. In a classical billiard, the particle is reflected elastically from the boundaries corresponding to a potential that is zero inside the boundary and infinite outside the boundary. In a quantum billiard, this results in a free particle Schrödinger equation with wave functions that vanish on and outside this boundary. Depending on the shape of the boundary, the classical motion of the particle can be regular or chaotic (or mixed). Examples of classically regular billiards are the rectangle and the ellipse. Important classically chaotic billiards are the stadium billiard (i.e., two semicircles at opposite sides of an open rectangle) and the Sinai billiard (i.e., the region outside a circle but inside a concentric square surrounding the circle). According to the conjecture by Bohigas, Giannoni, and Schmit (23), the level correlations of a quantum billiard whose classical counterpart is chaotic should be given by *RMT*, whereas the eigenvalues of a quantum billiard whose classical analog is regular should be uncorrelated and thus described by a Poisson distribution. This conjecture was investigated—numerically, semiclassically, or using periodic orbit theory—in a number of works and confirmed in almost all cases (40). One can also vary the shape of a billiard as a function of some parameter, thus interpolating between a classically regular and a classically chaotic billiard. As a function of the parameter, one then observes a transition from Poisson to *RMT* behavior in the level correlations of the corresponding quantum billiard.

Quantum Dots. Semiconducting microstructures can be fabricated such that the electrons are confined to a two-dimensional area. If this region is coupled to external leads, we speak of a quantum dot. Such systems have many interesting properties. If the elastic mean free path of the electrons (which at very low temperatures

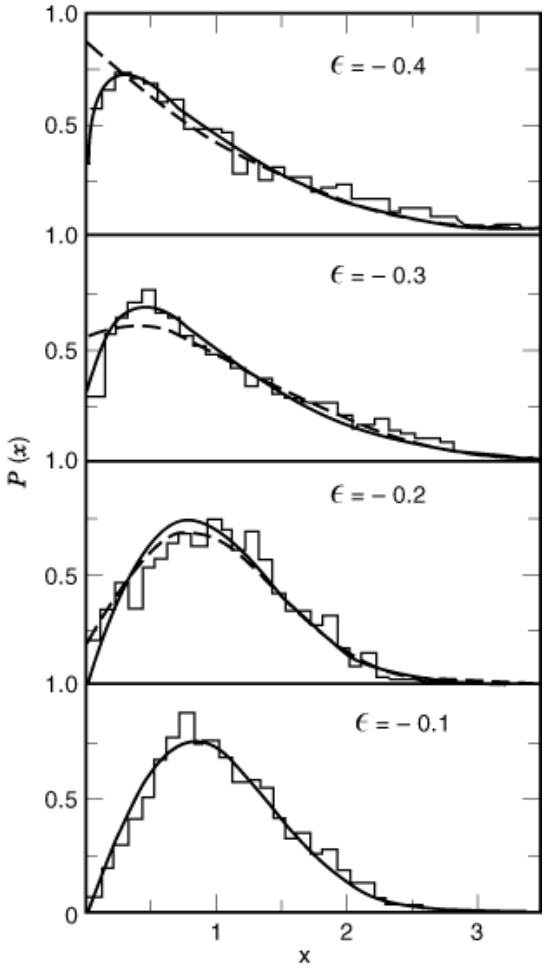


Fig. 2. Nearest-neighbor spacing distribution of the energy levels of the hydrogen atom in a magnetic field (histograms). The solid line in the bottom plot is the *RMT* prediction for the GOE, all other lines are fits. As a function of the scaled variable ϵ, which increases from top to bottom, a transition from Poisson $[P(x) = \exp(-x)]$ to *RMT* behavior is observed. Taken from Ref. 39 with kind permission from Elsevier Science.

is $\gtrsim$10 μm) is larger than the linear dimensions ($\sim$1 μm) of the quantum dot, and if the Coulomb interaction is neglected, the electrons can move around freely inside the boundary, and the quantum dot can be thought of as a realization of a quantum billiard. Depending on the shape of the quantum dot, certain observables (e.g., the fluctuations of the conductance as a function of an external magnetic field) show a qualitatively different behavior. If the shape is classically chaotic (e.g., a stadium), the experimental results agree with predictions from *RMT* as expected, in contrast to data obtained with quantum dots of regular shape where the fluctuations are not universal (41). For a recent review of quantum dots and universal conductance fluctuations to be discussed in the following section, we refer to Ref. 42

Universal Conductance Fluctuations. A mesoscopic system in condensed matter physics is a system whose linear size is larger than the elastic mean free path of the electrons but smaller than the phase coherence length, which is essentially the inelastic mean free path. A typical size is on the order of 1 μm. The conductance

g of mesoscopic samples is closely related to their spectral properties. Using a scaling block picture, Thouless found that in the diffusive regime, $g = E_c/\Delta$, where E_c/h is the inverse diffusion time of an electron through the sample and Δ is the mean level spacing (43). This can be rewritten as $g = \langle N(E_c)\rangle$, where $\langle N(E)\rangle$ is the mean level number in an energy interval E. Thus the variance $\langle \delta g^2\rangle$ of the conductance is linked to the number variance Σ^2 of the energy levels.

In experiments at very low temperatures where the conductance of mesoscopic wires was measured as a function of an external magnetic field, people have observed fluctuations in g on the order of e^2/h, independent of the details of the system (shape, material, etc.). These are the so-called universal conductance fluctuations (44). This phenomenon can be understood qualitatively by estimating the number fluctuations of the electron levels using *RMT* results. However, the magnitude of the effect is much larger than expected, because of complicated quantum interference effects. While a truly quantitative analysis requires linear response theory (the Kubo formula) or the multichannel Landauer formula, both the magnitude of the fluctuations as well their universality can be obtained in a simpler approach using the transfer matrix method. Here, the assumption (although not quantitatively correct) is that certain parameters of the transfer matrix have the same long-range stiffness as in *RMT* spectra.

Anderson Localization. Anderson localization is the phenomenon of a good conductor becoming an insulator when the disorder becomes sufficiently strong. Instead of a description of the electron wave functions by Bloch waves, the wave function of an electron becomes localized and decays exponentially, that is

$$\psi(r) \sim e^{-r/L_c} \tag{75}$$

The length scale L_c is known as the localization length. This phenomenon was first described in the Anderson model (22), which is a hopping model with a random potential on each lattice point. The dimensionality of the lattice plays an important role. It has been shown that in one dimension all states are localized. The critical dimension is two, whereas for $d = 3$ we have a delocalization transition at an energy E_L. All states below E_L are localized whereas all states above E_L are extended (i.e., with a wave function that scales with the size of the system).

The eigenvalues of the localized states are not correlated, and their correlations are described by the Poisson distribution. In the extended domain, the situation is more complicated. An important energy scale is the Thouless energy (43), which is related to the diffusion time of an electron through the sample. With the latter given by L^2/D (the diffusion constant is denoted by D) this results in a Thouless energy given by

$$E_c = \frac{hD}{L^2} \tag{76}$$

Correlations on an energy scale below the Thouless energy are given by random matrix theory, whereas on higher energy scales the eigenvalues show weaker correlations.

Other Wave Equations. So far, we have implicitly considered quantum systems that are governed by the Schrödinger equation. It is an interesting question to ask if the eigenmodes of systems obeying classical wave equations display the same spectral fluctuation properties as predicted by *RMT*. Classical wave equations arise in the study of microwave cavities and in elastomechanics and acoustics.

In three-dimensional microwave cavities, the electric and magnetic fields are determined by the Helmholtz equation, $(\Delta^2 + \mathbf{k}^2)\mathbf{A}(\mathrm{r}) = 0$, where $\mathbf{A} = \mathbf{E}$ or $\mathbf{B}$. It was found experimentally that the spacing of the eigenmodes of the system is of *RMT* type if the cavity has an irregular shape (45). If the cavity has some regular features, the spacing distribution interpolates between *RMT* and Poisson behavior (46).

Elastomechanical eigenmodes have been studied both for aluminum and for quartz blocks. Here, there are two separate Helmholtz equations for the longitudinal (pressure) and transverse (shear) waves, respectively,

making the problem even more different from the Schrödinger equation. Several hundred to about 1500 eigenmodes could be measured experimentally. A rectangular block has a number of global symmetries, and the measured spectrum is a superposition of subspectra belonging to different symmetries. In such a situation, the spacing distribution of the eigenmodes is expected to be of Poisson type, and this was indeed observed experimentally. The symmetry can be broken by cutting off corners of the block, and the resulting shape is essentially a three-dimensional Sinai billiard. Depending on how much material was removed from the corners, a Poisson to *RMT* transition was observed in the spacing distribution of the eigenmodes (47).

Thus, we conclude that *RMT* governs not only the eigenvalue correlations of the Schrödinger equation but also those of rather different wave equations.

Zeros of the Riemann Zeta Function. This is an example from number theory that, at first sight, is not related to the theory of dynamical systems. The Riemann zeta function is defined by $\zeta(z) = \Sigma^{\infty}_{k=1} k^{-z}$ for Re $z > 1$. Its nontrivial zeros z_N are conjectured to have a real part of 1/2 (i.e., $z_n = 1/2 + i\gamma_n$). An interesting question is how the γ_n are distributed on the real axis. To this end, it was argued that the two-point correlation function of the γ_n has the form $Y_2(r) = 1 - [\sin(\pi r)/(\pi r)]^2$ (48). This is identical to the result obtained for the unitary ensemble of *RMT* and consistent with a conjecture (apparently by Polya and Hilbert) according to which the the zeros of $\zeta(z)$ are related to the eigenvalues of a complex Hermitian operator. By computing the γ_n numerically to order 10^{20} (49), it was shown that their distribution indeed follows the *RMT* prediction for the unitary ensemble (for large enough γ_N).

Universal Eigenvalue Fluctuations in Quantum Chromodynamics. Quantum chromodynamics (*QCD*) is the theory of the strong interactions, describing the interaction of quarks and gluons, which are the basic constituents of hadrons. Quantum chromodynamics is a highly complex and nonlinear theory for which most nonperturbative results have been obtained numerically in lattice *QCD* using the world's fastest supercomputers. The Euclidean *QCD* partition function is given by

$$Z(m) = \int DA \det{}^{N_f}(D + m) e^{-S_{YM}(A)/\hbar} \qquad (77)$$

where S_{YM} is the Euclidean Yang–Mills action and the path integral is over all $SU(N_c)$ valued Yang–Mills fields $A^{ij}{}_{\mu}$ (μ is the Lorentz index, N_c the number of colors, and N_f the number of quark flavors). The Euclidean Dirac operator is defined by $D = \gamma_\mu \partial_\mu + ig\gamma_\mu A_\mu$, where g is the coupling constant and γ_μ are the Euclidean gamma matrices. Because of the chiral symmetry of *QCD*, in a chiral basis the matrix of D has the block structure

$$iD = \begin{pmatrix} 0 & T \\ T^{\dagger} & 0 \end{pmatrix} \qquad (78)$$

In a lattice formulation, the dimension of the matrix T is a multiple of the total number of lattice points. The smallest eigenvalues of the Dirac operator play an important role in the *QCD* partition function. In particular, the order parameter of the chiral phase transition is given by

$$\sum = \lim_{m \to 0} \lim_{V \to \infty} \frac{\pi \langle \rho(0) \rangle}{V} \qquad (79)$$

where $\langle \rho(\lambda) \rangle$ is the average spectral density of the Dirac operator and V is the volume of space-time.

Although the *QCD* partition function can be calculated only numerically, in certain domains of the parameter space it is possible to construct effective theories that can be solved analytically. An important ingredient is the chiral symmetry of the *QCD* Lagrangian, which is broken spontaneously in the ground state. Considering Euclidean *QCD* in a finite volume, the low-energy behavior of the theory can be described in terms of

an effective chiral Lagrangian if the linear length L of the box is much larger than the inverse of a typical hadronic scale. Furthermore, if L is smaller than the inverse of the mass of the pion, which is the Goldstone boson of chiral symmetry breaking, then the kinetic terms in the chiral Lagrangian can be neglected. It was found by Leutwyler and Smilga that the existence of this effective partition function imposes constraints on the eigenvalues of the *QCD* Dirac operator (50). However, in order to derive the full spectrum of the Dirac operator, one needs a different effective theory defined by the partially quenched chiral Lagrangian, which in addition to the usual quarks includes a valence quark and its superpartner (51). As is the case with the usual chiral Lagrangian, the kinetic terms of this Lagrangian can be neglected if the inverse mass of the Goldstone bosons corresponding to the valence quark mass is much larger than the size of the box. It has been shown that in this domain the corresponding spectral correlators are given by the chiral ensembles that have the same block structure as the Dirac operator in Eq. (78). The β-value of the ensemble is determined by N_c and the representation of the fermions. The energy scale for the validity of chiral *RMT* is the equivalent of the Thouless energy and is given by $F^2/\Sigma L^2$, where F is the pion decay constant that enters in the chiral Lagrangian.

The fluctuation properties of the Dirac eigenvalues can be studied directly by diagonalizing the lattice *QCD* Dirac operator. Correlations in the bulk of the spectrum agree perfectly with the various *RMT* results (52). However, as was already pointed out, the small Dirac eigenvalues are physically much more interesting. Because of the relation in Eq. (79), the spacing of the low-lying eigenvalues goes like $1/(V\Sigma)$. To resolve individual eigenvalues, one must magnify the energy scale by a factor of $V\Sigma$ and consider the microscopic spectral density (20) defined in Eq. (24).

Because of the chiral structure of the Dirac operator in Eq. (78), all nonzero eigenvalues of iD come in pairs $\pm\lambda_n$, leading to level repulsion at zero. This is reflected in the fact that $\rho_s(0) = 0$ even though $\lim_{\lambda\to 0} \lim_{V\to\infty} \rho(\lambda)/V > 0$. The spectrum is said to have a "hard edge" at $\lambda = 0$.

The result for $\rho_s(z)$ for the chGUE (appropriate for *QCD* with three or more colors) and gauge fields with topological charge ν reads (7,53)

$$\rho_s(z) = \frac{z}{2}\left[J^2_{N_{f+|\nu|}}(z) - J_{N_{f+|\nu|+1}}(z)J_{N_{f+|\nu|-1}}(z)\right] \qquad (80)$$

where J denotes the Bessel function. The results for the chGSE and the chGOE are more complicated. Lattice *QCD* data agree with *RMT* predictions as seen in Fig. 3, which represents results corresponding to the chGSE.

***QCD* at Nonzero Temperature and Chemical Potential.** Random-matrix models can also be used to model and analyze generic properties of the chiral symmetry restoration phase transition at finite temperature or finite baryon chemical potential μ. For example, the effect of the chemical potential can be described by the non-Hermitian deformation in Eq. (14) of the chGUE. The eigenvalues of such a matrix are not constrained to lie on the real axis. The quantity that signals chiral symmetry breaking is the discontinuity (a cut) of the averaged resolvent $\langle G(z)\rangle$ at $z = 0$. One can calculate $\langle G(z)\rangle$ in a theory with $n \neq 0$, which corresponds to *QCD* with N species of quarks. There is a critical value of μ above which $\langle G(z)\rangle$ becomes continuous at $z = 0$, and, therefore, chiral symmetry is restored. In lattice Monte Carlo, the problem of calculating the partition function and expectation values such as $\langle G(z)\rangle$ at finite μ, which are of a paramount interest to experiment, is still unresolved. The difficulty lies in the fact that the determinant of the Dirac matrix is complex and cannot be used as part of the probabilistic measure to generate configurations using the Monte Carlo method. For this reason, exploratory simulations at finite μ have been done only in the quenched approximation in which the fermion determinant is ignored, $n = 0$. The results of such simulations were in puzzling contradiction with physical expectations: The transition to restoration of chiral symmetry occurs at $\mu = 0$ in the quenched approximation.

The chiral random matrix model at $\mu \neq 0$ allows for a clean analytical explanation of this behavior, since one can calculate $\langle G(z)\rangle$ both at $n = 0$ and $n \neq 0$. (As before, the number of replicas is denoted by n.) The behavior

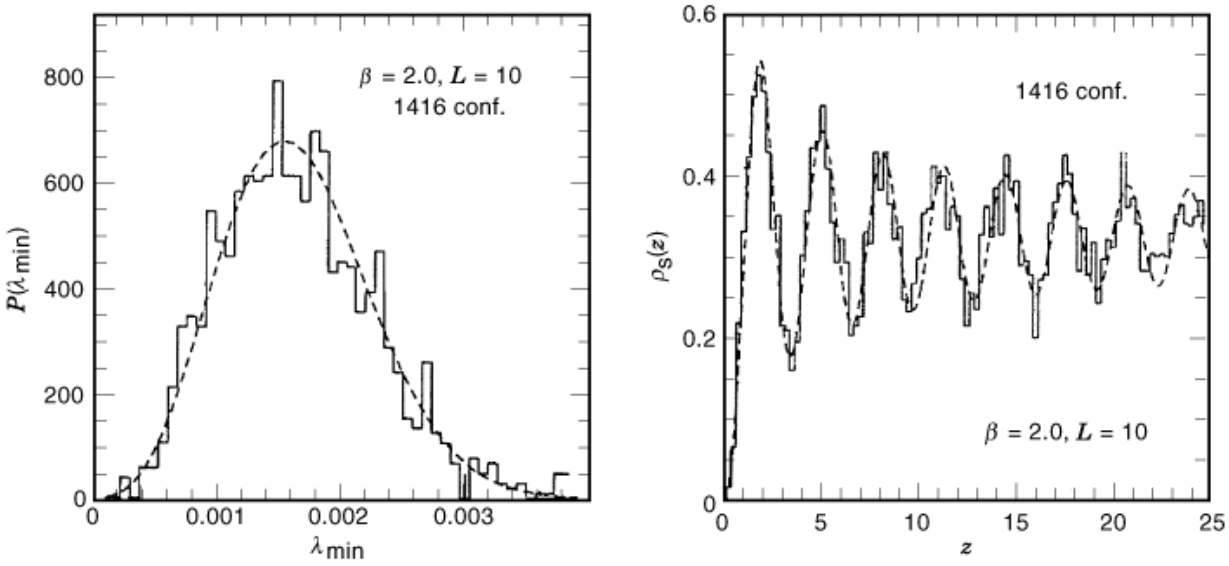


Fig. 3. Distribution of the smallest eigenvalue (left) and microscopic spectral density (right) of the *QCD* Dirac operator. The histograms represent lattice data in quenched SU (2) with staggered fermions on a 10^4 lattice using $\beta = 4/g^2 = 2.0$ (not to be confused with the Dyson index β). The dashed curves are the parameter-free *RMT* predictions. Taken from Ref. 54 with kind permission from the American Physical Society.

of $\langle G(z)\rangle$ at $n = 0$ and $n \neq 0$ is drastically different. While at $n \neq 0$, the nonanalyticities of $\langle G(z)\rangle$ come in the form of one-dimensional cuts, for $n = 0$ they form two-dimensional regions, similar to the Ginibre circle in the case of the non-Hermitian GUE. This means that the $n = 0$ (quenched) theory is not a good approximation to the $n \neq 0$ (full) theory at finite μ, when the Dirac operator is non-Hermitian. The quenched theory is an approximation (or, the $n \to 0$ limit) of a theory with the determinant of the Dirac operator replaced by its absolute value, which has different properties at finite μ (17).

Quantum Gravity in Two Dimensions. In all cases we have discussed so far, the random-matrix model was constructed for the Hamiltonian (or a similar operator) of the system, and the universal properties were independent of the distribution of the random matrix. In contrast, in quantum gravity the elementary fields are replaced by matrices, and the details of the matrix potential do influence the results. For a recent review we refer to Ref. 55

Two dimensional quantum gravity is closely related to string theory. The elementary degrees of freedom are the positions of the string in d dimensions. The action S of the theory involves kinematic terms and the metric. The partition function z is then given as a path integral of $\exp(-S)$ over all possible positions and metrics. The string sweeps out two-dimensional surfaces, and Z can be computed in a so-called genus expansion, (i.e., as a sum over all possible topologies of these surfaces). This is typically done by discretizing the surfaces. One can then construct dual graphs by connecting the centers of adjacent polygons (with n sides). These dual graphs turn out to be the Feynman diagrams of a ϕ^n-theory in zero dimensions which can be reformulated in terms of a matrix model. The partition function of this model is given by

$$Z = \int DM e^{-N\mathrm{tr}v(M)} \quad \text{with} \quad v(M) = \sum_{n>1} g_n M^n \qquad (81)$$

where the M are Hermitian matrices of dimension n and the g_N are coupling constants involving appropriate powers of the cosmological constant. The mathematical methods used to deal with the matrix model of quantum gravity are closely related to those employed in *RMT*, giving rise to a useful interchange between the two areas.

BIBLIOGRAPHY

1. J. Wishart The generalized product moment distribution in samples from a normal multivariate population, *Biometrika*, **20**: 32, 1928.
2. E. P. Wigner Characteristic vectors of bordered matrices with infinite dimensions, *Ann. Math.*, **62**: 548, 1955.
3. F. J. Dyson, , a) Statistical theory of the energy levels in complex systems: I, b) Statistical theory of the energy levels in complex systems: II, c) Statistical theory of the energy levels in complex systems: III, d) A Brownian-Motion model for the eigenvalues of a random matrix, e) The threefold way. Algebraic structure of symmetry groups and ensembles in quantum mechanics, *J. Math. Phys.*, **3**: 140, 157, 166, 1191, 1199, 1962.
4. C. E. Porter *Statistical Theory of Spectra: Fluctuations*, New York: Academic Press, 1965.
5. M. L. Mehta *Random Matrices*, 2nd ed., San Diego: Academic Press, 1991.
6. T. Guhr A. Müller-Groeling, H. A. Weidenmüller Random matrix theories in quantum physics: Common concepts, *Phys. Rept.*, **299**: 189, 1998.
7. J. J. M. Verbaarschot The spectrum of the QCD Dirac operator and chiral random matrix theory: The threefold way, *Phys. Rev. Lett.*, **72**: 2531, 1994.
8. R. Gade Anderson localization for sublattice models, *Nucl. Phys. B*, **398**: 499, 1993.
9. A. Altland M. R. Zirnbauer Random matrix theory of a chaotic andreev quantum dot, *Phys. Rev. Lett.*, **76**: 3420, 1996.
10. M. R. Zirnbauer Riemannian symmetric superspaces and their origin in random-matrix theory, *J. Math. Phys.*, **37**: 4986, 1996.
11. R. Oppermann Anderson localization problems in gapless superconducting phases, *Physica A*, **167**: 301, 1990.
12. L. K. Hua *Harmonic Analysis*, American Mathematical Society, Providence, RI, 1963.
13. J. Ginibre Statistical ensembles of complex, quaternion, and real matrices, *J. Math. Phys*, **6**: 440, 1965.
14. J. J. M. Verbaarschot H. A. Weidenmüller M. R. Zirnbauer Grassmann integration in stochastic quantum physics: The case of compound-nucleus scattering, *Phys. Rept.*, **129**: 367, 1985.
15. K. B. Efetov, a) Directed Quantum Chaos, *Phys. Rev. Lett.*, **79**: 491, 1997; b) Quantum Disordered Systems with a Direction, *Phys. Rev. B*, **56**: 9630, 1997.
16. Y. V. Fyodorov B. A. Khoruzhenko H. J. Sommers Almost-Hermitian random matrices: Crossover from Wigner-Dyson to Ginibre eigenvalue statistics, *Phys. Rev. Lett.*, **79**: 557, 1997.
17. M. A. Stephanov Random matrix model of QCD at finite density and the nature of the quenched limit, *Phys. Rev. Lett.*, **76**: 4472, 1996.
18. G. Hackenbroich H. A. Weidenmüller , Universality of random-matrix results for non-Gaussian ensembles, *Phys. Rev. Lett.*, **74**: 4118, 1995.
19. G. Akemann, *et al.* Universality of random matrices in the microscopic limit and the Dirac operator spectrum, *Nucl. Phys. B*, **487**: 721, 1997.
20. E. V. Shuryak J. J. M. Verbaarschot Random matrix theory and spectral sum rules for the Dirac operator in QCD, *Nucl. Phys. A*, **560**: 306, 1993.
21. S. M. Nishigaki P. H. Damgaard T. Wettig Smallest Dirac eigenvalue distribution from random matrix theory, *Phys. Rev. D*, **58**: 087704, 1998.
22. P. W. Anderson Absence of diffusion in certain random lattices, *Phys. Rev.*, **109**: 1492, 1958.
23. O. Bohigas M.J. Giannoni C. Schmit Characterization of chaotic quantum spectra and universality of level fluctuation laws, *Phys. Rev. Lett.*, **52**: 1, 1984.

24. A. Selberg Bermerkninger om et multiplet integral, *Norsk Mat. Tidsskr.*, **26**: 71, 1944.
25. J. J. M. Verbaarschot Spectral sum rules and Selberg's integral formula, *Phys. Lett. B*, **329**: 351, 1994.
26. K. Aomoto Jacobi polynomials associated with Selberg integrals, *SIAM J. Math. Anal.*, **18**: 545, 1987.
27. J. Kaneko Selberg integrals and hypergeometric functions associated with Jack polynomials, *SIAM J. Math. Anal.*, **24**: 1086, 1993.
28. K. Efetov Supersymmetry and theory of disordered metals, *Adv. Phys.*, **32**: 53, 1983.
29. K. Efetov *Supersymmetry in Disorder and Chaos*, Cambridge: Cambridge Univ. Press, 1997.
30. T. Guhr a) Dyson's correlation functions and graded symmetry, b) An Itzykson-Zuber-like integral and diffusion for complex ordinary and supermatrices, *J. Math. Phys.*, **32**: 336, 1991; T. Guhr and T. Wettig, *J. Math. Phys.*, **37**: 6395, 1996.
31. S. F. Edwards P. W. Anderson Theory of spin glasses, *J. Phys. F: Met. Phys.*, **5**: 965, 1975.
32. J. J. M. Verbaarschot M. R. Zirnbauer Critique of the replica trick, *J. Phys. A: Math. Gen.*, **17**: 1093, 1985.
33. D. Agassi H. A. Weidenmüller G. Mantzouranis The statistical theory of nuclear reactions for strongly overlapping resonances as a theory of transport phenomena, *Phys. Rept.*, **22**: 145, 1975.
34. V. L. Girko *Theory of Random Determinants*, Dordrecht: Kluwer, 1990.
35. H. J. Sommers et al., Spectrum of large random asymmetric matrices, *Phys. Rev. Lett.*, **60**: 1895, 1988.
36. J. Feinberg A. Zee Non-hermitian random matrix theory: Method of hermitian reduction, *Nucl. Phys. B*, **504**: 579, 1997.
37. O. Bohigas R. U. Haq A. Pandey K. H. Bochhoff (ed.), *Nuclear Data for Science and Technology*, Dordrecht: Reidel, p. 809, 1983.
38. D. Wintgen a) Connection between long range correlations in quantum spectra and classical periodic orbits, *Phys. Rev. Lett.*, **58**: 1589, 1987; D. Wintgen and H. Friedrich, b) Classical and quantum mechanical transition between regularity and irregularity in a Hamiltonian system, *Phys. Rev. A*, **35**: 1464, 1987.
39. H. Friedrich D. Wintgen The Hydrogen atom in a uniform magnetic field—an example of chaos, *Phys. Rept.*, **183**: 37, 1989.
40. M. C. Gutzwiller *Chaos in Classical and Quantum Mechanics*, New York: Springer, 1990.
41. C. M. Marcus et al., Conductance fluctuations and chaotic scattering in ballistic microstructures, *Phys. Rev. Lett*, **69**: 506, 1992.
42. C. W. J. Beenakker Random matrix theory of quantum transport, *Rev. Mod. Phys.*, **69**: 731, 1997.
43. D. J. Thouless Electrons in disordered systems and the theory of localization, *Phys. Rept.*, **13**: 93, 1974.
44. S. Washburn R. A. Webb Aharonov-Bohm effect in normal metal quantum coherence and transport, *Adv. Phys.*, **35**: 375, 1986.
45. S. Deus P. M. Koch L. Sirko Statistical properties of the eigenfrequency distribution of 3-dimensional microwave cavities, *Phys. Rev. E*, **52**: 1146, 1995.
46. H. Alt et al., Chaotic dynamics in a three-dimensional superconducting microwave billiard, *Phys. Rev. E*, **54**: 2303, 1996.
47. C. Ellegaard et al., Spectral statistics of acoustic resonances in aluminum blocks, *Phys. Rev. Lett.*, **75**: 1546, 1995.
48. H. L. Montgomery The pair correlations of zeros of the zeta function, *Proc. Symp. Pure Maths.*, **24**: 181, 1973.
49. A. M. Odlyzko On the distribution of spacings between zeros of the zeta function, *Math. Comput.*, **48**: 273; 1987 [online]. Available WNN: http://www.research.att.com/amo/unpublished/zeta. 10to20.1992.ps
50. H. Leutwyler A. V. Smilga Spectrum of Dirac operator and role of winding number in QCD, *Phys. Rev. D*, **46**: 5607, 1992.
51. J. C. Osborn D. Toublan J. J. M. Verbaarschot From chiral random matrix theory to chiral perturbation theory, *Nucl. Phys. B*, **540**: 317, 1999.
52. M. A. Halasz J. J. M. Verbaarschot a) Universal fluctuations in spectra of the lattice Dirac operator, *Phys. Rev. Lett.*, **74**: 3920, 1995, R. Pullirsch et al., b) Evidence for quantum chaos in the plasma phase of QCD, *Phys. Lett. B*, **427**: 119, 1998.
53. J. J. M. Verbaarschot I. Zahed Spectral density of the QCD Dirac operator near zero virtuality, *Phys. Rev. Lett.*, **70**: 3852, 1993.

54. M. E. Berbenni-Bitsch et al., Microscopic universality in the spectrum of the lattice Dirac operator, *Phys. Rev. Lett.*, **80**: 1146, 1998.
55. P. Di Francesco P. Ginsparg J. Zinn-Justin Gravity and random matrices, *Phys. Rept.*, **254**: 1, 1995.

M. A. STEPHANOV
State University of New York at Stony Brook
J. J. M. VERBAARSCHOT
Yale University
T. WETTIG
State University of New York at Stony Brook

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Roundoff Errors
Standard Article
Shantanu Dutt[1] and Daniel Boley[2]
[1]University of Illinois at Chicago, Chicago, IL
[2]University of Minnesota, Minneapolis, MN
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2449
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (185K)

Abstract

The sections in this article are

Representation of Floating-Point Numbers

Catastrophic Effects of Round-Off Error

Effect on Algorithms

Synopsis of Fault Tolerance Techniques for Linear Algebra

Analysis of Error Propagation

Integer Checksums for Floating-Point Computations

| | |

Copyright © 1999-2008 All Rights Reserved.

ROUNDOFF ERRORS

Rounding errors are the errors arising from the use of floating-point arithmetic on digital computers. Since the computer word has only a fixed and finite number of bits or digits, only a finite number of real numbers can be represented on a computer, and the collection of those real numbers that can be represented on the computer is called the *floating-point system* for that computer. Since only finitely many real numbers can be represented exactly, it is possible, indeed likely, that the exact solution to any particular problem is not part of the floating point system and hence cannot be represented exactly. Ideally, one would hope that one could obtain the *representable number* closest to the true exact answer. With simple computations this is usually possible, but is more problematic after long or complicated computations. Even the four basic operations, addition, subtraction, multiplication, and division, cannot be carried out exactly, so the intermediate results in any computation will suffer from contamination of *rounding errors,* and the final results will suffer from the accumulated effects of all the intermediate rounding errors. The field of *numerical analysis* is the study of the behavior of various algorithms when implemented in the floating-point system subject to rounding errors. In this article, we describe the main features typically found in floating-point systems in computers today, give some examples of unusual effects that are caused by the presence of rounding errors, and discuss techniques developed to perform accurate fault tolerance in the presence of these errors.

REPRESENTATION OF FLOATING-POINT NUMBERS

Mantissa Plus Exponent

All computers today represent floating-point numbers in the form mantissa $\times$ base$^{\text{exponent}}$, where the mantissa is typically a number less than 2 in absolute value, and the exponent is a small integer. The base is fixed for all numbers and hence is not actually stored at all. Except for hand-held calculators, the base is usually 2 except for a few older computers where the base is 8 or 16. The mantissa and exponent are represented in binary with a fixed number of bits for each. Hence a typical representation is

$$[s \quad e_7 \quad e_6 \ldots e_0 \quad m_{23} \quad m_{22} \ldots m_1 \quad m_0] \tag{1}$$

where s is the sign bit for the mantissa, e_7, . . ., e_0 are the bits for the exponent, and m_{23}, . . ., m_0 are the bits for the mantissa. If the base is fixed at 2, then the number represented by the bits in Eq. (1) is

$$(-1)^s \times (m_{23} \cdot 2^0 + m_{22} \cdot 2^{-1} + \cdots + m_0 \cdot 2^{-23}) \times 2^{\text{exponent}} \tag{2}$$

where the exponent is an 8-bit signed integer. In this example, we have fixed the number of bits for the mantissa and the exponent to 24 and 8, respectively, but in general these vary from computer to computer, and even within the computer vary from *single* to *double precision.* Notice that the mantissa represented in Eq. (2) has the "binary point" (analog to the usual decimal point) right after the leftmost digit. Regarding the exponent as a signed integer, it is not typically represented as a ones or twos complement number but more often in *excess 127* notation, which is essentially an unsigned integer representing the number 127 larger than the true exponent. Again, if we have k bits instead of 8 as in this example, then the 127 is replaced by $2^{k-1} - 1$.

We illustrate this with a few examples, where we shorten the mantissa to 7 bits plus a sign and the exponent to 4 bits. Hence the exponent is in *excess 7* notation:

decimal	binary	bits	remarks
$+5/2$	$+1.01 \times 2^1$	0 1000 *1*010000	
$-5/2$	-1.01×2^1	1 1000 *1*010000	
$+20$	$+1.01 \times 2^4$	0 1011 *1*010000	
$1/3$	$+1.010101 \times 2^{-2}$	0 0101 *1*010101	inexact
$1/10$	$+1.100110 \times 2^{-3}$	0 0100 *1*100110	inexact

(3)

We remark that this representation, using normalized mantissas and excess notation for the exponents, allows one to compare two positive floating-point numbers using the usual integer compare instructions on the bit patterns.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

Normalization

Notice that in Eq. (3) there can be multiple ways to represent any particular decimal number. If the leading digit of the mantissa is zero, or more generally, if the digit(s) of the mantissa to the left of its binary point do not represent 1 (e.g., 0.*xxx*. . ., 10.*xxx*. . .), then the number is said to be *unnormalized,* otherwise it is said to be *normalized.* So we could also use the representation

decimal	binary	bits	remarks
+20	$+0.00101 \times 2^{7}$	0 1110 0001010	unnormalized
1/3	$+0.101010 \times 2^{-1}$	0 0110 0101010	inexact and unnormalized

(4)

When the number is unnormalized, we lose space for significant digits; hence floating-point numbers are always stored in normalized fashion. We see that in Eq. (3), the normalized representation for the number 1/3 captures more nonzero bits than the unnormalized representation in Eq. (4). When the base is equal to 2, then the leading digit of the mantissa is just a bit whose only possible nonzero value is 1, and hence it is not even stored. So in the representation in Eq. (1), the bit m_{23} is always 1 and is not actually stored in the computer. When not stored in this way, the bit m_{23} is called an *implicit bit.* These bits are written in *italics* in Eq. (3).

Special Numbers, Overflow, Underflow

The representation in Eq. (1) with the implicit bit m_{23} does not admit the number 0, since 0 would have an all-zero mantissa that must be unnormalized. To accommodate this, certain special bit patterns are reserved to zero and certain other special "numbers." A zero is often represented by a word of all zero bits, which would otherwise represent the smallest representable positive floating-point number. If a calculation gives rise to an answer less than the smallest representable number (in absolute value), then an *underflow* condition is said to exist. In the past, the result was simply set to zero, but in the recent IEEE standard, the result is denormalized.

The use of gradually denormalized numbers involves those floating-point numbers which are less (in absolute value) than the smallest representable normalized number. As discussed in Ref. (1), there is a relatively big gap between the smallest representable normalized number and zero. To fill this gap, the IEEE decided to allow for the use of unnormalized numbers. We can illustrate this with the representation in Eq. (3). The smallest normalized number representable in Eq. (3) is $+1.00_{\text{binary}} \times 2^{-7}$. However, we can represent smaller numbers in an unnormalized manner, such as $+0.10_{\text{binary}} \times 2^{-7}$. Since we have adopted the convention of using the implicit bit, such an unnormalized number cannot be encoded in this format. The solution is to provide that the smallest representable normalized number be actually $+1.00_{\text{binary}} \times 2^{-6}$, reserving the smallest possible exponent value for unnormalized numbers. This was adopted in the IEEE standard (see below). Since this smallest exponent value has all its bits equal to zero, the representation of the number zero in this format becomes just a special case of such unnormalized numbers. As pointed out by Goldberg (1), the use of denormalized numbers also guarantees that the computed difference of two unequal numbers will never be zero.

A more serious problem occurs if the result of the calculation is larger than the largest representable number. This is called an *overflow* condition, and in most older computers, this would generate an error. However, in the recent IEEE floating point standard (discussed below), such a result would be replaced with a special bit pattern representing *plus-infinity* or *minus-infinity*. When two such infinities are combined, the result can be totally undefined, so yet another special bit pattern is reserved for such a result. This last result is called *Not A Number,* and is often printed by most computer systems as `NaN`. By not generating an exception upon overflow, programs may fail more gracefully.

Rounding versus Chopping

Another issue affecting rounding errors is the choice of rounding strategy. Given any particular real number, which nearby floating point number should one use? For example, in Eq. (3), when we represented 1/3 as an unnormalized number, we chopped away the last bit, but an alternative choice would be to round up to the next higher number to yield $+0.101011_{\text{binary}} \times 2^{-1}$. The error committed in chopping in this case is .0052, but in rounding is only .0026. But rounding requires slightly more computation since the digits being removed must be examined. This issue arises when converting a number from an external decimal representation and when trying to fit the result of an intermediate arithmetic operation into a memory word. This is because the arithmetic logic units on most computers actually operate on more digits than can fit in a word, the extra digits being called *guard digits,* discussed below.

The IEEE standard actually provides that the default rounding strategy should be a "round-to-even" strategy. The round-to-even mode is exactly the rounding strategy described above, except when the number being rounded lies exactly half-way between two representable numbers, as in rounding 12.5 to an integer. The default round-to-even strategy selects the representable number whose last digit is even, so that 12.5 would round to 12 and not 13. If the rounding in this case were always up, then more numbers would end up being increased than decreased during the rounding process, on average. If the combinations of trailing digits occur equally likely, it is generally desirable that the number of times the rounding is up is about equal to the number of times the rounding is down, to try to cancel out this bias as much as possible.

Guard Digits

Guard digits are extra digits kept only within the Arithmetic Logic Unit (ALU) during the course of individual floating-point operations. They are never stored in memory. The ALU carries out the operation using at least one extra guard digit, then the result is rounded to fit in the register of a memory word. We illustrate the effect of guard digits using the simple addition of two decimal floating point-numbers, $1.01 \times 10^{+1}$ and -9.93×10^{0} (this example is from Ref. 1), where we keep 3 decimal digits in the mantissa. To accomplish this, the first step for the ALU is to shift the decimal point in the second operand to make the exponents match, yielding $-.993 \times 10^{+1}$. Then the mantissas may be added together directly. The accuracy of the answer is greatly affected by the number of digits kept for the computation. The simplest approach is to

use simple chopping and to keep only the digits corresponding to the larger operand. The result in this case is $1.01 \times 10^{+1} - 0.99 \times 10^{+1} = 2.00 \times 10^{-1}$. If, however, we keep at least one extra guard digit, then we obtain $1.010 \times 10^{+1} - 0.993 \times 10^{+1} = 1.70 \times 10^{-1}$. The latter answer is exact, whereas the former result has no correct digits.

The reader may ask whether keeping just one guard digit suffices to make a significant enhancement to the accuracy of floating-point arithmetic operations. The answer can be found in Ref. (1), in which it is proved that if no guard digit is kept during additions, then the error could be so large as to yield no correct digits in the answer, whereas if just one guard digit is kept during the operation, the result being rounded to fit in the memory word, then the error will be at most the equivalent of 2 units in the last significant digit. In this context, the "correct answer" is regarded as the answer computed using all available digits and keeping "infinite precision" for the intermediate results.

IEEE Standard

The previous discussion has shown that there are many choices to be made in representing floating-point numbers, and in the past different manufacturers have made different, incompatible, choices. The result is that the behavior of floating-point algorithms can vary from computer to computer, even if the precision (number of bits used for exponent and mantissa) stays the same. In an attempt to make the behavior of algorithms more uniform across platforms, as well as to improve the performance of such algorithms, the IEEE has established a *floating-point standard* which specified some of these choices (2,3). This standard specifies the kind of rounding that must be used, the use of guard digits, the behavior when underflow or overflow occurs, etc. The first standard (2) was limited to 32- and 64-bit floating-point words, and provided for optional extended formats for computers with longer words. The second standard (3) extended this to general length words and bases. The principal choices made in (2) include the following:

- Rounding to nearest (also known as round to even)
- Base 2 with a sign bit and an implicit bit
- Single precision with 8-bit exponent and 23-bit mantissa fields (not including the implicit bit)
- Double precision with 11 bit exponent and 52 bit mantissa fields (not including the implicit bit)
- The presence of $\pm\infty$ and `NaN`, as well as $\pm$ 0
- Gradually denormalized numbers for those numbers unrepresentable as normalized numbers
- User-settable bits to turn on exception handling for overflow, underflow, etc. and to vary the rounding strategies

We have tried to explain the reasons for some of these choices with the above discussion, but detailed formal analyses of these choices can be found in Ref. (1).

Usual Model for Round-Off Error

In order to analyze the behavior of algorithms in the presence of round-off errors, a mathematical model for round-off errors is defined. The usual model is as follows, where $\odot$ represents any of the four arithmetic operations:

$$fl(a \odot b) = (a \odot b) \cdot (1 + \epsilon) \tag{5}$$

where $|\epsilon| \leq$ macheps, and macheps is called the *unit roundoff* or *machine epsilon* for the given computer. The motivation behind this model is that the best any computer could do is to perform any individual arithmetic operation exactly, and then round or chop to the nearest floating-point number when finished. The rounding or chopping involves changing the last bit in the (base 2) mantissa, and hence the macheps is the value of this last bit—always relative to the size of the number itself. This model can be expensive to implement, so some computer manufacturers have designed arithmetic operations that do not obey it, but one can show that one or two guard digits suffice to be consistent with this model.

In most higher level languages, the details of the floating-point representation (especially the length of a computer word) are generally hidden from the user. Hence the macheps has a definition that can be computed in a higher level language, not specifically by the number of bits in a word. The macheps is defined as the value of ϵ yielding the minimum in

$$\min_{\epsilon > 0} fl(1 + \epsilon) > 1 \tag{6}$$

This formula can be used to calculate macheps by trying a sequence of trial values for ϵ, each entry one-half the previous, until equality in Eq. (6) is achieved. The specific value of macheps depends on the rounding strategy. This can be most easily illustrated with 3 digit decimal floating point arithmetic. The smallest s such that $fl(1 + s) > 1$ is 1.00×10^{-3} in chopping, 5.00×10^{-4} if a traditional rounding strategy is used, and 5.01×10^{-4} if rounding to even is used. In general, macheps in rounding is approximately half that obtained using chopping.

CATASTROPHIC EFFECTS OF ROUND-OFF ERROR

To illustrate how rounding errors can accumulate catastrophically in unexpected ways, we give two examples adapted from Ref. (4). An extensive introductory discussion on the effects of rounding error in scientific computations involving the use of floating point can be found in Ref. (4,5).

Of the four arithmetic operations, subtraction and addition are really the same operation. Most loss of significance and cancellation errors described below arise from these two operations. Multiplication and division give rise to problems only if the results overflow, underflow, or must be denormalized. An unusual effect of the fact that floating-point numbers are discrete in nature is that the operations no longer obey the usual laws of real numbers. For example, the associative law for addition does not hold for floating-point numbers. If s is a positive number less than macheps, but more than macheps $\div$ 2, then $1 + (s + s)$ will be strictly bigger than 1, but $(1 + s) + s$ will equal 1. This is an extreme case, but the order in which numbers are added up can affect the computed sum markedly. This is further illustrated by the first example below.

It has been pointed out (1) that the use of the denormalized numbers means that programs can depend on the fact that $fl(a - b) = 0$ implies $a = b$. However, it can still happen that

$fl(a * b) = a$ when $a \neq 0$ and $b \neq 1$. This can happen, for example, when a is the smallest representable floating-point number, and b is a number between .6 and 1, when rounding is used. Programs whose logic depend on $fl(a * b)$ being always different from a can suffer very mysterious failures. However, generally, multiplication and division do not give rise to catastrophic rounding errors unless numbers near the ends of the exponent range are involved, or when combined with other operations.

Taylor Series for e^{-40}

A simple algorithm to compute the exponential function e^x is to use its well-known Taylor series:

$$e^x = \sum_{i \geq 0} \frac{x^i}{i!}$$

When $x \geq 0$, this can yield accurate results if one is willing to take enough terms, but if used when $x < 0$, this can yield catastrophic results, all due to the finite word length of the machine. To take an extreme case, let $x = -40$. Then all the terms after the 140th term are much less than 10^{-16} and decay rapidly, and the result is also very small: $e^{-40} = 4.2484 \times 10^{-18}$. But simply adding up the terms of the Taylor series will yield 1.8654, which is nowhere near the true answer. The problem is the terms in this series alternate in sign, and the intermediate terms reach $1.4817 \times 10^{+16}$ in magnitude, and we end up subtracting very large numbers that are almost equal and opposite. This results in severe cancellation.

Numerical Derivative of e^x at $x = 1$

Suppose we take the naive approach to approximate the numerical derivative of a function f:

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

for some suitable small h. Applying this to $f(x) = e^x$ and taking the derivative at $x = 1$, we find that we get as much accuracy with $h = 2 \times 10^{-6}$ as with $h = 10^{-10}$ on a machine with approximately 16 decimal digits in the mantissa. In both cases, the error is about 3×10^{-6}, and less than half the computed digits are good. Here again we have severe cancellation from subtracting numbers that are almost equal. Hence, simply making the step size h smaller does not lead to more accuracy.

EFFECT ON ALGORITHMS

Round-Off Causes Perturbation to Data and Intermediate Results

The examples above are extreme cases showing that catastrophic loss of accuracy can result if floating-point arithmetic is not used carefully. The effect of round-off error is applied to each intermediate result and is guaranteed to be small relative to those intermediate results. However, in some cases, those intermediate results can be larger than the final desired results, leading to errors much larger than would be expected from just the sizes of the input and final output of a particular algorithm. However, in some algorithms, such as when simulating an ordinary differential equation (such as a control system) $\dot{x} = Ax + f$ where f is a forcing function, the intermediate results may not be any larger than than the final or initial values, yet severe loss of accuracy can result. One source of error is the propagation of intermediate errors, and in nasty cases, the effect of those intermediate errors can grow, becoming more and more significant as the algorithm proceeds.

Algorithm Stability versus Conditioning of Problem

In an attempt to analyze and alleviate the effects of rounding errors, numerical analysts have developed paradigms for the analysis of the behavior of numerical algorithms and have used these paradigms to develop algorithms themselves for which one can prove that the effect of rounding errors is bounded. It is useful to describe these paradigms. The most fundamental is the concept of algorithm stability versus conditioning of the problem. The latter refers to the ill posedness of the problem. If a problem is ill posed, then slight variations to the coefficients in the problem will yield massive changes to the exact solution. In this case, no floating point algorithm will be able to compute a solution with high accuracy. If the problem is well posed, then one would expect a good algorithm to compute a solution with full accuracy. An algorithm that fails that requirement is called *unstable*. An algorithm that is able to compute solutions with reasonable accuracy for well-posed problems, and that does not lose more accuracy on ill-posed problems than the ill-posed problems deserve, is called *stable.*

Relevance to Fault Tolerance

The study of rounding errors is relevant to fault tolerance in two ways. At the most elementary level, the presence of rounding errors means that no computed solution will be exact, and we cannot check for the presence of faults by checking if the computed solution satisfies some condition *exactly.* Any fault detection system would have to allow for the presence of errors in the solution arising naturally from normal rounding errors. This thus leads to the difficult task of distinguishing between errors arising from natural rounding errors and errors arising from faults. If the underlying problem is ill posed to any degree (called *ill-conditioned*) then the accuracy of the computed solution will be very poor, even if that solution were computed correctly.

On the other hand, many numerical algorithms have been shown to be stable in a certain sense. Algorithms arising in matrix computations have been especially well studied. In particular, in the domain of solving systems of linear equations, certain algorithms have been shown to compute the exact solution to a system within a small multiple of *macheps* of the original system of equations, even when the system is moderately ill posed. In some cases, precise bounds on the possible discrepancy have been derived. These can be used to develop conditions that then can be used to check for faults. Note that even if the computed solution exactly satisfies a nearby system of equations, that does not imply that the error in the solution is small, unless the system of equations is very well conditioned. As a consequence, any validation procedure for fault detection can only check for the correctness of the computed solutions indirectly, and not by computing the accuracy of the solution itself.

The result of this analysis has been the development of conditions to check the correctness of numerical computations, mainly in the domain of matrix computations and signal processing. These conditions all involve the determination of a set of precise tolerances that are tight enough to enforce sufficient accuracy in the solutions, yet guaranteed to be loose enough to be satisfiable even when solving problems that are moderately ill posed. The principal approaches in this area involve the use of checksums, backward error assertions, and mantissa checksums. In all cases, it has been found that applying these techniques to series of operations instead of checksumming each individual operation has been the most successful.

Instead of using tolerances, an alternative approach that has been used with some success is interval arithmetic. Space does not permit a full treatment here, since most software, languages, compilers, and architectures do not provide interval arithmetic as part of their built-in features. A synopsis of interval arithmetic, including its uses and applications can be found in Ref. 6. In this article, we limit our discussion to a short description. The easiest way to view interval arithmetic is to consider replacing each real number or floating-point number in the computer with two numbers representing an interval $[a, b]$ in which the "true" result is supposed to lie. Arithmetic operations are performed on the intervals. For example, addition would result in $[a_1, b_1] + [a_2, b_2] = [a_1 + a_2, b_1 + b_2]$. If all endpoints are positive, then multiplication of intervals would be computed by $[a_1, b_1] \cdot [a_2, b_2] = [a_1 \cdot a_2, b_1 \cdot b_2]$. All the other arithmetic operations and more general situations can be defined similarly. However, if no special precautions are taken, the size of the intervals can grow too large to give useful bounds on the location of the "true" answers. So most successful applications involve more sophisticated analysis of whole series of arithmetic operations such as an inner product rather than analysis of each individual operation, or else use some statistical techniques to narrow the intervals. As pointed out in Ref. 1, in order to maintain the guarantee that computed intervals contain the "true" answer, it is necessary to round down the left endpoint and round up the right endpoint of each computed interval. This requires the user to vary the rounding strategy used within the computer. The IEEE standards require that the hardware provide a way for the user to vary the rounding strategy as well as some other parameters of the arithmetic, but, as pointed out by Kahan (7) most compilers and systems today do not actually provide the user access to that level of hardware control.

SYNOPSIS OF FAULT TOLERANCE TECHNIQUES FOR LINEAR ALGEBRA

We present a short synopsis of various techniques that have been proposed for the verification of floating-point computations, mostly in linear algebra. The use of checksums was made popular by Abraham (8). This method takes advantage of the fact that the result of most computations in linear algebra bears a linear relation to the arguments originally supplied. So a linear combination of those results bears the same linear relation to that same linear combination of the original data. For example, the row operations in Gaussian elimination (used to solve systems of linear equations) can be checksummed by taking linear combinations of the entries in each row. When two rows are added in a row operation, the checksums are also added and compared with the checksum generated from scratch from the newly computed row. In a floating-point environment, the checksums will be corrupted by round-off error, and hence a tolerance must be used to decide if they match. This tolerance depends on the condition number of the matrix of *checksum coefficients* (9).

Another class of methods involves comparing the results with certain error tolerances. For matrix multiplication, the error tolerances are forward error bounds ("how far is the computed answer from the true answer?") (10). For solving systems of linear equations, the error tolerances are backward error bounds ("how well does the computed answer fit the original problem?" or more precisely, "how much must the original problem be changed so that the computed answer fits it exactly?") (11). In these methods, the error bounds used depend critically on the properties of the arithmetic, particularly the macheps, and in some cases on the conditioning of the underlying system being solved. Hence these techniques can sometimes detect violations of the mathematical assumptions of solvability that are due to ill posedness of the problem.

Yet a third class of methods is derived by considering the mantissas alone. It turns out that for certain floating-point operations (like multiplication), one can compute checksums of the mantissas alone, treating them as integers (12,13). Then the checksum computed the same way derived from the mantissa of the result must match the combination of the original mantissa checksums. Since the checksums are computed using integer arithmetic, round-off errors do not apply. The only limitation to this approach is that this technique cannot be applied to all floating-point operations (like addition), but can be used to check the multiplication part of inner products. However, when both the floating-point and integer mantissa checksum tests are applied in a "hybrid test," all operations are covered and much higher error coverages are obtained compared to using only the floating-point test.

The latter two techniques are discussed further below.

ANALYSIS OF ERROR PROPAGATION

The research area of numerical analysis is devoted to the study of the behavior of algorithms that must emulate continuous mathematics on a digital computer using floating-point arithmetic. Such analyses are based on the previously mentioned model for the error in floating-point arithmetic in Eq. (5):

$$fl(a \odot b) = (a \odot b) \cdot (1 + \epsilon)$$

where $|\epsilon| \leq$ macheps, and macheps is called the *unit roundoff* or *machine epsilon* for the given computer. We illustrate with a couple of examples how the propagation of errors is typically analyzed. Space does not permit a complete derivation of error bounds, but we refer the reader to Refs. 14 and 15 for complete discussions on error analysis of numerical algorithms.

The dot product or inner product of two vectors provides a simple example of how round-off errors can propagate. The

inner product of two vectors $\boldsymbol{x}, \boldsymbol{y}$ can be computed by

$$\boldsymbol{x} \cdot \boldsymbol{y} = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$$

In floating-point arithmetic, however, one will obtain

$$\begin{aligned} fl(\boldsymbol{x} \cdot \boldsymbol{y}) &= fl\{\ldots fl[fl(x_1 y_1) + fl(x_2 y_2)] + \cdots + fl(x_n y_n)\} \\ &= \{[(x_1 y_1)(1+\epsilon_1) + (x_2 y_2)(1+\epsilon_2)](1+\delta_2) \\ &\quad + \cdots + (x_n y_n)(1+\epsilon_n)\}(1+\delta_n) \\ &= (x_1 y_1)(1+\epsilon_1)(1+\delta_2)\ldots(1+\delta_n) \\ &\quad + (x_2 y_2)(1+\epsilon_2)(1+\delta_2)\ldots(1+\delta_n) \\ &\quad + \cdots + (x_n y_n)(1+\epsilon_n)(1+\delta_n) \end{aligned}$$

where the ϵ_i, δ_i's are quantities bounded by the macheps of the machine. Carrying out the analysis in [Ref. 15, sec. 2.4] one can obtain the relation, for some δ:

$$(1+\epsilon_1)(1+\delta_2)\ldots(1+\delta_n) = (1+\delta) \quad \text{such that} \quad |\delta| \leq 1.01n\mathrm{u}$$

where n is the dimension of the vectors and u = macheps is the *unit round-off* for the machine, under the assumption that $n\mathrm{u} < .01$. This leads to the bound on the error in the dot product (15, sec. 2.4.5).

$$|fl(\boldsymbol{x} \cdot \boldsymbol{y}) - \boldsymbol{x} \cdot \boldsymbol{y}| \leq 1.01n\mathrm{u}(|\boldsymbol{x}| \cdot |\boldsymbol{y}|)$$

where $|\boldsymbol{x}|$ denotes the vector of absolute values of the entries in $\boldsymbol{x}$. This formula can be interpreted as saying that if two vectors are accumulated together, the accumulated error is bounded by the machine unit round-off amplified by a factor growing only linearly in the dimension n. Applying this result to matrix-matrix multiplication using the usual inner product algorithm, we obtain the bound

$$fl(A \cdot B) = A \cdot B + E \quad \text{with} \quad |E| \leq 1.01n\mathrm{u}|A| \cdot |B|$$

where $\leq$ here denotes elementwise inequality.

Most algorithms, even in linear algebra, do not consist solely of inner products, and in such cases a different approach to error analysis based on the *backward error analysis* has been very successful. We consider the example of Gaussian elimination, used to solve systems of linear equations expressed in matrix terms as $A\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{x}$ is the vector of unknowns. The Gaussian elimination algorithm with row interchanges (e.g., partial pivoting) (15, sec. 3.2) can be viewed as computing the factorization of the matrix A of the form $PA = LU$, where P is a permutation matrix encoding the row interchanges occurring during the elimination process, L is a lower triangular matrix holding the multipliers, and U is an upper triangular matrix encoding the coefficients of the eliminated equations. This factorization of A into a product of simpler matrices then permits the solution of the original set of equations $A\boldsymbol{x} = \boldsymbol{b}$ by forward and back substitution.

The Gaussian elimination algorithm must compute multiples of certain rows to be added to other rows in order to eliminate variables one at a time, but in floating-point arithmetic, the multiples computed will be subject to round-off error. This means that variables will be eliminated only approximately. It becomes extremely complicated to analyze the effect of such approximations on the values of subsequent multipliers and eliminated rows. In an extreme case, slight perturbations may affect the row interchanges performed during the algorithm, yielding very different results. Hence it is possible that the computed L and U will not be close to the L, U that would be obtained in exact arithmetic. Thus it is not possible to obtain a tight *forward error bound* of the form $\|U_{\text{computed}} - U_{\text{exact}}\| \leq$ some_bound. However it has been shown that a tight *backward error bound* can be obtained. One such bound has the form (15, sec. 3.3.1).

$$L_{\text{computed}} U_{\text{computed}} = PA + H \quad \text{with} \quad |H| \leq 3n\mathrm{u}|A|\rho + O(\mathrm{u}^2)$$

where ρ is a growth factor depending on the pivoting strategy used, and is typically a small number. This bound does not say anything about how close U_{computed} is to the "true" U, but does say that L_{computed}, U_{computed} are the *exact* factors for a matrix $A + P^{\mathrm{T}}H$ that is very close to the original one. When used to compute the solution to the original system of linear equations, this will guarantee that the computed solution will almost satisfy that system of equations, or exactly satisfy a nearby system of equations, even if there is no guarantee that the solution obtained will be anywhere close to the solution that would be obtained in exact arithmetic.

INTEGER CHECKSUMS FOR FLOATING-POINT COMPUTATIONS

Floating-Point Checksum Test

Many previous approaches for error detection and correction of linear numerical computations have been based on the use of checksum schemes (8,16–18). A function f is *linear* if $f(\boldsymbol{u} + \boldsymbol{v}) = f(\boldsymbol{u}) + f(\boldsymbol{v})$, where $\boldsymbol{u}$ and $\boldsymbol{v}$ are vectors. We discuss here a commonly used checksum technique for a frequently encountered computation, matrix multiplication.

The floating-point checksum technique for matrix multiplication due to Ref. 8 is as follows. Consider an $n \times m$ matrix A with elements $a_{i,j}$, $1 \leq i \leq n$, $1 \leq j \leq m$. The *column checksum matrix* A_c of the matrix A is an $(n+1) \times m$ matrix whose first n rows are identical to those of A, and whose last row *rowsum*(A) consists of elements $a_{n+1,j} := \sum_{i=1}^{n} a_{i,j}$ for $1 \leq j \leq m$. Matrix A_c can also be defined as

$$A_c := \left[\frac{A}{\boldsymbol{e}^{\mathrm{T}}A}\right]$$

where $\boldsymbol{e}^{\mathrm{T}}$ is the $1 \times n$ row vector (1, 1, . . ., 1). Similarly, the *row checksum matrix* A_r of the matrix A is an $n \times (m+1)$ matrix whose first m columns are identical to those of A, and whose last column *colsum*(A) consists of elements $a_{i,n+1} := \sum_{j=1}^{n} a_{i,j}$ for $1 \leq i \leq n$. Matrix A_r can also be defined as $A_r := [A|A\boldsymbol{e}]$, where $A\boldsymbol{e}$ is the column summation vector. Finally, a *full checksum matrix* A_f of A is defined to be the $(n+1) \times (m+1)$ matrix, which is the column checksum matrix of the row checksum matrix A_r. Corresponding to the matrix multiplication $C := A \times B$, the relation $C_f := A_c \times B_r$ was established in Ref. 8. This result leads to their ABFT scheme for error detection in matrix multiplication, which can be described as follows:

Algorithm Mult_Float_Check(A,B)
/* A is an $n \times m$ matrix and B an $m \times l$ matrix. */

1. Compute A_c and B_r.
2. Compute $C_f := A_c \times B_r$.
3. Extract the $n \times l$ submatrix D of C_f consisting of the first n rows and l columns. Compute D_f.
4. Check if $\boldsymbol{c}_{n+1} \stackrel{?}{=} \boldsymbol{d}_{n+1}$, where $\boldsymbol{c}_{n+1}$ and $\boldsymbol{d}_{n+1}$ are the $(n+1)$th rows of C_f and D_f, respectively.
5. Check if $\boldsymbol{c}^{n+1} \stackrel{?}{=} \boldsymbol{d}^{n+1}$, where $\boldsymbol{c}^{n+1}$ and $\boldsymbol{d}^{n+1}$ are the $(n+1)$th columns of C_f and D_f, respectively.
6. If any of the above equality tests fail then return ("error") else return ("no error").

The following result was proved indirectly in Theorem 4.6 of Ref. 8.

Theorem 1 At least three erroneous elements of any full checksum matrix can be detected, and any single erroneous element can be corrected.

Theorem 1 implies that Mult_Float_Check can detect at least three errors and correct a single error in the computation of $C_f = A_c \times B_r$, as long as all operations, especially floating-point additions, have a large enough precision such that no round-off inaccuracies are introduced. Of course, such an "infinite" precision assumption is unrealistic, and thus the above checksum scheme is susceptible to round-off introduced by finite-precision floating-point arithmetic, as described earlier. In particular, there can be *false alarms* in which the checksum test fails because of round-off in spite of the absence of real errors (those occurring due to hardware glitches or failures) in the computation. Alternatively, real errors could be masked/canceled by round-off leading to nondetection of a potential problem in the hardware.

Integer Checksum Test

The susceptibility of the floating point checksum test to roundoff inaccuracies can be largely mitigated by applying integer checksums to various (linear) computations that are "mantissa preserving." This results in high error coverage and zero false alarms stemming from the fact that integer checksums do not have to contend with the round-off error problem of floating-point checksums. The integers involved are derived from the mantissas of the intermediate floating-point results of the floating-point computation. To date, we have successfully applied integer checksums (hereafter also called mantissa checksums) to two important matrix computations, matrix–matrix multiplication and **LU** decomposition (using the Gaussian elimination algorithm) (12,13). Here we briefly discuss the general theory of mantissa checksums and how they are applied to these two computations.

General Theory. In the following discussion, we use $\boldsymbol{u} = (u_1, \ldots, u_n)^T$ to represent column vectors and a, b, c, etc., for scalars. Unless otherwise specified, these variables will denote floating-point quantities. We use the notation $mant(a)$ to denote the mantissa of the floating-point number a treated as an integer. For example, considering 4-bit mantissas and integers, if 1.100 is the mantissa portion of a, with its implicit binary point shown, then the value of the mantissa is 1.5 in decimal. However, $mant(a) = 1100.$, and has value 12 in decimal. Furthermore, for a vector $\boldsymbol{v} := (v_1, \ldots, v_n)^T$, $mant(\boldsymbol{v}) := [mant(v_1), \ldots, mant(v_n)]^T$, and for a matrix $A := [a_{i,j}]$, $A^{mant} := mant(A) := [mant(a_{i,j})]$—we use the $:=$ symbol to denote equality by definition, $=$ to denote the standard (derived) equality, and $\stackrel{?}{=}$ to denote an equality test of two quantities that are theoretically supposed to be equal, but may not be because of errors and/or round-off.

Let f be any linear function on vectors. The linearity of f allows us to apply the following floating-point checksum test on the computation of f on a set S of vectors:

$$f\left(\sum_{\boldsymbol{v}\in S} \boldsymbol{v}\right) \stackrel{?}{=} \sum_{\boldsymbol{v}\in S} f(\boldsymbol{v}) \tag{7}$$

Ignoring the round-off problem, the left- hand side (LHS) and right-hand side (RHS) of the above equation should be equal, if there are no errors in computing the $f(\boldsymbol{v})$'s for all $\boldsymbol{v} \in S$ (which is the original computation), in summing up these $f(\boldsymbol{v})$'s to get the RHS, and in summing up the $\boldsymbol{v}$'s and applying f to the sum to get the LHS. If they are not equal, then an error is detected. Unfortunately, because of round-off, the test of Eq. (7) often fails to hold in the absence of computation errors. Therefore, we want to seek an integer version of this test that is not susceptible to round-off problems. Of course, this integer checksum test should involve integers derived from the floating-point quantities.

Now, since f is a linear function, irrespective of whether the vectors are floating points or integers, the following checksum property also holds:

$$f\left[\sum_{\boldsymbol{v}\in S} mant(\boldsymbol{v})\right] = \sum_{\boldsymbol{v}\in S} f[mant(\boldsymbol{v})] \tag{8}$$

where the $mant(\boldsymbol{v})$'s are integer quantities, as we saw above. Note that Eq. (8) is in general not related to the original floating-point computation $f(\boldsymbol{v})$, and can be used to check it only if f is *mantissa preserving,* that is, $f[mant(\boldsymbol{v})]$ is equal to $mant[f(\boldsymbol{v})]$, $[f(\boldsymbol{v})]$, which is derived from the original computation $f(\boldsymbol{v})$. Then the above equation becomes

$$f\left[\sum_{\boldsymbol{v}\in S} mant(\boldsymbol{v})\right] = \sum_{\boldsymbol{v}\in S} mant[f(\boldsymbol{v})] \tag{9}$$

Thus, if there are errors introduced in the mantissas of the $f(\boldsymbol{v})$'s, then those errors are also present in the $mant[f(\boldsymbol{v})]$'s and these will be detected by the integer checksum test of Eq. (9). Furthermore, this test is not susceptible to round-off. Hence it will not cause any false alarms, and very few computation errors will go undetected vis-a-vis the floating-point test of Eq. (7). In practice, since an integer word can store a finite range of numbers, integer arithmetic is effectively done modulo q where $q - 1$ is the largest integer that can be stored in the computer. Some higher-order bits can be lost in a modulo summation. However, as we will establish shortly, a single error on either side of Eq. (9) will always be detected even in the presence of overflow.

The crucial condition that must be satisfied to apply a mantissa-based integer checksum test on f is that $f[mant(\boldsymbol{v})] = mant[f(\boldsymbol{v})]$. To check if f is mantissa preserving, we have to look at the basic floating-point operations like multiplication, division, addition, subtraction, square-root, etc. that f is composed of, and see if they are mantissa preserving. A binary operator $\odot$ is said to be *mantissa preserving* if $mant(a) \odot$

$mant(b) = mant(a \odot b)$. Let a floating-point number a be represented as $a_1 \times 2^{a_2}$, where a_1 is the mantissa and a_2 the exponent of a. Ignoring the position of the implicit binary point, that is, in terms of just the bit pattern of numbers, floating-point multiplication is mantissa preserving, since

$$mant(a) \cdot mant(b) := a_1 \cdot b_1$$

while

$$mant(a \cdot b) = mant(a_1 \cdot b_1 \times 2^{a_2+b_2}) := a_1 \cdot b_1$$

Note that sometimes the mantissa c_1 of the product $c = a \cdot b$ is "forcibly" normalized by the floating-point hardware when the "natural" mantissa of the resulting product is unnormalized (e.g., $1.100 \times 1.110 = 10.101000$; the product mantissa is unnormalized, and is normalized to 1.010100, assuming 6 bits of precision after the binary point, and the exponent is incremented by 1). In such a case, c_1 is either equal to $(a_1 \cdot b_1) \div 2$ as in the previous example, or is equal to $(a_1 \cdot b_1) \div 2 - 1$ when the unnormalized mantissa has a 1 in its least-significant bit. When normalization is performed, the exponent of c becomes $a_2 + b_2 + 1$. However, this normalization done by the floating-point multiplication unit is easy to detect and reverse in c (a process we call *denormalization*) so that floating-point multiplication is effectively mantissa preserving. Similarly, floating-point division is also mantissa preserving. However, floating-point addition and subtraction are not mantissa preserving.

Thus, if f is composed of only floating-point multiplications and/or divisions, it is mantissa preserving, and we can apply the integer checksum test to it. On the other hand, if f has floating-point additions also, and there is no guarantee that the exponents of all numbers involved are equal, then f is not mantissa preserving. However, all is not lost in such a case, since it might be possible to formulate f as a composition $g \circ h (g \circ h(\boldsymbol{u}) := g[\mathrm{h}(\boldsymbol{u})])$ of two (or more) linear functions g and h, where, without loss of generality h is mantissa preserving, while g is not. In such a case, we can apply an integer checksum test to the h portion of f, that is, after computing $h(\boldsymbol{u})$, and a floating-point checksum test to f, that is, after computing $g[h(\boldsymbol{u})] := f(\boldsymbol{u})$. Since errors in $h(\boldsymbol{u})$ are caught precisely, this will still increase the error coverage and reduce the false alarm rate in checking f vis-a-vis just applying the floating-point checksum test to f. This type of a combined mantissa and floating-point checksum is called a *hybrid* checksum.

Application to Matrix Multiplication. We discuss here the application of integer mantissa checksums to matrix multiplication; the description of this test for LU decomposition can be found in Ref. 12. Matrix multiplication is not mantissa preserving, since it contains floating-point additions. However, we can formulate matrix multiplication as a composition of two functions, one mantissa preserving and the other not, as shown below.

First of all, matrix multiplication can be thought of as a sequence of vector-matrix multiplications, that is,

$$A_{n\times m} \cdot B_{m\times l} := \begin{bmatrix} \boldsymbol{a}_1^{\mathrm{T}} \cdot B \\ \boldsymbol{a}_2^{\mathrm{T}} \cdot B \\ \vdots \\ \boldsymbol{a}_n^{\mathrm{T}} \cdot B \end{bmatrix}$$

where $\boldsymbol{a}_i^{\mathrm{T}}$ is the ith row of A, and $\boldsymbol{a}_i^{\mathrm{T}} \cdot B$ is a vector-matrix multiplication. We have that $f_B(\boldsymbol{a}_i^{\mathrm{T}}) := \boldsymbol{a}_i^{\mathrm{T}} \cdot B$ is a linear function. This property leads to the *floating-point row checksum test* for matrix multiplication. In terms of f_B, the row checksum test is:

$$f_B\left(\sum_{i=1}^{n} \boldsymbol{a}_i^{\mathrm{T}}\right) \stackrel{?}{=} \sum_{i=1}^{n} f_B(\boldsymbol{a}_i^{\mathrm{T}}) \tag{10}$$

Matrix multiplication can also be thought of as a sequence of matrix-vector products $A \cdot B = (A \cdot \boldsymbol{b}_1, A \cdot \boldsymbol{b}_2, \ldots, A \cdot \boldsymbol{b}_l)$. This leads to a similar *column checksum test.*

We define a vector-vector *component-wise* product $\diamond$ for two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ as the vector

$$\boldsymbol{n} \diamond \boldsymbol{v} := (u_1 \cdot v_1, u_2 \cdot v_2, \ldots, u_n \cdot v_n)^T$$

For a matrix $B_{m\times l}$, and an m-vector $\boldsymbol{u}$, we define $\boldsymbol{u}^T \diamondsuit B$ as

$$\boldsymbol{u}^T \diamond B := (\boldsymbol{u}^T \diamond \boldsymbol{b}_1, \boldsymbol{u}^T \diamond \boldsymbol{b}_2, \ldots, \boldsymbol{u}^T \diamond \boldsymbol{b}_l)$$

where $\boldsymbol{b}_i$ denotes the i-th column of B. Thus $\boldsymbol{u}^T \diamondsuit B$ is an $m \times l$ matrix. For example,

$$(5,2)^T \diamond \begin{pmatrix} 2 & 3 \\ 1 & 4 \end{pmatrix} := \left((5,2)^T \diamond (2,1)^T, (5,2)^T \diamond (3,4)^T\right) = \begin{pmatrix} 10 & 15 \\ 2 & 8 \end{pmatrix}$$

It is easy to see that h_B defined by $h_B(\boldsymbol{u}) := \boldsymbol{u}^T \diamondsuit B$ is linear and mantissa preserving.

Finally, defining function $rowsum(C)$ for a matrix $C = (\mathbf{c}_1, \ldots, \mathbf{c}_m)$ as $rowsum(C) := [\vec{+}(\mathbf{c}_1), \ldots, \vec{+}(\mathbf{c}_m)]$ where $\vec{+}(\mathbf{v}) := \sum_{j=1}^{m} v_i$, we obtain the decomposition:

Theorem 2 (12) The vector-matrix product $\boldsymbol{u}^T \cdot B := f_B(\boldsymbol{u}) = rowsum \circ h_B(\boldsymbol{u})$.

Since matrix multiplication $A \cdot B$ is a sequence of $f_B(\boldsymbol{a}_i)$ computations, one for each row of A, we can apply a mantissa-based integer row checksum test to the $h_B(\boldsymbol{a}_i)$ components to precisely check for errors in the floating-point multiplies in $A \cdot B$. This integer row checksum test is:

$$h_{B^{mant}}\left[\sum_{i=1}^{n} mant(\boldsymbol{a}_i)\right] \stackrel{?}{=} \sum_{i=1}^{n} mant[h_B(\boldsymbol{a}_i)] \tag{11}$$

or, in other words,

$$rowsum[mant(A)] \diamond mant(B) \stackrel{?}{=} \sum_{i=1}^{n} mant(\boldsymbol{a}_i^T \diamond B) \tag{12}$$

Note that the RHS of Eq. (12) is obtained almost for free from the floating-point computations $\boldsymbol{a}_i^T \diamondsuit B$ that are computed as part of the entire floating-point vector matrix product $\boldsymbol{a}_i^T \times B$. A similar derivation can be made for an integer column checksum test.

The floating-point additions have to be tested by applying the floating-point checksum tests to $rowsum \circ h_B(\boldsymbol{u}) := f_B(\boldsymbol{u})$,

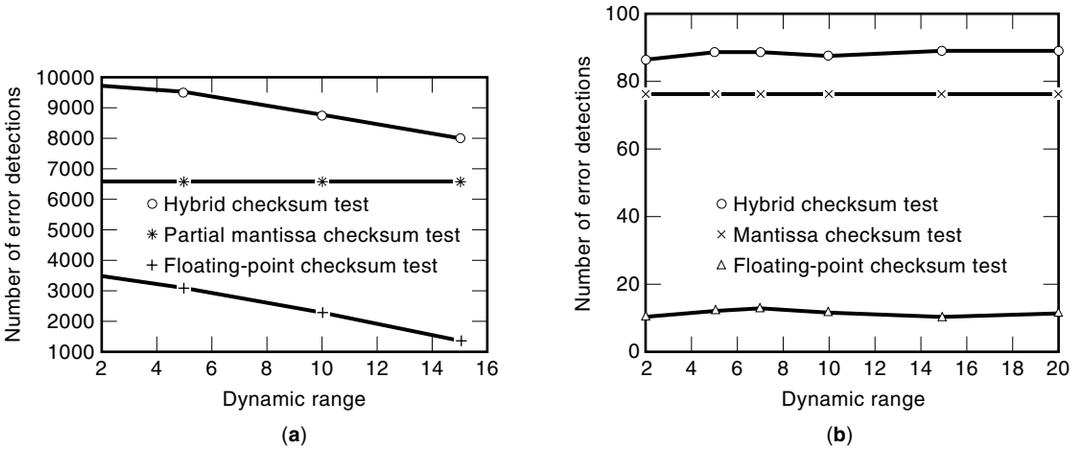


Figure 1. Error coverage vs. dynamic range of data for the mantissa checksum test, a properly thresholded floating-point checksum test, and the hybrid checksum test for (a) matrix multiplication, and (b) **LU** decomposition.

that is, to the final matrix product $A \cdot B$, to give rise to the hybrid test for matrix multiplication.

Error Coverage Results

Analytical Results. Two noteworthy results that have been obtained regarding the error coverage of the mantissa checksum method are given in the two theorems below.

Theorem 3 (12) If either modulo or extended-precision integer arithmetic is used in a mantissa checksum test of the form of Eq. (9) shown again below

$$f\left[\sum_{\boldsymbol{v}\in S} mant(\boldsymbol{v})\right] \stackrel{?}{=} \sum_{\boldsymbol{v}\in S} mant[f(\boldsymbol{v})]$$

then any single-bit error in each scalar component of this test will be detected even in the presence of overflow in modulo (or single-precision) integer arithmetic.

In Eq. (9), we compare scalars a_i and b_i, where $\boldsymbol{a} := (a_1, \ldots, a_n)^T$ and $\boldsymbol{b} := (b_1, \ldots, b_n)^T$ are the LHS and RHS, respectively, of Eq. (9). The above result means that we can detect single-bit errors in either a_i or b_i, for each i, even when single-precision integer arithmetic is used. We also have the following two results regarding the maximum number of arbitrarily distributed errors (i.e., not necessarily restricted to one error per scalar component of the check) that can be detected by the mantissa checksum test.

Theorem 4 (13) The row and column mantissa checksums for matrix multiplication can detect errors in any three elements of the product matrix $C = A \cdot B$ that are due to errors in the floating-point multiplications used to compute these elements.

The mantissa checksum test also implicitly detects errors in the exponents of the floating point products. This is done during the denormalization process by checking if $exp(a) + exp(b) = exp(a \cdot b)$ (this occurs when the floating-point multiplier did not need to normalize the product $a \cdot b$) or if $exp(a) + exp(b) = exp(a \cdot b) + 1$ (this means that a normalization was needed and the mantissa of $a \cdot b$ needs to be denormalized for use in the mantissa checksum test). If neither of these conditions hold, then an error is detected in the exponent of $a \cdot b$.

Empirical Results. A *dynamic range* of x means that the exponents of the input data lie in the interval $[-x, x]$. In Fig. 1 coverage or the number of detection events (for single errors) is plotted against different dynamic ranges of the input data for the following tests.

1. The thresholded floating-point checksum test (with the lower 24 bits masked in the checksum comparison for matrix multiplication, and 12 bits for **LU** decomposition). The threshold of the floating-point checksum test component of the hybrid checksum test was chosen to

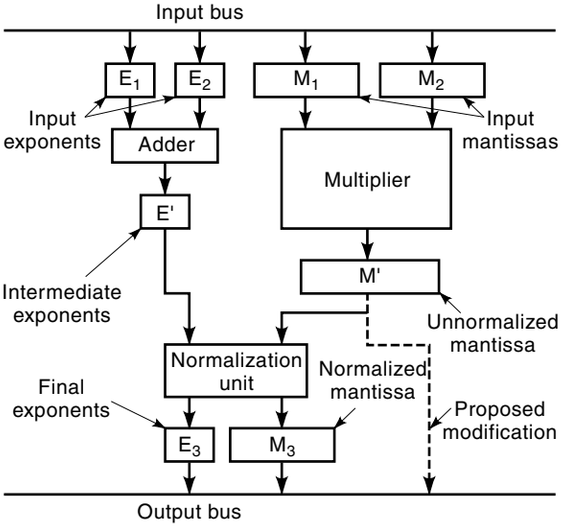


Figure 2. A simple modification of a floating-point multiplier, shown by the dashed line from internal register M' to the output bus, to make the unnormalized mantissa of the product available at no extra time penalty.

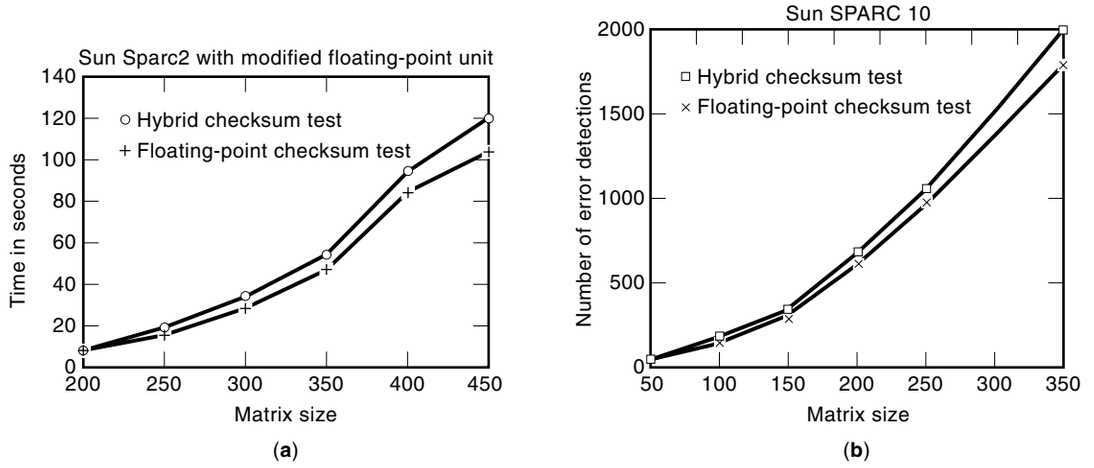


Figure 3. Timing results with a simulated modification of the floating point multiplier for (a) matrix multiplication and (b) **LU** decomposition.

correspond to masking the lower 24 (12) bits, which guarantees almost zero false alarm in matrix multiplication (**LU** decomposition).

2. The mantissa checksum test alone as described above.
3. The hybrid checksum test that uses both the thresholded floating-point test and the mantissa checksum test—an error is detected in the hybrid test if either an error is detected in its mantissa checksum test or in its floating-point checksum test.

The plots clearly show the significant improvements in coverage of the hybrid checksum test with respect to both the mantissa and the floating-point checksum tests. They also show that for the low false alarm case, the mantissa checksum test has a superior coverage compared to the floating-point checksum test. An important point to be noted that is not apparent from the plots of Fig. 1 is that the mantissa checksum test detects 100% of all multiplication errors for both matrix multiplication and **LU** decomposition.

Note that for matrix multiplication, the error coverage of the hybrid test is as high as 97% for a dynamic range of 2, and is 80% for a dynamic range of 15; this is much higher error coverage than the technique of forward error propagation used with the floating point checksum test in Ref. 10. For **LU** decomposition, we obtain error coverage of 90% for a dynamic range of 7 and Roy-Chowdhury and Banerjee (10) report a comparable coverage.

Timing Results. Note that part of the overhead of a mantissa checksum test is extracting the mantissas of the input matrices or vectors and also extracting and denormalizing the mantissas of the intermediate multiplications $a_{i,j} \cdot b_{j,k}$. The latter overhead can be eliminated by a very simple modification to the floating-point multiplication unit that is shown in Fig. 2. With this modification, the unnormalized mantissa is also available (along with the normalized mantissa) as an output of the floating-point multiplier. In many computers, the floating-point product is also available in unnormalized form by using the appropriate multiply instruction—this requires tinkering with the compiler in such a machine in order to use the unnormalized multiply instruction where appropriate. No hardware modification is needed in this case to extract the mantissa for free.

Assuming the above scenarios in which mantissa extraction and denormalization is available without any time penalty, Fig. 3 shows the plots of the times of the fault-tolerant computations that use the hybrid checksum test and that use only the floating-point checksum test. The average overhead of the hybrid checksum for matrix multiplication is 15%, while that for **LU** decomposition is only 9.5%. Thus the significantly higher error coverages yielded by the mantissa checksum test are obtained at only nominal time overheads, which are lower than those of previous techniques (10) developed for addressing the susceptibility of the floating-point checksum test to roundoff.

BIBLIOGRAPHY

1. D. Goldberg, What every computer scientist should know about floating point arithmetic. *ACM Comput. Surveys,* **23** (1): 5–48, 1991.
2. IEEE. *ANSI / IEEE Standard 754-1985 for Binary Floating Point Arithmetic.* IEEE, 1985.
3. IEEE. *ANSI / IEEE Standard 854-1987 for Radix-Independent Floating Point Arithmetic.* IEEE, 1987.
4. D. K. Kahaner, C. Moler, and S. Nash, *Numerical Methods and Software.* Englewood Cliffs, NJ: Prentice-Hall, 1989.
5. M. Heath, *Scientific Computing, An Introductory Survey.* New York: McGraw-Hill, 1997.
6. R. B. Kearfott, Interval computations: Introduction, uses, and resources. *Euromath Bull.,* **2** (1): 95–112, 1996.
7. W. Kahan, The baleful effect of computer languages and benchmarks upon applied mathematics, physics and chemistry. Presented at the SIAM Annual Meeting, 1997.
8. K. H. Huang and J. A. Abraham, Algorithm-based fault tolerance for matrix operations. *IEEE Trans. Comput.,* **C-33** (6): 518–528, 1984.
9. D. L. Boley and F. T. Luk, A well conditioned checksum scheme for algorithmic fault tolerance. *Integration, VLSI J.,* **12**: 21–32, 1991.
10. A. Roy-Chowdhury and P. Banerjee, Tolerance determination for algorithm based checks using simple error analysis techniques.

In *Fault Tolerant Comput. Symp. FTCS-23,* IEEE Press, 1993, pp. 290–298.

11. D. L. Boley et al., Floating point fault tolerance using backward error assertions, *IEEE Trans. Computers,* **44** (2): 302–311, February 1995.
12. S. Dutt and F. Assaad, Mantissa-preserving operations and robust algorithm-based fault tolerance for matrix computations, *IEEE Trans. Comput.,* **45**: 408–424, 1996.
13. F. T. Assaad and S. Dutt, More robust tests in algorithm-based fault-tolerant matrix multiplication, *22nd Fault-Tolerant Comput. Symp.,* July 1992, pp. 430–439,.
14. N. J. Higham, *Accuracy and Stability of Numerical Algorithms.* Philadelphia, PA: SIAM, 1996.
15. G. H. Golub and C. F. Van Loan, *Matrix Computations,* 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
16. P. Banerjee et al., Algorithm-based fault tolerance on a hypercube multiprocessor, *IEEE Trans. Comput.,* **39**: 1132–1145, 1990.
17. J. Y. Jou and J. A. Abraham, Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures. *Proc. IEEE,* **74**: 732–741, 1986.
18. F. T. Luk and H. Park, An analysis of algorithm-based fault tolerance. *J. Parallel Distr. Comput.,* **5**: 172–84, 1988.

SHANTANU DUTT
University of Illinois at Chicago

DANIEL BOLEY
University of Minnesota

ROUTH HURWITZ STABILITY CRITERION. See STABILITY THEORY, INCLUDING SATURATION EFFECTS.

ROUTING. See NETWORK ROUTING ALGORITHMS.

RULE-BASED SYSTEMS. See KNOWLEDGE ENGINEERING.

Switching Functions
Standard Article
Achim Ilchmann[1]
[1]University of Exeter, Exeter, England
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2455
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (142K)

Abstract

The sections in this article are

Universal Adaptive Control

Nussbaum Functions

Switching Decision Functions

Switching Functions

Unbounded Switching Functions

Applications

Acknowledgment

Keywords: nussbaum function; switching function; switching decision function; universal adaptive control; high-gain adaptive control; adaptive stabilization

| | |

Copyright © 1999-2008 All Rights Reserved.

SWITCHING FUNCTIONS

In this article, we begin by illustrating the concept of universal adaptive control by considering a simple class of scalar systems and also motivate the use of switching functions for this class. We then present Nussbaum functions. These arise naturally in the feedback law if the sign of the high-frequency gain of the system to be stabilized is unknown. An alternative to Nussbaum functions are switching decision functions which are considered in the next section. Then we discuss switching functions and unbounded switching functions, respectively. Finally, we give a brief overview of how the switching functions described above are related and used to solve the universal adaptive control problem for different classes of systems.

UNIVERSAL ADAPTIVE CONTROL

Simplified and loosely speaking, in universal adaptive control we consider a *class* of systems of the form

$$\dot{x}(t) = f(t, x(t), u(t)), \qquad y(t) = h(t, x(t)) \tag{1}$$

satisfying certain structural assumptions, and we want to design a single feedback law

$$u(t) = K_{k(t)} y(t) \tag{2}$$

and an adaptation law

$$k(t) = \varphi(t, y(\cdot)) \tag{3}$$

so that if Eqs. (2) and (3) are applied to Eq. (1), then the closed-loop system has bounded signals and meets certain other control objectives; for example, $\lim_{t\to\infty} y(t) = 0$. No iden-

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

tification mechanisms or probing signals should be incorporated.

If we restrict our attention to universal adaptive controllers that do not use any observers, then this approach was introduced for linear minimum phase systems by the seminal work of Byrnes (1), Mareels (2), Morse (3), and Willems (4) in the early 1980s.

To understand the idea, consider, instead of Eq. (1), the class of scalar systems

$$\dot{x}(t) = ax(t) + bu(t), \qquad y(t) = cx(t), \qquad x(0) = x_0 \tag{4}$$

where a, b, c, $x_0 \in \mathbb{R}$ are unknown and the only structural knowledge is $cb \neq 0$. Suppose, for a moment, the stronger assumption $cb > 0$, that is, the sign of the high-frequency gain is known, and apply

$$u(t) = -k(t)y(t) \tag{5}$$

$$\dot{k}(t) = y(t)^2 \tag{6}$$

Note that Eqs. (5) and (6) are a very simple specification of Eqs. (2) and (3), and they consist of a time-varying proportional output feedback and a monotonically nondecreasing gain adaptation. The closed-loop system becomes

$$\dot{x}(t) = [a - k(t)cb]x(t) \tag{7}$$

$$\dot{k}(t) = c^2x(t)^2 \tag{8}$$

As long as Eq. (7) is not exponentially stable, $|x(t)|$ will grow and therefore $k(t)$ will grow. Finally, $k(t)$ becomes so large that Eq. (7) is exponentially stable, and then exponential decay of $|x(t)|$ also ensures that $k(t)$ converges to a finite limit as t tends to ∞.

Morse (3) raised the question whether the knowledge of the sign of the high-frequency gain of single-input, single-output, minimum phase systems is a necessary information to achieve stabilization. For the above example, this means whether one can achieve stabilization if $cb \neq 0$. If $cb < 0$, then obviously Eq. (5) fails because the system [Eq. (7)] becomes unstable. So if the sign of cb is unknown, one has to search adaptively for the correct sign. This was achieved by Nussbaum's contribution (5), which suggested that we modify the feedback law [Eq. (5)] as follows:

$$u(t) = -k(t)\cos\sqrt{k(t)}y(t) \tag{9}$$

In fact, Nussbaum (5) presented a more general but more complicated solution. However, Eq. (9) captures the essence and is easier to understand. The intuition behind the fact that Eqs. (6) and (9) comprise a universal adaptive controller of the class Eq. (4) with $cb \neq 0$ follows: The controller has to find by itself the correct sign so that the feedback equation [Eq. (9)] stabilizes Eq. (4). The function $\cos\sqrt{k(t)}$ in Eq. (9) is responsible for the search of the sign; and while $k(t)$ in Eq. (9) is monotonically increasing, it switches sign. If the sign is "correct" (i.e., sgn $\cos\sqrt{k(t)}$ = sgn cb) and the gain is sufficiently large, then $\dot{x}(t) = [a - cb\,k(t)\cos\sqrt{k(t)}]x(t)$ is exponentially stable and $|x(t)|$ decays to zero exponentially. If the convergence is sufficiently fast so that $k(t) = k(0) + \int_0^t y(\tau)^2\,d\tau$ converges without becoming so large that $\cos\sqrt{k(t)}$ changes sign again, then the closed-loop system remains stable. The latter is ensured by the square root in $\cos\sqrt{k}$.

To see this and also to gain a deeper understanding of the general nature of this switching function approach, we sketch the proof of the universal adaptive stabilization. Observe that the closed-loop system consisting of Eqs. (4), (6), and (9) satisfies

$$\begin{aligned}\frac{d}{dt}\frac{1}{2}y(t)^2 = y(t)\dot{y}(t) &= [a - cbk(t)\cos\sqrt{k(t)}]y(t)^2 \\ &= [a - cbk(t)\cos\sqrt{k(t)}]\dot{k}(t)\end{aligned}$$

and integration together with the substitution $k(\tau) = \mu$ yields, provided that $k(t) > k(0)$:

$$\begin{aligned}\frac{1}{2}y(t)^2 - \frac{1}{2}y(0)^2 &= \int_0^t [a - cbk(\tau)\cos\sqrt{k(\tau)}]\dot{k}(\tau)\,d\tau \\ &= \int_{k(0)}^{k(t)} [a - cb\,\mu\,\cos\sqrt{\mu}]\,d\mu \\ &= [k(t) - k(0)] \\ &\quad \times \left[a - \frac{cb}{k(t) - k(0)}\int_{k(0)}^{k(t)} \mu\cos\sqrt{\mu}\,d\mu\right]\end{aligned} \tag{10}$$

Seeking a contradiction, suppose that $k(t)$ tends to ∞ as t goes to ∞ [note that by Eq. (6), $t \mapsto k(t)$ is monotonically nondecreasing]. Since

$$\frac{1}{k}\int_0^k \mu\cos\sqrt{\mu}\,d\mu = \frac{2}{k}\int_0^k \tau^3\cos\tau\,d\tau \tag{11}$$

takes arbitrary large positive and negative values as $k \to \infty$, we derive a contradiction at Eq. (10). Therefore $k(\cdot)$ must be bounded. This is equivalent to $y \in L_2(0, \infty)$. Using Eq. (7) gives $\dot{y} \in L_2(0, \infty)$. Now by a simple argument it follows that $\lim_{t\to\infty} y(t) = 0$.

The property that the function in Eq. (11) takes arbitrarily large positive and negative values as $k \to \infty$ is crucial and will be considered more generally in the following section.

NUSSBAUM FUNCTIONS

If the underlying class of systems consists of linear, multi-input, multi-output systems

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t) \tag{12}$$

where $A \in \mathbb{R}^{n\times n}$, B, $C^T \in \mathbb{R}^{n\times m}$ and the structural assumptions are minimum phase and

$$\sigma(CB) \subset \mathbb{C}_+ \quad \text{or} \quad \sigma(CB) \subset \mathbb{C}_- \tag{13}$$

then it is well known that static output feedback

$$u(t) = -Sky(t)$$

stabilizes Eq. (12) provided that k is sufficiently large and the sign is correct; that is, $S = +1$ if $\sigma(CB) \subset \mathbb{C}_+$ and $S = -1$ otherwise. If the sign is unknown, and that is what we assume in Eq. (13), then it has to be found adaptively similarly as described in the section entitled "Universal Adaptive Control." The feedback law [Eq. (2)] now becomes

$$u(t) = -N(k(t))y(t) \tag{14}$$

where $N(\cdot)$ captures the essential features of the function $k \mapsto k \cos \sqrt{k}$, and the gain adaptation [Eq. (3)] becomes

$$\dot{k}(t) = \|y(t)\|^2 \tag{15}$$

Now Eqs. (14) and (15) comprise a universal adaptive stabilizer for the class consisting of Eqs. (12) and (13) of minimum-phase systems if $N(\cdot)$ is a Nussbaum function defined as follows; see Nussbaum (5).

Definition 1. A piecewise right continuous and locally Lipschitz function $N(\cdot):[0, \infty) \to \mathbb{R}$ is called a *Nussbaum function* if, and only if, it satisfies

$$\limsup_{k>0} \frac{1}{k} \int_0^k N(\tau)\, d\tau = +\infty \quad \text{and} \quad \liminf_{k>0} \frac{1}{k} \int_0^k N(\tau)\, d\tau = -\infty \tag{16}$$

It is easy to see that Eq. (16) implies that, for every $k_0 \in (0, \infty)$,

$$\limsup_{k>k_0} \frac{1}{k-k_0} \int_{k_0}^k N(\tau)\, d\tau = +\infty$$

and

$$\liminf_{k>k_0} \frac{1}{k-k_0} \int_{k_0}^k N(\tau)\, d\tau = -\infty$$

Example 1. The following functions are Nussbaum functions:

$$N_1(k) = k^2 \cos k, \qquad k \in \mathbb{R}$$

$$N_2(k) = k \cos \sqrt{|k|}, \qquad k \in \mathbb{R}$$

$$N_3(k) = \ln k \cos \sqrt{\ln k}, \qquad k > 1$$

$$N_4(k) = \begin{cases} k & \text{if } n^2 \le |k| < (n+1)^2, \quad n \text{ even}, \\ -k & \text{if } n^2 \le |k| < (n+1)^2, \quad n \text{ odd}, \end{cases} \qquad k \in \mathbb{R}$$

$$N_5(k) = \begin{cases} k & \text{if } \tau_n \le |k| < \tau_0 \\ k & \text{if } \tau_n \le |k| < \tau_{n+1}, \quad n \text{ even} \\ -k & \text{if } \tau_n \le |k| < \tau_{n+1}, \quad n \text{ odd} \end{cases}$$

$$\text{with } \tau_0 > 1,\ \tau_{n+1} := \tau_n^2, \qquad k \in \mathbb{R}$$

$$N_6(k) = \cos\left(\frac{\pi}{2} k\right) \cdot e^{(k^2)}, \qquad k \in \mathbb{R}$$

Of course, the cosine in the above examples can be replaced by the sine. It is easy to see that $N_1(k)$, $N_2(k)$, $N_4(k)$, and $N_5(k)$ are Nussbaum functions. For a proof for $N_3(k)$ and $N_6(k)$ see Refs. 6 and 7, respectively.

$N_3(k)$ was successful if Eq. (12) consists of single-input, single-output, high-gain stabilizable systems of relative degree two (Ref. 6), and is also important when the output is sampled (Ref. 8). The function has the property that the intervals where the sign is kept constant are increasing. In fact we have $\lim_{k\to\infty} (d/dk) N_3(k) = 0$.

If the system class is subjected to actuator and sensor nonlinearities, then Eq. (16) is too weak. Therefore Logemann and Owens (7) introduced the following more restrictive concept.

Definition 2. A Nussbaum function $N(\cdot):[0, \infty) \to \mathbb{R}$ is called *scaling-invariant* if, and only if, for arbitrary $\alpha, \beta > 0$, we have

$$\tilde{N}(t) := \begin{cases} \alpha N(t) & \text{if } N(t) \ge 0 \\ \beta N(t) & \text{if } N(t) < 0 \end{cases}$$

is a Nussbaum function, too.

Scaling invariance of $N_6(k)$ is proved in Ref. 7.

SWITCHING DECISION FUNCTIONS

An alternative approach to the Nussbaum switching strategy is via a switching decision function as introduced by Ilchmann and Owens (9). As in the section entitled "Nussbaum Functions," consider the class of minimum phase systems [Eq. (12)] satisfying Eq. (13). The gain adaptation [Eq. (15)] can be slightly generalized by

$$\dot{k}(t) = \|y(t)\|^p \tag{17}$$

where $p \ge 1$, and Eq. (14) is replaced by

$$u(t) = -k(t)\Theta(t)y(t) \tag{18}$$

where $\Theta(\cdot)$ is defined as follows: Let $0 < \lambda_1 < \lambda_2 < \cdots$ be a strictly increasing sequence with $\lim_{i\to\infty} \lambda_i = \infty$ and define the function

$$\Theta(\cdot) : [0, \infty) \to \{-1, +1\}$$

by the *switching decision function*

$$\psi(t) = \frac{k_0 + \int_0^t \Theta(\tau) k(\tau) \|y(\tau)\|^p \, d\tau}{1 + \int_0^t \|y(\tau)\|^p \, d\tau}$$

and the algorithm

$$\begin{aligned} & i := 0 \\ & \Theta(0) := -1, \qquad t_0 := 0 \\ (*) \quad & t_{i+1} := \inf\{t > t_i \,|\, |\psi(t)| \le \lambda_{i+1} k(0)\} \\ & \Theta(t) := \Theta(t_i) \qquad \text{for all} \quad t \in [t_i, t_{i+1}) \\ & \Theta(t_{i+1}) := -\Theta(t_i) \\ & i := i + 1 \\ & \text{go to } (*) \end{aligned} \tag{19}$$

Then equations (17)–(19) comprise a universal adaptive stabilizer for the class of minimum phase systems [Eq. (12)] which satisfy Eq. (13). The intuition behind this control relies on the fact that the switching function $\Theta(\cdot)$ switches at each time t_i when the switching decision function $\psi(\cdot)$, which is a stability indicator, reaches the 'threshold' $\lambda_{i+1} k(0)$.

For $k(t) \ge k(0) > 0$, it is easy to see that, for every $t \ne t_i$, we obtain

$$\frac{d}{dt}\psi(t) = \begin{cases} \ge 0 & \text{if } \Theta(t) = +1 \\ \le 0 & \text{if } \Theta(t) = -1 \end{cases}$$

It can be shown that if $k(t)$ is strictly increasing, then $\psi(t)$ is either strictly increasing or decreasing, taking larger negative and positive values. Therefore, by Eq. (17), the gain $k(t)$ will increase and, by Eq. (19), $\Theta(\cdot)$ will keep on switching, until finally $k(t)$ will be so large and the sign of $\Theta(t)$ will be correct, so that the system will be stabilized and $\Theta(t)$ will not switch sign again.

The advantage of this strategy, when compared to the Nussbaum-type switching strategy, is that the "stability indicator" $\psi(t)$ is more strongly related to the dynamics of the system and the controller tolerates large classes of nonlinear disturbances. Note also that no assumption is made on how fast the sequence $\{\lambda_i\}_{i\in}\mathbb{N}$ is tending to ∞.

The close relationship between the concept of switching decision functions and Nussbaum functions is made precise in the following lemma; a proof is given in Refs. 9 and 10.

Lemma 1. Consider Eq. (12) and suppose $\dot{k}(t) = \|y(t)\|^p > 0$ almost everywhere and $k(\cdot)$ is unbounded. Then the inverse functions $\tau \mapsto k^{-1}(\tau)$ is well-defined on $[0, \infty)$, $\psi(t)$ takes arbitrary large negative and positive values, and $\eta \mapsto (\Theta \circ k^{-1})(\eta) \cdot \eta$ is a Nussbaum function.

SWITCHING FUNCTIONS

For the more general class of systems [Eq. (12)] where, instead of Eq. (13), it is only assumed that

$$\det(CB) \neq 0$$

Mårtensson (11) introduced

$$u(t) = -k(t)K_{(S\circ k)(t)}y(t) \tag{20}$$

to replace Eq. (14). Suppose $K_{(S\circ k)(\cdot)} \equiv K \in \mathbb{R}^{m\times m}$ so that $\sigma(CBK) \subset \mathbb{C}_+$, then Eq. (20) obviously stabilizes each system (12) provided that $k(\cdot) \equiv k \in \mathbb{R}$ is sufficiently large.

Such a K belongs to the so-called *finite spectrum unmixing set*—that is, a set

$$\{K_1, \ldots, K_N\} \subset GL_m(\mathbb{R})$$

so that, for any $M \in GL_m(\mathbb{R})$ there exists $i \in \{1, \ldots, N\}$ such that

$$\sigma(MK_i) \subset \mathbb{C}_+$$

The existence of this set was proved in Ref. 12. Now in the adaptive setup K is unknown and therefore $K_{(S\circ k)(t)}$ has to travel through the finite spectrum unmixing set and stay sufficiently long with the system to give it enough time to settle down. This is a similar scenario as in the single-input, single-output case ($m = 1$) where the set $\{1, -1\}$ is obviously unmixing.

In general the switching is achieved by the following function.

Definition 3. Let $N \in \mathbb{N}$. If the sequence $0 < \tau_1 < \tau_2 < \ldots$ satisfies $\lim_{i\to\infty} \tau_i = \infty$, then the associated function

$$S(\cdot) : \mathbb{R} \to \{1, \ldots, N\}, k \mapsto S(k) = \begin{cases} 1 & \text{if } k \in (-\infty, \tau_1) \\ i, & \text{if } k \in [\tau_{lN+i}, \tau_{lN+i+1}) \\ & \quad \text{for some } l \in \mathbb{N}_0, i \in \{1, \ldots, N\} \end{cases}$$

is called a *switching function.*

As for Nussbaum functions, the growth of the switching points τ_i is important, and quite often a growth condition such as

$$\lim_{i\to\infty} \frac{\tau_{i-1}}{\tau_i} = 0 \tag{21}$$

is needed.

Obviously, if $\{\tau_i\}_{i\in\mathbb{N}}$ satisfies Eq. (21), then $\lim_{i\to\infty} \tau_i = \infty$. An example for a sequence satisfying Eq. (21) is $\tau_{i+1} := \tau_i + e^{(i^2)}$; see Ref. 13.

However, the cardinality of the unmixing set can be very large. For $m = 2$ there exists an unmixing set of cardinality 6, and $GL_3(\mathbb{R})$ can be unmixed by a set with cardinality 32; see Ref. 14. Hardly anything is known on the minimum cardinality of unmixing sets for $m > 3$; see Ref. 12.

The relationship between a Nussbaum function and a switching function is given in the following lemma; for a proof see Refs. 10 and 13.

Lemma 2

1. If $S(\cdot) : \mathbb{R} \to \{1, 2\}$ is a switching function with associated sequence $\{\tau_i\}_{i\in\mathbb{N}}$ satisfying Eq. (21), then
$$N(k) = k \cdot K_{S\circ k}$$
is a Nussbaum function, where $K_1 := 1$, $K_2 := -1$, that is, a spectrum unmixing set for $\mathbb{R}\backslash\{0\}$.
2. Suppose $S(\cdot) : \mathbb{R} \to \{1, \ldots, N\}$, $N \in \mathbb{N}$, is a switching function associated with $\{\tau_i\}_{i\in\mathbb{N}}$ satisfying Eq. (21). Then, for arbitrary $\alpha > 0$ and every $i \in \{1, \ldots, N\}$, the function
$$F_i^\alpha(\cdot) : \mathbb{R} \to \mathbb{R}, \qquad k \mapsto \begin{cases} k & \text{if } S(k) = i \\ -\alpha k & \text{if } S(k) \neq i \end{cases}$$
is a scaling-invariant Nussbaum function.

UNBOUNDED SWITCHING FUNCTIONS

If even the minimum phase assumption for systems of the form presented in Eq. (12) is dropped and the only structural assumption being made is that for each system there exists a stabilizing output feedback $u(t) = -Ky(t)$ for some $K \in \mathbb{R}^{m\times m}$, then Mårtensson (11) introduced the feedback

$$u(t) = -k(t)K_{(\sigma\circ k)(t)}y(t) \tag{22}$$

Now $t \mapsto K_{(\sigma \circ k)(t)}$ has to travel through a countable set of controllers $\{K_i\}_{i \in \mathbb{N}}$ which contains some $K \in R^{m \times m}$ so that $u(t) = -Ky(t)$ stabilizes Eq. (12). $\{K_i\}_{i \in \mathbb{N}}$ could be, for example, $\mathbb{Q}^{m \times m}$.

The problem is again that $K_{(\sigma \circ k)(t)}$ stays sufficiently long at K so that the output converges to zero sufficiently fast to ensure that no more switchings occur. Otherwise, $(\sigma \circ k)(t)$ has to ensure that $K_{(\sigma \circ k)(t)}$ comes back to a neighborhood of K and this time stays even longer there. The property of "coming back" is achieved by requiring $\sigma(\cdot)$ to be an unbounded switching function defined as follows.

Definition 4. Suppose $0 < \tau_1 < \tau_2 < \ldots$ is a sequence satisfying $\lim_{i \to \infty} \tau_i = \infty$. A right continuous function $\sigma(\cdot) : \mathbb{R} \to \mathbb{N}$ is called an *unbounded switching function* with discontinuity points $\{\tau_i\}$ if, and only if, for all $a \in \mathbb{R}$, $\sigma([a, \infty)) = \mathbb{N}$.

In the literature an unbounded switching function is mostly called switching function, but here we like to emphasize the difference between a switching function and an unbounded switching function.

As in the case of switching and Nussbaum functions, the growth of the switching points is important and ensures that the system stays sufficiently long with a possibly stabilizing feedback. If we consider the class of systems described at the beginning of this section, then Eq. (22) together with the gain adaptation

$$\dot{k}(t) = \|y(t)\|^2 + \|u(t)\|^2$$

is a universal adaptive stabilizer provided that $\sigma(\cdot)$ is an unbounded switching function, the discontinuity points are given by $\tau_{i+1} = \tau_i^2$, $\tau_1 > 1$, and $\{K_i\}_{i \in \mathbb{N}} = \mathbb{Q}^{m \times m}$; for a proof see Refs. 11 and 15.

Very closely related to this concept are the so-called *tuning functions* used by Miller and Davison, who extended Mårtensson's approach considerably; for a survey of their work see Ref. 16.

APPLICATIONS

In recent years the concepts discussed above have been pushed much further for applications in adaptive control. A sophisticated switching strategy called *cyclic switching* was introduced by Morse and Pait (17,18) to solve stabilization problems which arise in the synthesis of identifier-based adaptive control. The scope of so-called *logic-based switching controllers* was discussed at a recent workshop, and many different approaches are encompassed in Ref. 19.

In the previous sections we have motivated the use of Nussbaum functions (NFs), switching decision functions (SDFs) switching functions (SFs), and unbounded switching functions (USFs) for different linear system classes. Survey articles on this subject are Refs. 10 and 20 for finite-dimensional systems and Ref. 21 for infinite-dimensional systems. In the following we relate these functions to various other classes that they have been used for and give references to where they have been studied. We only consider continuous-time systems. There are a few results available which make use of switching functions in adaptive control of discrete-time systems.

The acronym SISO is used for single-input, single-output systems, and the acronym MIMO is used for multi-input, multi-output systems.

The following first three lists are only concerned with universal adaptive *stabilization* of *minimum-phase* systems.

Linear, Finite-Dimensional, Minimum-Phase Systems

NF: SISO, $cb \neq 0$: (4,22)
NF: SISO, relative degree 2: (6,23,24)
NF: SISO, $cb \neq 0$, exponential stabilization: (25)
NF: SISO, $cb \neq 0$, nonlinear perturbations: (26,27)
NF: MIMO, $\sigma(CB) \subset \mathbb{C}_-$ or $\subset \mathbb{C}_+$, exponential stabilization: (28)
SDF: SISO, $cb \neq 0$: (29)
SF: MIMO, $\det(CB) \neq 0$, exponential stabilization: (13)
SDF: MIMO, $\sigma(CB) \subset \mathbb{C}_-$ or $\subset \mathbb{C}_+$, nonlinear perturbations: (9)

Linear, Infinite-Dimensional, Minimum-Phase Systems

NF: SISO: (30–33)
NF: SISO, nonlinear perturbations: (7,34)
NF: SISO, sector-bounded perturbations, exponential stabilization: (35)
SF: MIMO, $\det(CB) \neq 0$: (36)

Nonlinear Systems, Stabilization

NF: scalar: (37)
NF: SISO, homogeneous: (38)

Discontinuous-Feedback, Finite-Dimensional, Minimum-Phase Systems

SF: MIMO, linear, stabilization: (39)
NF: SISO, nonlinear, stabilization: (40–42)
NF: SISO, λ-tracking, nonlinear perturbations: (42–44)

So far the above articles all deal with stabilization. In the following we also consider asymptotic tracking of reference signals produced by a known linear finite-dimensional differential equation.

Tracking With Internal Model

NF: MIMO, $\sigma(CB) \subset \mathbb{C}_-$ or $\subset \mathbb{C}_+$, experimental tracking: (28)
SF: MIMO, $\det(CB) \neq 0$: (36)
NF: SISO, $cb \neq 0$, relative degree 1 or 2: (45–47)
NF: SISO, $cb \neq 0$, relative degree known: (48)
NF: MIMO, $\sigma(CB) \subset \mathbb{C}_-$ or $\subset \mathbb{C}_+$: (49)
SF: MIMO, $\det(CB) \neq 0$: (49)

In the following we consider λ-tracking of bounded reference signals with bounded derivatives. λ-tracking means that the tracking error converges to a ball around zero of prespecified radius $\lambda > 0$.

λ-Tracking, Continuous-Feedback, Minimum-Phase Systems

NF: SISO, piecewise constant gain: (50)

NF: SISO, linear, continuous gain: (51)

NF: SISO, nonlinear, continuous gain: (42,52)

Topological Aspects

SF: finite-dimensional linear, SISO, minimum phase, stabilization: (53,54)

SF: finite-dimensional linear, MIMO, $\sigma(CB) \neq 0$, minimum phase, tracking: (55)

USF: finite-dimensional linear, MIMO, nonminimum phase: (56)

SF and NF: scalar linear, exact solutions: (22,57)

Non-Minimum-Phase Systems, Stabilization

SF: MIMO, linear, stabilization: (11,58–60)

USF: MIMO, constant reference signals: (61)

USF: MIMO, linear, stabilization: (61)

USF: MIMO, tracking with internal model: (62)

USF: stable MIMO, low gain, tracking constant signals: (63)

SF & NF: stable infinite-dimensional MIMO, low gain, tracking constant signals: (64)

USF: MIMO, linear, infinite-dimensional stabilization: (15)

Non-Minimum-Phase Systems, Tracking

SF: MIMO, tracking: (65,66)

ACKNOWLEDGMENT

I am indebted to H. Logemann (Bath), D. E. Miller (Waterloo) and S. Townley (Exeter) for their constructive criticism of an earlier version of this article.

BIBLIOGRAPHY

1. C. I. Byrnes and J .C. Willems, Adaptive stabilization of multivariable linear systems, *Proc. 23rd Conf. Decis. Control,* Las Vegas, NV, 1984, pp. 1574–1577.
2. I. Mareels, A simple selftuning controller for stably invertible systems, *Syst. Control Lett.,* **4**: 5–16, 1984.
3. A. S. Morse, Recent problems in parameter adaptive control, in I. D. Landau (ed.), *Outils et Modèles Mathématiques pour l'Automatique, l'Analyse de Systèmes et le Traitment du Signal,* Paris: Editions du CNRS 3, 1983, pp. 733–740.
4. J. C. Willems and C. I. Byrnes, Global adaptive stabilization in the absence of information on the sign of the high frequency gain, *Lect. Notes Control Inf. Sci.,* **62**: 49–57, 1984.
5. R. D. Nussbaum, Some remarks on a conjecture in parameter adaptive control, *Syst. Control Lett.,* **3**: 243–246, 1983.
6. A. Ilchmann and S. Townley, Simple adaptive stabilization of high-gain stabilizable systems, *Syst. Control Lett.,* **20**: 189–198, 1993.
7. H. Logemann and D. H. Owens, Input-output theory of high-gain adaptive stabilization of infinite-dimensional systems with nonlinearities, *Int. J. Adapt. Control Signal Process.,* **2**: 193–216, 1988.
8. A. Ilchmann and S. Townley, Adaptive sampling control of high-gain stabilizable systems, *IEEE Trans. Autom. Control,* 1998, to appear.
9. A. Ilchmann and D. H. Owens, Threshold switching functions in high-gain adaptive control, *IMA J. Math. Control Inf.,* **8**: 409–429, 1991.
10. A. Ilchmann, *Non-Identifier-Based High-Gain Adaptive Control,* London: Springer-Verlag, 1993.
11. B. Mårtensson, Adaptive stabilization, Thesis, Lund Inst. of Tech., Lund, Sweden, 1986.
12. B. Mårtensson, The unmixing problem, *IMA J. Math. Control Inf.,* **8**: 367–377, 1991.
13. A. Ilchmann and H. Logemann, High-gain adaptive stabilization of multivariable linear systems—revisited, *Syst. Control Lett.,* **18**: 355–364, 1992.
14. X.-J. Zhu, A finite spectrum unmixing set for $GL_3(\mathbb{R})$, in K. Bowers and J. Lund (eds.), *Computation and Control,* Boston: Birkhäuser, 1989, pp. 403–410.
15. H. Logemann and B. Mårtensson, Adaptive stabilization of infinite-dimensional systems, *IEEE Trans. Autom. Control,* **37**: 1869–1883, 1992.
16. D. E. Miller, M. Chang, and E. J. Davison, An approach to switching control: Theory and application, *Lect. Notes Control Inf. Sci.,* **222**: 234–247, 1997.
17. F. M. Pait and A. S. Morse, A cyclic switching strategy for parameter-adaptive control, *IEEE Trans. Autom. Control,* **39**: 1172–1183, 1994.
18. A. S. Morse and F. M. Pait, MIMO design models and internal regulators for cyclicly switched parameter-adaptive control systems, *IEEE Trans. Autom. Control,* **39**: 1809–1818, 1994.
19. A. S. Morse (ed.), *Control Using Logic-Based Switching, Lect. Notes Control Inf. Sci.,* **222**, New York: Springer, 1997.
20. A. Ilchmann, Non-identifier-based adaptive control of dynamical systems: A survey, *IMA J. Math. Control Inf.,* **8**: 321–366, 1991.
21. H. Logemann and S. Townley, Adaptive control of infinite-dimensional systems without parameter estimation: an overview, *IMA J. Math. Control Inf.,* **14**: 175–206, 1997.
22. M. Heymann, J. H. Lewis, and G. Meyer, Remarks on the adaptive control of linear plants with unknown high-frequency gain, *Syst. Control Lett.,* **5**: 357–362, 1985.
23. A. S. Morse, Simple algorithms for adaptive stabilization, *Proc. ISSA. Conf. Model. Adap. Control,* Sopron, Hungary, 1998; *Lect. Notes Control Inf. Sci.,* **105**: 254–264, 1988.
24. M. Corless and E. P. Ryan, Adaptive control of a class of nonlinearly perturbed linear systems of relative degree two, *Syst. Control Lett.,* **21**: 59–64, 1993.
25. A. Ilchmann and D. H. Owens, Adaptive stabilization with exponential decay, *Syst. Control Lett.,* **14**: 437–443, 1990.
26. D. Prätzel-Wolters, D. H. Owens, and A. Ilchmann, Robust adaptive stabilization by high gain and switching, *Int. J. Control,* **49**: 1861–1868, 1989.
27. D. H. Owens, D. Prätzel-Wolters, and A. Ilchmann, Positive real structure and high gain adaptive stabilization, *IMA J. Math. Control Inf.,* **4**: 167–181, 1987.
28. A. Ilchmann and D. H. Owens, Adaptive exponential tracking for nonlinearly perturbed minimum phase systems, *Control-Theory Adv. Technol.,* **9**: 353–379, 1993.
29. A. Ilchmann and D. H. Owens, Exponential stabilization using non-differential gain adaptation, *IMA J. Math. Control Inf.,* **7**: 339–349, 1991.

30. M. Dahleh and W. E. Hopkins, Jr., Adaptive stabilization of single-input single-output delay systems, *IEEE Trans. Autom. Control,* **31**: 577–579, 1986.
31. T. Kobayashi, Global adaptive stabilization of infinite-dimensional systems, *Syst. Control Lett.,* **9**: 215–223, 1987.
32. H. Logemann and H. Zwart, Some remarks on adaptive stabilization of infinite-dimensional systems, *Syst. Control Lett.,* **16**: 199–207, 1991.
33. S. Townley, Simple adaptive stabilization of output feedback stabilizable distributed parameter systems, *Dynam. Control* **5** (2): 107–123, 1995.
34. H. Logemann and D. H. Owens, Robust and adaptive high-gain control of infinite-dimensional systems, in C. I. Byrnes, C. F. Martin, and R. E. Seaks (eds.), *Analysis and Control of Nonlinear Systems,* Amsterdam: Elsevier/North-Holland, 1988, pp. 35–44.
35. H. Logemann, Adaptive exponential stabilization for a class of nonlinear retarded processes, *Math. Control, Signals, Syst.,* **3**: 255–269, 1990.
36. H. Logemann and A. Ilchmann, An adaptive servomechanism for a class of infinite-dimensional systems, *SIAM J. Control Opt.,* **32**: 917–936, 1994.
37. B. Mårtensson, Remarks on adaptive stabilization of first order nonlinear systems, *Syst. Control Lett.,* **14**: 1–7, 1990.
38. E. P. Ryan, A universal adaptive stabilizer for a class of nonlinear systems, *Syst. Control Lett.,* **26**: 177–184, 1995.
39. E. P. Ryan, Adaptive stabilization of multi-input nonlinear systems, *Int. J. Robust Nonlinear Control,* **3**: 169–181, 1993.
40. E. P. Ryan, Discontinuous feedback and universal adaptive stabilization, in D. Hinrichsen and B. Mårtensson (eds.), *Control of Uncertain Systems,* Boston: Birkhäuser, 1990, pp. 245–258.
41. E. P. Ryan, A universal adaptive stabilizer for a class of nonlinear systems, *Syst. Control Lett.,* **16**: 209–218, 1991.
42. E. P. Ryan, An integral invariance principle for differential inclusions with applications in adaptive control, *SIAM J. Control Opt.,* **36**: 960–980, 1998.
43. E. P. Ryan, Universal $W^{1,\infty}$-tracking for a class of nonlinear systems, *Syst. Control Lett.,* **18**: 201–210, 1992.
44. E. P. Ryan, A nonlinear universal servomechanism, *IEEE Trans. Autom. Control,* **39**: 753–761, 1994.
45. A. S. Morse, An adaptive control for globally stabilizing linear systems with unknown high-frequency gains, *Lect. Notes Control Inf. Sci.,* **62**: 58–68, 1984.
46. A. S. Morse, A three-dimensional universal controller for the adaptive stabilization of any strictly proper minimum-phase system with relative degree not exceeding two, *IEEE Trans. Autom. Control,* **30**: 1188–1191, 1985.
47. A. S. Morse, A $4(n + 1)$-dimensional model reference adaptive stabilizer for any relative degree or two, minimum phase system of dimension n or less, *Automatica,* **23**: 123–125, 1987.
48. D. R. Mudgett and A. S. Morse, Adaptive stabilization of linear systems with unknown high-frequency gains, *IEEE Trans. Autom. Control,* **30**: 549–554,1985.
49. S. Townley and D. H. Owens, A note an the problem of multivariable adaptive tracking, *IMA J. Math. Control Inf.,* **8**: 389–395, 1991.
50. D. E. Miller and E. J. Davison, An adaptive controller which provides an arbitrarily good transient and steady-state response, *IEEE Trans. Autom. Control,* **36**: 68–81, 1991.
51. A. Ilchmann and E. P. Ryan, Universal λ-tracking for nonlinearly-perturbed systems in the presence of noise, *Automatica,* **30**: 337–346, 1994.
52. F. Allgöwer, J. Ashman, and A. Ilchmann, High-gain adaptive λ-tracking for nonlinear systems, *Automatica,* **33**: 881–888, 1997.
53. B. Mestel and S. Townley, Topological classification of universal adaptive dynamics, *Proc. 2nd Eur. Control Conf.,* Groningen, 1993, pp. 1597–1602.
54. J. A. Leach et al., The dynamics of universal adaptive stabilization: computational and analytical studies, *Control Theory Adv. Technol.,* **10**: 1689–1716, 1995.
55. A. Ilchmann, Adaptive controllers and root-loci of minimum phase systems, *Dyn. Control,* **4**: 203–226, 1994.
56. S. Townley, Topological aspects of universal adaptive stabilization, *SIAM J. Control Opt.,* **34**: 1044–1070, 1996.
57. A. C. Hicks and S. Townley, On exact solutions of differential equations arising in universal adaptive control, *Syst. Control Lett.,* **20**: 117–129, 1993.
58. B. Mårtensson, Adaptive stabilization of multivariable linear systems, *Contemp. Math.,* **68**: 191–225, 1987.
59. M. Fu and B. R. Barmish, Adaptive stabilization of linear systems via switching control, *IEEE Trans. Autom. Control,* **31**: 1097–1103, 1986.
60. B. Mårtensson and J. W. Polderman, Correction and simplification to "The order of a stabilizing regulator is sufficient a priori information for adaptive stabilization," *Syst. Control Lett.,* **20**: 465–470, 1993.
61. D. E. Miller and E. J. Davison, An adaptive controller which provides Lyapunov stability, *IEEE Trans. Autom. Control,* **34**: 599–609, 1989.
62. D. E. Miller and E. J. Davison, An adaptive tracking problem, *Int. J. Adapt. Track. Signal Process.,* **6**: 45–63, 1992.
63. D. E. Miller and E. J. Davison, The self-tuning robust servomechanism problem, *IEEE Trans. Autom. Control,* **34**: 511–523, 1989.
64. H. Logemann and S. Townley, Low-gain control of uncertain regular linear systems, *SIAM J. Control Optim.,* **35**: 78–116, 1997.
65. D. E. Miller, Model reference adaptive control for nonminimum phase systems, *Syst. Control Lett.,* **26**: 167–176, 1995.
66. K. Poolla and J. S. Shamma, Optimal asymptotic robust performance via nonlinear controllers, *Int. J. Control,* **62**: 1367–1389, 1995.

ACHIM ILCHMANN
University of Exeter

SWITCHING POWER SUPPLIES. See DC-DC POWER CONVERTERS; RESONANT POWER CONVERTERS.

/ /

- HOME
- ABOUT US
- CONTACT US
- HELP

Wiley Encyclopedia of Electrical and Electronics Engineering

***Browse* this title**

- *Search* this title

Enter words or phrases Go

- Advanced Product Search
- Acronym Finder

Temporal Logic
Standard Article

Ernie Cohen[1] and Sanjai Narain[1]
[1]Bellcore, Morristown, NJ

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2457
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (152K)

Abstract

The sections in this article are

Transition Systems

Linear-Time Temporal Logic

Branching-Time Logics

Reasoning About Real-Time Systems

Tools

Further Reading

| | |

Copyright © 1999-2008 All Rights Reserved.

TEMPORAL LOGIC

Concurrent systems are notoriously hard to design and debug. Part of the problem is that concurrent systems exhibit a surprising variety of behaviors, and some bugs lead to failure

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

only under pathological scenarios. The difficulty of catching such errors through conventional software engineering methods (such as testing) creates a need for more formal, systematic approaches to the design and analysis of such systems.

Temporal logic provides one such approach. The adjective *temporal* refers to the introduction of special logical modalities that allow the specification of *when* a property is expected to hold. For example, with temporal logic, one can state that if a process waits forever, it will eventually be serviced; this statement might be formalized as

$$(\Box \,\text{wait}) \Rightarrow (\Diamond \,\text{service})$$

where $\Box$ means *always* and $\Diamond$ means *eventually*. Although such analyses can be carried out within classical mathematics (by treating the system state as an explicit function of time), the encapsulation of time within temporal modalities makes the analyses easier to understand and more amenable to automation. Temporal logics are most often applied to systems that evolve through a sequence of discrete state transitions, but there are also logics designed for systems exhibiting both discrete and continuous behavior (discussed later).

An Example

As a running example, we consider Peterson's protocol for mutual exclusion (1). This protocol allows two processes (labeled P and Q) to share access to a resource, while making sure that the processes do not access the resource at the same time (this is the mutual exclusion property) and without requiring special hardware support (beyond atomic access to shared variables). While the Peterson protocol is less complex than most industrial examples (by orders of magnitude), it is still far from trivial.

For now we present the protocol with pseudocode; later, we model the protocol more precisely.

P :	Q :
$try_p := true;$	$try_q := true;$
$t := 1;$	$t := 0;$
$wait(\neg try_q \vee t = 0);$	$wait(\neg try_p \vee t = 1);$
access resource;	*access resource;*
$try_p := false;$	$try_q := false;$

The system starts with $try_p = try_q = false;$ the code shown for P is executed every time P wants to obtain access the resource (and similarly for Q).

There are several questions one might ask about this protocol:

- Does the protocol indeed prevent P and Q from accessing the resource simultaneously?
- If a P starts to execute the protocol, is it guaranteed to get access to the resource? If not, is it guaranteed that at least one of the processes will get access? If not, is it at least guaranteed that the system will not reach a deadlock state where neither component can do anything?
- On what process scheduling assumptions do these properties depend? How are these properties affected if the protocol is modified slightly (e.g., if a process is allowed to fail)?

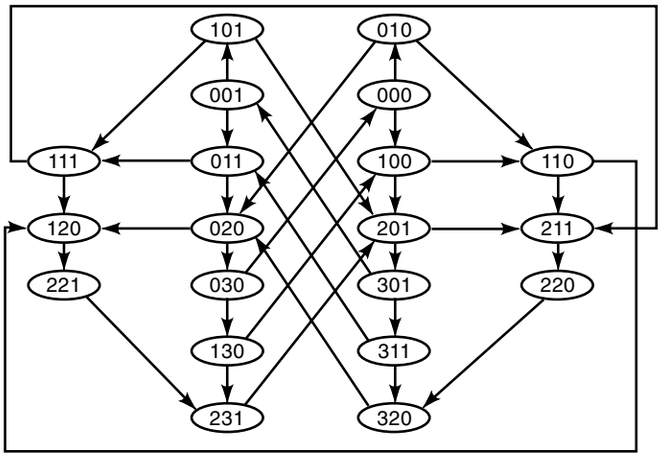


Figure 1. Peterson's protocol.

Preview

These questions are nontrivial, even for this (rather simple) protocol. One can ask similar questions about much more complex systems (e.g., microprocessors, distributed memory systems, and communication protocols). Temporal logic tools have been successfully applied to a number of such systems.

To analyze the Peterson protocol, we model it formally as a *transition system*. We then show how properties of its executions can be formulated in *linear-time logic* and proved using ordinary mathematics along with some special rules for handling temporal operators. Later we show how somewhat different properties can be formulated in *branching-time logic* and verified automatically using a special program called a *model checker*. We then survey some additional logics that can be used to reason about systems with timing constraints and systems that can undergo continuous state evolution.

TRANSITION SYSTEMS

To simplify the treatment of the Peterson protocol, we add to each process an explicit program counter, and eliminate the *try* variables; this transformation does not change the behavior of the protocol:

P :	Q :
$p := 1;$	$q := 1;$
$p, t := 2, 1;$	$q, t := 2, 0;$
$wait(q = 0 \vee t = 0);$	$wait(p = 0 \vee t = 1);$
$p := 3;$	$q := 3;$
access the resource;	*access the resource;*
$p := 0;$	$q := 0;$

Here, p, q, and t are counters (modulo 4, 4, and 2 respectively); initially, $p = q = 0$.

A *state* of the system is given by an assignment of values to its state variables (here p, q, and t). We notate states of the Peterson protocol by listing p, q, and t in order (e.g., the possible starting states of the protocol are 000 and 001). The behavior of a system can be specified by writing down a state graph showing how the state can change; the state graph of the Peterson protocol is shown in Fig. 1. (The figure does not

include states, such as 321, which are not reachable from the starting states.) However, most systems of interest are too large to be specified in this way; it is usually more practical to describe this graph with formulas, as follows.

A *transition formula* is a Boolean formula built up from primed and unprimed state variables. If f is a transition formula and $s1$ and $s2$ are states, $s1 \xrightarrow{f} s2$ is the formula obtained from f by replacing unprimed variables with their values in $s1$, and replacing primed variables with their values in $s2$. For example, if f is the transition formula $p = 3 \wedge p' = 0$, then $300 \xrightarrow{f} 001$ simplifies to *true,* but $300 \xrightarrow{f} 301$ simplifies to *false.* Note that a transition formula does not restrict how unmentioned state variables can change at the same time.

A *state formula* is a transition formula without primed variables. If f is a state formula and s is a state, $f(s)$ abbreviates $s \xrightarrow{f}$ s, which is equivalent to f with variables replaced by their values in s. If f is a state formula, f' denotes the transition formula obtained from f by priming all state variables. As a convenience, we sometimes use states as state formulas (e.g., 210 is shorthand for the state formula $p = 2 \wedge q = 1 \wedge t = 0$).

A *transition system* T is given by a state formula $T.init$ (specifying the possible starting states of the system) along with a transition formula $T.\ trans$ (specifying how the state of the system can change from one moment to the next). We omit T when its value is clear from the context. A *path* e of T is an infinite sequence of states $(e_0, e_1, \ldots)$ in which consecutive pairs of states are related by the transition relation:

$$(\forall n : 0 \leq n \Rightarrow e_n \xrightarrow{T.trans} e_{n+1})$$

If, in addition, $init(e_0)$, then e is an *initial path* of T. The initial paths of a transition system capture its possible executions; for example, the initial paths of the Peterson protocol include the path

$$(000, 100, 110, 211, 220, 320, 020, 030, 130, \ldots)$$

A systematic way to translate an ordinary concurrent program into a transition system is to introduce explicit program counters (as above), to write a transition for each atomic step of each process, and to take the disjunction of all these transitions, along with a special transition *skip* in which all of the state variables remain fixed.

For example, the transitions of the process P of the Peterson protocol can be read as follows:

$$\begin{aligned} p = 0 \wedge p' = 1 \wedge t' = t \wedge q' = q \\ p = 1 \wedge p' = 2 \wedge t' = 1 \wedge q' = q \\ (q = 0 \vee t = 0) \wedge p = 2 \wedge p' = 3 \wedge t' = t \wedge q' = q \\ p = 3 \wedge p' = 0 \wedge t' = t \wedge q' = q \end{aligned}$$

The disjunction of these transitions can be written more compactly as the logically equivalent formula

$$\begin{aligned} & (p' = p + 1 \bmod 4) \\ \wedge\ & (q' = q) \\ \wedge\ & (p\ = 2 \Rightarrow (q = 0 \vee t = 0)) \\ \wedge\ & (p\ = 1 \Rightarrow t' = 1) \\ \wedge\ & (p\ \neq 1 \Rightarrow t' = t) \end{aligned}$$

Finally, a transition of the whole system is either a transition of P, a transition of Q, or a "stuttering" step where none of the variables changes:

$$\begin{aligned} init &\equiv p = q = 0 \\ trans &\equiv P \vee Q \vee skip \\ P &\equiv p' = (p + 1 \bmod 4) \wedge q' = q \\ &\quad \wedge (p = 2 \Rightarrow (q = 0 \vee t = 0) \\ &\quad \wedge (p = 1 \Rightarrow t' = 1) \wedge (p \neq 1 \Rightarrow t' = t) \\ Q &\equiv q' = (q + 1 \bmod 4) \wedge p' = p \\ &\quad \wedge (q = 2 \Rightarrow (p = 0 \vee t = 1)) \wedge (q = 1 \Rightarrow t' = 0) \\ &\quad \wedge (q \neq 1 \Rightarrow t' = t) \\ skip &\equiv p = p' \wedge q = q' \wedge t = t' \end{aligned}$$

This system is analyzed in subsequent sections.

In passing, we note that there are alternative notations available for describing transition systems. The state-chart notation (2) provides a number of tools for making state graphs (Fig. 1) practical for somewhat larger systems. It is also possible to work directly with sequential programs (3) or communicating state machines (4).

LINEAR-TIME TEMPORAL LOGIC

In analyzing a transition system, we are primarily interested in proving that all of its paths satisfy some property. *Linear-time logics* provide languages for stating and proving properties of an individual path. There are many such logics; we illustrate some of their principles with a particularly simple logic, which we refer to as LTL. (This logic is closely related to the logic simple TLA of Ref. 5).

LTL formulas are defined as follows. Every transition formula is an LTL formula, and if f and g are LTL formulas, so are $\neg f$ ("not f"), $f \vee g$ ("f or g"), and $\Box f$ ("always f"). The semantics of LTL is given by the following rules, which define what it means for a formula f to hold for a path e (written $e \vDash f$) (the nth suffix of e, e^n, is defined as the path $e_i^n = e_{n+i}$):

$$\begin{aligned} e \vDash f &\triangleq e_0 \xrightarrow{f} e_1 \text{ for transition formula } f \\ e \vDash \neg f &\triangleq \neg(e \vDash f) \\ e \vDash f \vee g &\triangleq (e \vDash f) \vee (e \vDash g) \\ e \vDash \Box f &\triangleq (\forall n : 0 \leq n : e^n \vDash f) \end{aligned}$$

These definitions can be understood as follows. A transition formula holds for a path if and only if it relates the first two states of the path. (As a special case, a state formula holds for a path if and only if it holds for the first state of the path.) The negation of a formula holds for a path if and only if the formula does not hold for the path; the disjunction of formulas holds for a path if and only if either disjunct holds for the path. (The logical operators $\wedge$, $\Rightarrow$, and $\equiv$ can be defined from $\vee$ and $\neg$ in the usual way.) $\Box f$ holds for a path if f holds for every suffix of the path.

We define

$$\Diamond f \triangleq \neg\Box\neg f$$

$\Diamond f$ ("sometime f") holds for a path if and only if f holds for some suffix of the path. The operators $\Box$ and $\Diamond$ can be used to define a number of interesting properties:

- $\Box\Diamond f$ says that f holds infinitely often
- $\Diamond\Box f$ says that f holds almost everywhere
- $\Box(f \Rightarrow \Box f)$ says that f, once true, remains true
- $\Box(f \Rightarrow \Diamond g)$ says that every f state is followed by a g state

For any transition system T, the initial paths of T are precisely those paths satisfying the formula $T.\mathit{init} \wedge \Box T.\mathit{trans}$. This means that we can prove properties of a transition system by reasoning purely in terms of LTL formulas. It is possible to give a complete proof system for LTL, but we will instead concentrate on rules used for practical reasoning about transition systems.

Two classes of properties are of particular interest: *safety properties* ("nothing bad ever happens," e.g., the system never reaches a state where both processes are accessing the resource) and *progress properties* ("something good happens," e.g., a process trying to access the resource will eventually get in). These two classes are treated in the following sections.

Reasoning About Safety

Formulas of the form $\Box f$, where f is a transition formula, are typically proved with the following three rules:

- Propositional equivalences can be used to rewrite formulas to equivalent ones. For example, since the formulas X and $\neg\neg X$ are equivalent for any Boolean X, we can rewrite $\Box(p = 3)$ to $\Box\neg\neg(p = 3)$. We call this *the tautology rule.*
- For formulas f and g,

$$\Box(f \wedge g) \equiv \Box f \wedge \Box g$$

  (the *conjunction rule*). Note that the tautology and conjunction rules imply that $\Box$ is monotonic: if $f \Rightarrow g$ follows from ordinary propositional reasoning, then $\Box f \Rightarrow \Box g$.
- For state formula f,

$$\Box f \equiv f \wedge \Box(f \Rightarrow f')$$

  In terms of transition systems, if $T.\mathit{init}$ satisfies f and $T.\mathit{trans}$ preserves f, then f always holds throughout every initial path; such an f is called an *invariant* of the transition system. This rule says that an invariant is always true.

To prove $\Box f$, where f is a state formula, it is usually necessary to strengthen f to an invariant g. For example, the mutual exclusion condition $\neg(p = q = 3)$ holds in every reachable state of the Peterson protocol, but it is not an invariant, since *trans* does not preserve it (for example, $321 \overset{\mathit{trans}}{\Rightarrow} 331$). $\Box\neg(p = q = 0)$ can be proved as follows:

1. $(p = 3 \Rightarrow (q \leq 1 \vee t = 0))$ is an invariant of the Peterson protocol, so $\Box(p = 3 \Rightarrow (q \leq 1 \vee t = 0))$ by the invariance rule
2. Similarly, $\Box(q = 3 \Rightarrow (p \leq 1 \vee t = 1))$
3. $\Box((p = 3 \Rightarrow (q \leq 1 \vee t = 0)) \wedge (q = 3 \Rightarrow (p \leq 1 \vee t = 1)))$ from (1), (2), and the conjunction rule
4. $\Box\neg(p = q = 3)$ from (3) and the monotonicity of $\Box$

For interesting systems, the required invariants are often much more complicated than the properties being proved; this phenomenon is the primary source of complexity in most state-based program reasoning.

Reasoning About Progress

Recall that the Peterson protocol, as defined previously, has the stuttering step *skip* as one of its possible actions. Thus one possible behavior of the protocol is to remain in the same state forever; to prove that the system ever does anything, we need to add additional assumptions. We first consider how to specify these assumptions, and then show how to use them to prove more general types of progress properties.

Fairness. *Fairness conditions* provide a formal way to capture the assumption that certain things that can happen eventually do happen. For example, they can be used to guarantee that some process eventually takes a step, or that a process eventually stops accessing the resource. If f is a transition formula, f is *enabled* in those states where it is possible to execute the transition $f \wedge \mathit{trans}$; formally,

$$\mathit{enabled}.f \equiv (\exists v' : f \wedge T.\mathit{trans})$$

(where v' is the vector of all primed variables). For example, if *enterP* is the transition formula $p = 2 \wedge p' = 3$, *enabled.enterP* is the formula

$$(\exists p', q', t' : T \wedge p = 2 \wedge p' = 3)$$

If T is the Peterson protocol, this simplifies (using ordinary logical reasoning) to the state formula $p = 2 \wedge (q = 0 \vee t = 0)$.

There are several ways to specify that a transition is not unreasonably ignored.

Unconditional Fairness. The formula $\Box\Diamond f$ says that f is executed infinitely often. Note that this may have undesirable side effects; for example, unconditional fairness for *enterP* forces P to access the resource infinitely often.

Strong Fairness. The formula $\Box\Diamond \mathit{enabled}.f \Rightarrow \Box\Diamond f$ says that if f is enabled infinitely often, it must be executed infinitely often. For example, strong fairness for *enterP* says that if P infinitely often has the opportunity to access the resource, it will do so infinitely often.

Weak Fairness. The formula $\Diamond\Box \mathit{enabled}.f \Rightarrow \Box\Diamond f$ says that if f is almost always enabled, it must be happen infinitely often. For example, weak fairness for *enterP* says that if P is permanently able to enter, it will eventually do so. Note that weak fairness of f is equivalent to unconditional fairness for $\mathit{enabled}.f \Rightarrow f$.

In LTL, fairness conditions can be added directly as additional hypotheses to the formula being checked. For example, when we say that a formula f holds assuming unconditional fairness for g, we mean that $(\Box\Diamond g) \Rightarrow f$ holds.

Exploiting Fairness Hypotheses. The usual way to make use of weak or unconditional fairness is to use the following rule,

which generates a progress property from an unconditional fairness property:

$$\Box(f \Rightarrow f' \vee g \vee g') \wedge \Box\Diamond h \Rightarrow \Box(f \Rightarrow \Diamond((f \wedge h) \vee g))$$

The first hypothesis says that whenever f holds, it remains true up until the first point that g holds. For example, in the Peterson protocol, if f is the formula $p = 2 \wedge t = 0$, then, assuming weak fairness of *enterP*,

$\Box(f \Rightarrow f' \vee p' = 3)$	from $\Box$*trans* and logical reasoning
$\Box\Diamond(f \Rightarrow p = 3)$	from weak fairness of *enterP*
$\Box(f \Rightarrow \Diamond(p' = 3))$	from the last two conclusions and the progress rule

Using similar reasoning, we can obtain the following progress properties for the Peterson protocol from the corresponding weak fairness properties:

1. $\Box(p = 1 \Rightarrow \Diamond(p = 2))$	from fairness of $p = 1 \wedge p' = 2$
2. $\Box(p = 2 \wedge t = 0 \Rightarrow \Diamond(p = 3))$	from fairness of $p = 2 \wedge p' = 3$
3. $\Box(211 \Rightarrow \Diamond(p = 2 \wedge t = 0))$	from fairness of $q = 1 \wedge q' = 2$
4. $\Box(201 \Rightarrow \Diamond(p = 3 \vee 211))$	from fairness of $p = 2 \wedge p' = 3$
5. $\Box(221 \Rightarrow \Diamond 231)$	from fairness of $q = 2 \wedge q' = 3$
6. $\Box(231 \Rightarrow \Diamond 201)$	from fairness of $q = 3 \wedge q' = 0$

These basic progress properties are then combined with the following rules, that say that progress is idempotent, transitive, and disjunctive:

$$\Box(p \Rightarrow \Diamond p)$$
$$\Box(p \Rightarrow \Diamond q) \wedge \Box(q \Rightarrow \Diamond r) \Rightarrow \Box(p \Rightarrow \Diamond r)$$
$$\Box(p \Rightarrow \Diamond r) \wedge \Box(q \Rightarrow \Diamond s) \Rightarrow \Box(p \vee q \Rightarrow \Diamond(r \vee s))$$

For example, from the progress properties proved above, we can prove

$$\Box(p \geq 1 \Rightarrow \Diamond p = 3)$$

which says that if P is trying to access the resource, it will eventually obtain access:

7. $\Box(211 \Rightarrow \Diamond p = 3)$	by (3), (2), and transitivity
8. $\Box(201 \Rightarrow \Diamond p = 3)$	by (4), (7), disjunctivity and transitivity
9. $\Box(231 \Rightarrow \Diamond p = 3)$	by (6), (8), and transitivity
10. $\Box(221 \Rightarrow \Diamond p = 3)$	by (5), (9), and transitivity
11. $\Box(p = 2 \wedge t = 1 \Rightarrow \Diamond p = 3)$	by (7), (8), (9), (10), and disjunction
12. $\Box(p = 2 \Rightarrow \Diamond p = 3)$	by (11), (2), and disjunctivity
13. $\Box(p = 1 \Rightarrow \Diamond p = 3)$	by (1), (12), and transitivity
14. $\Box(p > 0 \Rightarrow \Diamond p = 3)$	by (12), (13), idempotence and disjunctivity

Strong fairness properties are exploited in the same way; the only difference from unconditional fairness is the additional disjunct $\Diamond\Box\neg$*enabled.f*, which is just treated as a separate case.

Decision Procedures for LTL

The problem of checking if an LTL formula is true is PSPACE-complete, which means that in practice the time to perform the check is exponential in the length of the formula. One way to perform this check is to treat the formula as an omega-regular language (encoding individual states as ordered lists of variable-value pairs), which reduces the validity problem to the well-known problem of checking equivalence of omega-regular languages (6).

BRANCHING-TIME LOGICS

In linear-time logics, formulas specify properties that hold for all paths. Branching-time logics provide additional flexibility by allowing one to specify that a property must hold for some path. Although sound engineering demands that a system should work for every possible execution, there are several reasons that branching-time logics are useful:

- Branching-time formulas can guarantee that the system does not unrealistically constrain the environment in which is embedded. For example, for the Peterson protocol, one might specify that in every state, it is possible that $q > 0$ in the following state; this effectively says that the process Q is free to enter the protocol at any time.
- Branching-time formulas can specify that the system cannot reach a state in which the operations are forever stuck waiting for each other to release resources (for example, by requiring that it is always possible to reach a state where $p = q = 0$).
- Possibility can sometimes be used as a substitute for guaranteed progress under fairness hypotheses, which can make model checking much easier to carry out.

Our example of a branching-time logic is computation tree logic (CTL) (7), which is the logic used by most current model checkers. As opposed to linear-time logics, which specify properties of arbitrary paths, CTL formulas are always interpreted in the context of a transition system, and formulas hold or fail to hold for a particular state, rather than for a path.

CTL formulas are defined as follows. A *path quantifier* is either **A** (*necessarily*) or **E** (*possibly*). Every state formula is a CTL formula; if **Q** is a path quantifier and f and g are CTL formulas, then $\neg f$, $f \vee g$, $\mathbf{QX}f$, and $\mathbf{Q}f\mathbf{U}g$ are CTL formulas. Formulas are interpreted as follows:

$$s \vDash f \triangleq f(s) \text{ for state formula } f$$
$$s \vDash f \vee g \triangleq (s \vDash f) \vee (s \vDash g)$$
$$s \vDash \neg f \triangleq \neg(s \vDash f)$$
$$e \vDash \mathbf{X}f \triangleq e_1 \vDash f$$
$$e \vDash f\mathbf{U}g \triangleq (\exists n : e_n \vDash g \wedge (\forall m : m < n \Rightarrow e_m \vDash f))$$
$$s \vDash \mathbf{A}f \triangleq (\forall e : e \text{ a path of } T \wedge e_0 = s \Rightarrow e \vDash f)$$
$$s \vDash \mathbf{E}f \triangleq (\exists e : e \text{ a path of } T \wedge e_0 = s \wedge e \vDash f)$$

These definitions can be read as follows. A state formula holds in a state if it holds in the sense the section entitled "Transition Systems." The disjunction of two formulas holds in a state if and only if either disjunct holds; the negation of a formula holds if and only if the formula fails to hold. $\mathbf{X}f$ ("next time f") holds for a path if and only if f holds for the second state of the path; $f\mathbf{U}g$ ("f until g") holds for a path if and only if g holds for some state of the path and f holds for every state up to the first state in which g holds. $\mathbf{A}f$ ("necessarily f") holds in a state if and only if f holds for every path starting at that state; dually, $\mathbf{E}f$ ("possibly f") holds for a state if and only if f holds for some path starting at that state.

The $\Box$ and $\Diamond$ operators can be defined from $\mathbf{U}$, that is, $\Diamond f \equiv (true\mathbf{U}f)$, and $\Box f \equiv \neg\Diamond\neg f$. The path quantifiers $\mathbf{A}$ and $\mathbf{E}$ allow one to speak about the possible futures of the system. For example,

- $\mathbf{A}\Box f$ says that f always holds
- $\mathbf{E}\Box f$ says that it is possible for f to always hold
- $\mathbf{A}\Diamond f$ says that f is guaranteed to hold eventually
- $\mathbf{E}\Diamond f$ says that it is possible for f to eventually hold
- $\mathbf{A}\Box(f \Rightarrow \mathbf{E}\Diamond g)$ says that, from every f state it is possible to reach a g state

In the case of Peterson's protocol, one might wish to prove properties like

$$\mathbf{A}\Box(q = 0 \Rightarrow \mathbf{E}(q = 0\mathbf{U}p = 3))$$

which says that at any time at which Q is not trying to access the resource, it is possible for P to gain access without Q ever entering the protocol (i.e., P can gain access without any cooperation from Q).

Some LTL formulas can be translated to CTL equivalents by prefixing every temporal operator with $\mathbf{A}$ (e.g., the LTL formula $\Box\Diamond p = 0$ translates to the CTL formula $\mathbf{A}\Box\mathbf{A}\Diamond p = 0$). However, even without considering primed variables, there are LTL formulas that cannot be written as CTL formulas. For example, the formula $\Diamond(p = 1) \vee \Box(p = 0)$ has no CTL equivalent; it is not equivalent to $\mathbf{A}\Diamond(p = 1) \vee \mathbf{A}\Box(p = 0)$ (the first holds in the Peterson protocol, while the second does not), and $\mathbf{A}(\Diamond(p = 1) \vee \Box(p = 0))$ is not a CTL formula. There are branching-time logics that generalize both LTL and CTL, such as CTL* (8), but model checking procedures for such languages are at least exponential in the size of the formula being checked. The preferred solution is to work fairness assumptions into the system model and model checking procedures.

CTL Model Checking

We now describe a simple way to check CTL formulas in a transition system, by showing how to reduce each CTL formula f to an equivalent state formula, based on the transition relation *trans*. f then holds if and only if $init \Rightarrow f$ [because this is a state formula, it can be checked using ordinary (nontemporal) logic]. To reduce $\mathbf{QX}f$ or $\mathbf{Q}f\mathbf{U}g$ to a state formula, we first reduce f and g to state formulas. The next step depends on the formula being reduced:

- The state formula for $\mathbf{EX}f$ is $(\exists\, v'\colon (trans \wedge f'))$, where v' is the vector of all primed state variables. For example, in the Peterson protocol, if f is the formula $p = 3$, then $\mathbf{EX}f$ is

  $$(\exists\, p', q', t' : (trans \wedge p' = 3))$$

  which simplifies to the state formula

  $$p = 3 \vee (p = 2 \wedge (t = 0 \vee q = 0))$$

  Similarly, the state formula for $\mathbf{AX}f$ is $(\forall\, v'\colon (T.trans \wedge f'))$.
- The state formula of $\mathbf{E}f\mathbf{U}g$ is the strongest formula x satisfying the equation $x = ((f \wedge \mathbf{EX}x) \vee g)$; that is, $\mathbf{E}f\mathbf{U}g$ is the set of all states that can reach a g state via a sequence of f transitions. x can be calculated by starting with $x = g$, and repeatedly performing the assignment $x := x \vee (f \wedge \mathbf{EX}p)$ until a fixed point is reached (i.e., until the assignment yields a logically equivalent state formula).

  The state formula of $\mathbf{A}f\mathbf{U}g$ is calculated using the same procedure, but with all $\mathbf{A}$'s above changed to $\mathbf{E}$'s.

For example, to check that $\mathbf{E}((q = 0)\mathbf{U}(p = 3))$, we first calculate the state formula $\mathbf{E}((q = 0)\mathbf{U}(p = 3))$ as above; the successive values of x are

$$p = 3$$
$$p = 3 \vee (q = 0 \wedge p = 2)$$
$$p = 3 \vee (q = 0 \wedge (p = 1 \vee p = 2))$$
$$p = 3 \vee q = 0$$

which is a fixed point. Since $init \Rightarrow (p = 3 \vee q = 0)$, $\mathbf{E}((q = 0)\mathbf{U}(p = 3))$ checks successfully.

Each of these operations can be performed in time linear in the number of states in the transition system, so the time to check a system for any CTL property grows at worst as the product of the size of the transition system and the size of the formula being checked. In contrast, the model-checking problem for LTL is PSPACE-complete, which means in practice that it is exponential in the size of the formula. Thus, CTL model checking is much more efficient.

One drawback of CTL is that, unlike LTL, fairness properties cannot be expressed within the logic. However, the model-checking procedure can be extended to work with fairness conditions; the penalty is an extra factor that is polynomial in the number of fairness assumptions (9). Most modern model checkers allow fairness assumptions as part of the system model.

The above computations can be performed symbolically (as indicated above); a model checker that works in this way is called a *symbolic model checker* (10). The main technology that makes this practical is the use of *ordered binary decision diagrams* (11) to represent state formulas in a way that makes their logical manipulation (in particular, testing whether two formulas are equivalent) very efficient (at least for large classes of formulas). An important research problem is the investigation of alternative ways to represent formulas that allow this efficient manipulation.

Explicit State Search. Most model checkers do not work explicitly with formulas, but instead work one state at a time. For example, to test $\mathbf{A}\Box f$, the checker can just enumerate the

state graph of Fig. 1, checking that each state satisfies f. This approach, called *explicit state search,* is sometimes more efficient, particularly for system with many state variables but relatively few reachable states.

In practice, the coverage of explicit state search is limited by the space needed to keep track of the set of states that have been explored. Some special techniques have been developed to overcome this problem. One way to do this efficiently is to represent a set of states using a hash table of bits; a state is in the set if the table indices it hashes to (using several independent hash functions) all have their bits set. This representation is not perfect, because multiple states might hash to the same index, but it allows large sets to be represented very efficiently (using a few bits per state) (12).

A fair body of research has been devoted to techniques that avoid exploring redundant paths to the same state ("partial order techniques") and, more generally, to states that are equivalent under some symmetry relation (for systems composed of a number of identical processes).

REASONING ABOUT REAL-TIME SYSTEMS

Some systems depend on timing constraints for correctness. For example, many message transmission protocols depend on a sender's timeout being long enough to guarantee that a message was lost if no reply is received during the timeout interval. Systems that depend on such explicit timing assumptions are generically called *real-time* systems.

One way to reason about real-time systems is to represent the time with an explicit state variable t, along with axioms that say that time never moves backward and that that the time is guaranteed to move beyond any fixed boundary. Rules of inference are used to derive real-time formulas from other real-time formulas. See, for example, (13).

For automatic verification, it is necessary to introduce timing into the automata model. A *timed automaton* is like a regular finite-state transition system, except that it also has timers that can be set, and perform a specific action when they expire. Like ordinary transition systems, CTL properties of timed automata can be checked automatically (14).

A *hybrid system* is a system that can undergo both discrete transitions and continuous evolution (e.g., where the changes of real-valued variables are governed by differential equations). Hybrid systems typically arise in control applications; for example, in air-traffic control, the movement of planes is continuous, while the protocols governing communication are discrete. We conclude with a brief description of two formalisms for reasoning about hybrid systems.

Hybrid Automata

A hybrid system's behavior over time can be modeled as a sequence of phases; during each phase the system state is governed by a set of differential equations. At a phase boundary a discrete event occurs and the system state is governed by a new set of differential equations. In *hybrid automata* (15), phases are modeled by *modes* and discrete events by *control switches*. Associated with the hybrid automaton is a set of real-valued variables $\boldsymbol{x}$ denoting the "physical state" of the system. Associated with each mode is a set of differential inequalities governing $\boldsymbol{x}$ as well as an *invariant* condition upon $\boldsymbol{x}$. Whenever this invariant is falsified, the hybrid automaton jumps out of its current mode into an adjacent mode.

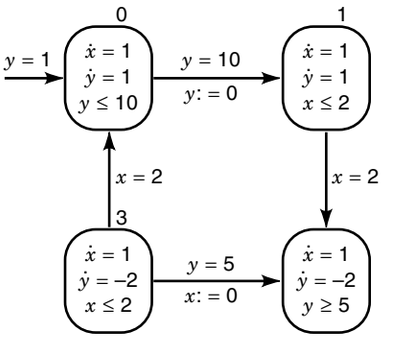


Figure 2. Water-level monitor.

An invariant is *linear* if it is a disjunction of inequalities of the form $A \cdot \boldsymbol{x} \sim \boldsymbol{c}$ where A is a constant matrix, $\boldsymbol{c}$ is a constant vector, and $\sim$ is either $\leq$ or $\geq$. A hybrid automaton is linear if its invariants are linear and its differential inequalities are of the form $A \cdot \dot{\boldsymbol{x}} \sim \boldsymbol{c}$ where $\dot{\boldsymbol{x}}$ is the vector of first derivatives of the variables $\boldsymbol{x}$.

Figure 2 shows a linear hybrid model of a control system for a water tank. The variable y represents the level of the water in the tank; the system is designed to keep this level between 1 and 12 in. It does this by turning a pump on or off; when the pump is on (states 0 and 1), the water level rises at 1 in./sec., and when the pump is off (states 2 and 3), the water level falls at 2 in./sec. The diagram can be read as follows. The system starts with the water level at 1 in. and the pump on (state 0). When the water level reaches 10 in., the timer x is reset (state 1). When the timer hits 2 sec., the pump turns off (state 2). When the water level falls to 5 in., the timer is reset (state 3); 2 sec. later, the pump turns on again (state 0).

Certain temporal properties of linear hybrid automata can be checked automatically using methods from the logical theory of linear arithmetic (15).

Duration Calculus

The duration calculus (16) is a calculus for specifying and reasoning about the duration over which formulas hold. If S is a Boolean function of time, $\int S$ denotes a function called the duration of S. The value of $\int S$ for a real-valued interval of time denotes the total duration for which S holds in that interval. Atomic formulas are predicates upon durations. Formulas are constructed from other formulas using logical connectives, and a special connective called the ; operator ("chop"). The formula $A;B$ is true for an interval $[b, e]$ when this interval can be divided into an initial subinterval $[b, m]$ for which A is true and a final subinterval $[m, e]$ for which B holds. The requirement that a formula S holds for every subinterval of an interval can be expressed as $\neg(true;\neg S;true)$, abbreviated $\Box S$. This states that it is not possible to find a subinterval in the interval for which S is false.

New duration calculus formulas can be derived using inference rules; an example of such a rule is

$$(\int S = x); (\int S = y) \Rightarrow (\int S = x + y)$$

which says that the duration of S for a sequential combination of intervals is equal to the sum of its durations for the two intervals.

For example, in the Peterson protocol, the requirement that P not wait for the resource for more than 4 time units could be written as

$$\int (p = 1 \vee p = 2) \leq 4$$

The requirement S that in a given interval of length at most 30, P not wait for more than 4 time units is expressed as

$$\int true \leq 30 \Rightarrow \int (p = 1 \vee p = 2) \leq 4$$

Note that this requirement would be trivially true for an interval larger than 30 s even when the interval contains a subinterval less than or equal to 30 s over which P waits for more than 4 s. The requirement that P waits for no more than 4 s out of any 30 s interval can be written as $\Box S$.

TOOLS

A good survey of applications of formal methods can be found in Ref. 17. Up-to-date listings of tools can be obtained from formal methods pages on the World-Wide Web. Some of the more popular notations and tools include the following:

COSPAN. The COSPAN system (6), sold commercially as FormalCheck, is an automata-based tool designed for verifying digital hardware systems designs written in the VHDL and Verilog.

HyTech. HyTech (15) is an implementation of linear hybrid automata.

SPIN. The SPIN system (12) is an explicit state-space search system for verifying systems written as synchronously communicating finite-state processes.

SMV. SMV (18) is a symbolic model checker for CTL (with fairness conditions). It has been used to find subtle published errors in a standardized cache coherence protocol.

TLA. TLA (5) (the Temporal Logic of Actions) is a linear-time logic similar to LTL, except that all formulas are stuttering-invariant, and temporal quantification is allowed. It has been applied to several large-scale specification problems.

UNITY. UNITY (19) is a linear-time programming notation and proof system. It has been applied to a large number of interesting concurrent and distributed programming problems.

FURTHER READING

The survey article, Ref. 20, provides an excellent overview of the theory of temporal logic. Ref. 21 provides a very thorough treatment of temporal logic as a specification tool, particularly for structured programs. Ref. 6, surveys automata-theoretic techniques for temporal verification. Reference 22 surveys logics for real-time systems; the book, Ref. 23, surveys logics for real-time and hybrid systems.

For other approaches to the analysis of concurrent systems, see the articles on PROCESS ALGEBRA and PROGRAMMING THEORY.

BIBLIOGRAPHY

1. G. L. Peterson, Myths about the mutual exclusion problem, *Inf. Process. Lett.,* **12** (3): 1981.
2. D. Harel, Statecharts: A visual formalism for complex systems, *Sci. Comput. Prog.,* **8**: 231–274, 1987.
3. Z. Manna and A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems, Vol. 1: Specification,* New York: Springer, 1992.
4. N. Lynch and M. Tuttle, Hierarchical correctness proofs for distributed algorithms, *ACM Symp. Prin. Distrib. Comput.,* 1987, pp. 137–151.
5. L. Lamport, The temporal logic of actions, ACM TOPLAS, **16** (3): 872–923, May 1994.
6. R. Kurshan, *Computer-Aided Verification of Coordinating Processes: The Automata-Theoretic Approach,* Princeton, NJ: Princeton University Press, 1994.
7. E. A. Emerson, E. M. Clarke, and A. P. Sistla, Automatic verification of finite state concurrent systems using temporal logic, *ACM TOPLAS,* **8** (2): 244–263, 1986.
8. E. Emerson and J. Halpern, "Sometimes" and "not never" revisited: On branching versus linear time, *J. ACM,* **33** (1): 1986.
9. E. A. Emerson and C. L. Lei, Modalities for model checking: Branching-time logic strikes back, *Sci. Comput. Prog.,* **8**: 275–307, 1987.
10. K. McMillan, *Symbolic model checking,* Ph.D. thesis, Carnegie-Mellon University, 1993.
11. R. E. Bryant, Symbolic boolean manipulation with ordered binary-decision diagrams, *ACM Comput. Surv.,* **24** (3): 293–318, 1992.
12. G. Holzmann, *Design and Validation of Computer Protocols,* Englewood Cliffs, NJ: Prentice Hall, 1991.
13. A. Pnueli, T. Henzinger, and Z. Manna, Temporal proof methodologies for real-time systems, *Proc. 18th ACM Symp. Prin. Prog. Lang.,* 1991, pp. 353–366.
14. R. Alur and D. Dill, A theory of timed automata, *Theor. Comput. Sci.,* **126**: 183–235, 1994.
15. P.-H. Ho, T. Henzinger, and H. Wong-Toi, Hytech: a model checker for hybrid systems, *Software Tools Technol. Transfer,* **1** (1), 1997.
16. A. Ravn, Z. Chaochen, and C. A. R. Hoare, A calculus of durations, *Inf. Process. Lett.,* **40** (5): 269–276, 1991.
17. E. Clarke et al., Formal methods: state of the art and future directions, *ACM Comput. Surv.,* **28** (4), 1996.
18. K. McMillan, *Symbolic Model Checking: An Approach to the State-Explosion Problem,* City: Kluwer, 1993.
19. K. M. Chandy and J. Misra, *Parallel Program Design: A Foundation,* Reading, MA: Addison-Wesley, 1988.
20. A. E. Emerson, *Temporal Logic,* Amsterdam: North-Holland, 1989, pp. 997–1073.
21. A. Pnueli and Z. Manna, *The Temporal Verification of Reactive Systems, Vol. 1: Specification,* New York: Springer, 1992.
22. J. S. Ostroff, Formal methods for specification and design of safety-critical systems, *J. Syst. Software,* **33** (60), 1992.
23. C. Heitmeyer and D. Mandrioli (eds.), *Formal Methods for Real-Time Computing,* New York: Wiley, 1996.

ERNIE COHEN
SANJAI NARAIN
Bellcore

TERMINALS, TELECOMMUNICATION. See TELECOMMUNICATION TERMINALS.

TERMINATED CIRCULATORS. See MICROWAVE ISOLATORS.

/ /

Theory of Difference Sets
Standard Article

K. T. Arasu[1] and Alexander Pott[2]
[1]Wright State University, Dayton, OH
[2]University of Augsburg, Augsburg, Germany

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2451
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (228K)

Abstract

The sections in this article are

Known Families of Difference Sets

Multipliers

Nonexistence Results Via Self-Conjugacy

Relative Difference Sets

Known Families Of Relative Difference Sets

Difference Sets and Perfect Sequences

Almost Perfect Sequences and Divisible Difference Sets

| | |

Copyright © 1999-2008 All Rights Reserved.

THEORY OF DIFFERENCE SETS

The construction of periodic sequences with good correlation properties is very important in signal processing. Many applications require knowledge of sequences and their correlation functions. In the binary case, sequences with period v can be equivalently described as subsets D of the cyclic group of order v. The distribution of the differences that can be formed with the elements from this subset D can be computed from the correlation function of the corresponding sequence. Therefore, we obtain the following meta-statement: Instead of looking for sequences with good correlation functions, we can equivalently search for subsets of cyclic groups with a good distribution of differences. A difference set is a subset of a group such that the list of differences contains every nonidentity group element equally often. If the group is cyclic, these difference sets correspond to sequences whose correlation function has just two values. Small variations of this uniform difference property correspond to small variations of the two-value property of the sequence. This indicates that the study of difference sets is also important in connection with the design of sequences with good correlation properties.

The investigation of difference sets and their generalizations is of central interest in discrete mathematics. For instance, one of the most popular conjectures, the circulant Hadamard matrix conjecture, is actually a question about difference sets. Difference sets have a long tradition: In 1938, Singer (1) pointed out that the symmetric point-hyperplane design of a finite projective space $PG(n, q)$ contains a cyclic group acting regularly (or sharply transitively) on the points. Geometers call this group the Singer cycle of $PG(n, q)$. After the pioneering work of Singer, more symmetric designs admitting sharply transitive groups (equivalently, more difference sets) have been constructed.

In this article, we describe the parameters of all currently known series of Abelian difference sets and provide constructions for most of them. We also discuss slight generalizations of difference sets (relative difference sets).

Many symmetric designs exist that cannot be constructed via a difference set. Therefore, the question about nonexistence of difference sets has also been investigated. Classical nonexistence results include multiplier arguments and the so-called Mann test (2). This test is based on the prime ideal decomposition of the order of the difference set in an appropriate cyclotomic field. However, this test requires an unfortunate assumption (self-conjugacy). Recently, several authors have tried to overcome this self-conjugacy assumption (3–7). We will survey both the classical nonexistence results as well as this new development.

Difference sets are important in combinatorial design theory and in designing sequences with good correlation properties. In this article, we try to give a flavor of various topics in this general area by including many results-new and old-but there are many results that are not included here. For recent surveys on these topics, we refer the reader to Jungnickel (8,9), Davis and Jedwab (10), Jungnickel and Pott (11). Beth et al. (12), Hall (13), Lander (14), Baumert (15), and Pott (16) serve as good reference books on related topics. In particular, the second edition of the classic book *Design Theory* (12) provides constructions of all known Abelian difference sets. The

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

book also contains the most recent nonexistence results on difference sets without the self-conjugacy assumption.

In this section, we define difference sets, introduce group rings and their characters, and mention some fundamental results that can be used to study difference sets. In the following section, we summarize all the known families of Abelian difference sets. The next section deals with multipliers, a very useful tool in the investigation of existence tests. The section thereafter will be devoted to an important concept known as *self-conjugacy,* a notion introduced by Turyn (17). Relative difference sets are then be discussed. The last two sections deal with sequences having good autocorrelation properties, which can be constructed from difference sets and their generalizations.

Let G be a multiplicatively written group of order v. A subset D of G of size k is said to be a (v, k, λ) difference set in G if each nonidentity element can be expressed in exactly λ ways as $d\,(d')^{-1}$, where $d,d' \in D$. A (v, k, λ) difference set is said to be cyclic (Abelian) if the underlying group G is cyclic (Abelian). We confine ourselves to Abelian groups throughout this article. In this case the group is usually written additively, thus explaining the term *difference set.*

An easy counting shows

$$k(k-1) = \lambda(v-1) \tag{1}$$

for any (v, k, λ) difference set.

There are always trivial examples of difference sets with parameters $(v, 1, 0)$, $(v, 0, 0)$, (v, v, v) and $(v, v-1, v-2)$ in any group of order v. Moreover, difference sets always appear in pairs: If D is a (v, k, λ) difference set in G, then the complement $G\backslash D$ is again a difference set with parameters $(v, v-k, v-2k+\lambda)$. Therefore, we may assume $k \le v/2$ (actually, it is easy to see that $k = v/2$ cannot occur).

The existence of a (v, k, λ) difference set is equivalent to the existence of a symmetric (v, k, λ) design admitting a sharply transitive automorphism group. We refer the reader to Beth et al. (12) for further details.

The investigation of non-Abelian difference sets is a rapidly growing field in discrete mathematics. However, the non-Abelian case seems to be less important for constructing good sequences. Therefore, we restrict ourselves to the case of Abelian difference sets.

An important parameter for a difference set is its order n, which is defined as $n = k - \lambda$. Sometimes, we include the order in the parameter description of a difference set and speak about $(v, k, \lambda; n)$ difference sets.

We now introduce group rings. Let G be a multiplicatively written group of order v, and let R be a commutative ring with unity 1. Then the group ring RG is the free R module with basis G equipped with the following multiplication:

$$\left(\sum_g a_g g\right)\left(\sum_h b_h h\right) = \sum_k \left(\sum_{\substack{g,h \\ gh=k}} a_g b_h\right) k$$

We shall identify the unities of R, G, and RG and denote them by 1. We will use the obvious embedding of R into RG. For any subset S of G, we let S also denote the corresponding group ring element

$$S = \sum_{g \in S} g$$

For $A = \sum_{g \in G} a_g g \in RG$ and any integer t, we define

$$A^{(t)} = \sum_{g \in G} a_g g^t$$

We get the following result.

Lemma 1. Let D be a k-subset of a group G of order v, and let R be a commutative ring with 1. Assume that D is a $(v, k, \lambda; n)$ difference set in G; then the following identity holds in RG (where $n = k - \lambda$):

$$DD^{(-1)} = n + \lambda G$$

The converse also holds provided that R has characteristic 0.

We mostly deal with the case $R = \mathbb{Z}$, the ring of integers, and the group G being Abelian. For each positive integer l, we let ξ_l denote a primitive lth root of unity. A character χ of an Abelian group G is a homomophism from G to $\mathbb{C}^*$, the nonzero complex numbers. If G has exponent e, then χ maps G to the group of vth roots of unity. Each character χ of G can be extended linearly to $\mathbb{Z}G$. This extension χ is a ring homomophism from $\mathbb{Z}G$ to $\mathbb{Z}[\xi_e]$, the ring of algebraic integers in the eth cyclotomic field $Q(\xi_e)$. Let G^* denote the set of all characters of G; then G^* is a group under pointwise multiplication.

We have the following well-known result.

Lemma 2. Inversion formula: Let $A = \sum_{g \in G} a_g g \in \mathbb{Z}G$. Then

$$a_g = \frac{1}{|G|} \sum_{\chi \in G^*} \chi(A)\chi(g^{-1})$$

Hence, if $A, B \in \mathbb{Z}G$ satisfy $\chi(A) = \chi(B)$ for all characters χ of G, then $A = B$.

A symmetric design is an incidence structure consisting of v points and v blocks (which are subsets of points) with the following properties: Any two distinct points lie in exactly λ different blocks and the block size is k. The construction of such a design out of a difference set is easy: The points are the group elements, the blocks are the so-called translates $D + g = \{d + g : d \in D\}$ of D.

Finally we quote a well-known result of Bruck, Ryser, and Chowla (18–20). Their result is more general; it is applicable to any symmetric design. We state it only for (v, k, λ) difference sets.

Theorem 1. (19,20). Let D be a (v, k, λ) difference set in a group G.

1. If v is even, then $n = k - \lambda$ is a square.

2. If v is odd, then there exist integers x, y, and z, not all zero, such that $x^2 = (k - \lambda)y^2 + (-1)^{v-1/2}\,\lambda\, z^2$.

Part 1 of Theorem is actually due to Schutzenberger (21)

KNOWN FAMILIES OF DIFFERENCE SETS

In this section we summarize the known series of Abelian difference sets. In some cases, we describe a construction, but in others we give only the parameters. The reader is referred to the chapter on Abelian difference sets in Refs. 12 and 22.

Let us begin with the most classical family, the so-called Singer difference sets.

Family I: Singer Difference Sets. Let α be a generator of the multiplicative group of $\mathbb{F}_{q^{d+1}}$. Then the set of integers $\{i\colon 0 \le i < q^{d+1}_{-1}/q - 1,\ \mathrm{tr}_{(d+1)/i}\,(\alpha^i) = 0\}$ mod $(q^{d+1} - 1)/(q - 1)$ form a (cyclic) difference set with parameters

$$\left(\frac{q^{d+1}-1}{q-1}, \frac{q^d-1}{q-1}, \frac{q^{d-1}-1}{q-1}; q^{d-1}\right)$$

Here the trace denotes the usual trace function $\mathrm{tr}_{(d+1/1)}\,(\beta) = \sum_{i=0}^{d} \beta^{q^i}$ from $\mathbb{F}_{q^{d+1}}$ onto $\mathbb{F}_q$.

In the case $d = 1$, the designs corresponding to these difference sets are the classical Desarguesian planes. The parameters can be rewritten as $(n^2 + n + 1, n + 1, 1; n)$, where n is, in the classical case, a prime power. Difference sets with these parameters are called planar difference sets. Many non-Desarguesian planes are known; however, not a single example of a plane whose order is not a prime power is known. Moreover, not a single example of a planar difference set corresponding to a non-Desarguesian plane is known. Therefore, the following two questions are of central interest in connection with planar difference sets:

Do planar difference sets of nonprime power order exist?

Do planar difference sets exist corresponding to a non-Desarguesian plane?

There are more examples of difference sets with these Singer parameters if $d > 1$. However, only one infinite family is known.

Family II: Gordon–Mills–Welch Difference Sets. Ref. (23). If s divides $d + 1$ and r is relatively prime to $q^s - 1$, then the set of integers $\{i\colon 0 \le i < (q^{d+1} - 1)/q - 1,\ \mathrm{tr}_{s/1}\,(\alpha^i)^r = 0\}$ is a cyclic difference set with the same parameters as the ones in Family I.

No examples of difference sets with Singer parameters are known when q is not a prime power.

We refer the reader to Pott (16) for more difference sets with the Singer parameters, which are equivalent to neither Singer nor Gordon–Mills–Welch difference sets. (Two difference sets D' and D are called equivalent if a translate $D' + g$ is the image of D under some automorphism of the underlying group.)

The Singer difference sets and the Gordon–Mills–Welch difference sets are basically subsets of the cyclic multiplicative group of a finite field. However, the definition of the sets use the additive structure of the fields. It is also possible to use the multiplicative group of a finite field to define subsets of the additive group which are difference sets. These are the so-called cyclotomic difference sets. The most popular examples are the Paley difference sets [squares in $GF(q)$, $q \equiv 3$ mod 4].

The next family comprises difference sets obtained using cyclotomic classes in $\mathbb{F}_q$.

Family III: Cyclotomic Difference Sets. The following subsets of $\mathbb{F}_q$ are difference sets in the additive subgroup of $\mathbb{F}_q$:

- $\mathbb{F}_q^{(2)} = \{x^2\colon x \in \mathbb{F}_q\backslash\{0,\}\}$ $q \equiv 3(\mathrm{mod}\ 4)$ (quadratic residues, Paley difference sets)
- $\mathbb{F}_q^{(4)} = \{x^4\colon x \in \mathbb{F}_q\backslash\{0,\}\}$ $q = 4t^2 + 1$, t odd
- $\mathbb{F}_q^{(4)} \cup \{0\}$, $q = 4t^2 + 9$, t odd
- $\mathbb{F}_q^{(8)} = \{x^8\colon x \in \mathbb{F}_q\backslash\{0\}\}$, $q = 8t^2 + 1 = 64u^2 + 9$, t, u odd
- $\mathbb{F}_q^{(8)} \cup \{0\}$, $q = 8t^2 + 49 = 64u^2 + 441$, t odd, u even
- $H(q) = \{x^i\colon x \in \mathbb{F}_q\backslash\{0\},\ i \equiv 0, 1 \text{ or } 3(\mathrm{mod}\ 6)\}$, $q = 4t^2 + 27$, $q \equiv 1(\mathrm{mod}\ 6)$ (Hall difference sets)

These are cyclotomic difference sets.

The next family is due to Stanton and Sprott (24).

Family IV: Twin Prime Power Difference Sets. Let q and $q + 2$ be prime powers. Then the set $D = \{(x, y)\colon x, y \text{ are both nonzero squares or both nonsquares or } y = 0\}$ is a twin prime power difference set with parameters

$$\left(q^2 + 2q, \frac{q^2+2q-1}{2}, \frac{q^2+2q-3}{4}; \frac{q^2+2q+1}{4}\right)$$

in the group $(\mathbb{F}_q, +) \oplus (\mathbb{F}_{q+2}, +)$.

We note that the Paley difference sets, the twin prime power difference sets and the Singer difference sets with $q = 2$ have parameters $(4n - 1, 2n - 1, n - 1; n)$. Difference sets with these parameters are sometimes called Paley–Hadamard difference sets.

The next construction is due to McFarland (25). The recent new constructions (Families VIII and IX) of difference sets can be viewed as far-reaching generalizations of McFarland's original work; see, in particular, Davis and Jedwab (29).

Family V: McFarland Difference Set. Let q be a prime power and d a positive integer. Let G be an Abelian group of order $v = q^{d+1}(q^d + \cdots + q^2 + q + 2)$, which contains an elementary Abelian subgroup E of order q^{d+1}. Identify E as the additive group of $\mathbb{F}_q^{d+1}$. Let $r = (q^{d+1} - 1)/(q - 1)$ and $H_1, H_2, \ldots, H_r$ be the hyperplanes of order q^d of E. If $g_0, g_1, \ldots, g_r$ are distinct coset representatives of E in G, then

$$D = (g_1 + H_1) \cup (g_2 + H_2) \cup \cdots \cup (g_r + H_r)$$

is a McFarland difference set with parameters

$$\left(q^{d+1}\left(1 + \frac{q^{d+1}-1}{q-1}\right), q^d\left(\frac{q^{d+1}-1}{q-1}\right), q^d\left(\frac{q^d-1}{q-1}\right); q^{2d}\right)$$

Modifying McFarland's construction, Spence (27) obtained the following.

Family VI: Spence Difference Sets. Let E be the elementary Abelian group of order 3^{d+1} and G a group of order $v = 3^{d+1}[(3^{d+1} - 1)/2]$ containing E. Let $m = (3^{d+1} - 1/2$ and H_1, $H_2 \ldots, H_m$ denote the subgroups of E of order 3^d. If $g_1, \ldots, g_m$ are distinct coset representatives of E in G, then

$$D = (g_1 + (E \setminus H_1) \cup (g_2 + H_2) \cup (g_3 + H_3) \cup \cdots \cup (g_m + H_m))$$

is a Spence difference set with parameters

$$\left(3^{d+1}\left(\frac{3^{d+1}-1}{2}\right), 3^d\left(\frac{3^{d+1}+1}{2}\right), 3^d\left(\frac{3^d+1}{2}\right); 3^{2d}\right)$$

We now describe Menon–Hadamard difference sets and their generalizations. Difference sets are called Menon–Hadamard if their parameters can be written in the form $(4u^2, 2u^2 - u, u^2 - u; u^2)$.

Family VII: Menon–Hadamard Difference Set. A difference set with parameters

$$(4u^2, 2u^2 - u, u^2 - u; u^2)$$

is called a Menon–Hadamard difference set.

The following theorem summarizes the known Abelian groups that contain Menon–Hadamard difference sets.

Theorem 2. Let $G \cong H \times EA(w^2)$ be an Abelian group of order $4u^2$ with $u = 2^a 3^b w^2$ where w is the product of not necessarily distinct odd primes p and $EA(w^2)$ denotes the group of order w^2, which is the direct product of groups of prime order. If H is of type $(2^{a_1})\ (2^{a_2}) \cdots (2^{a_s})\ (3^{b_1})^2 \cdots (3^{b_r})^2$ with $\sum a_i = 2a + 2$ $(a \geq 0, a_i \leq a + 2)$, $\sum b_i = 2b$ $(b \geq 0)$, then G contains a Menon–Hadamard difference set of order u^2.

We provide one construction for Family VII; several others are known.

Theorem 3. Let $H = \langle a, b: a^{s+1} = b^{s+1} = 1\rangle$ be an Abelian group of type $(2^{s+1})\ (2^{s+1})$. Let f be a mapping $\mathbb{Z}_{2^{s+1}} \to \{\pm 1\}$ satisfying $f(i + 2^s) = -f(i)$. Define a mapping μ: $\mathbb{Z}_{2^{s+1}} \to \mathbb{Z}_{2^{s+1}}$ by $\mu(2^r i) = 2^r i^*$, where i is odd and $ii^* \equiv 1 \pmod{2^{s+1}}$. Then the set $D = \{a^i b^j: f(\mu(i)j) = -1\}$ is a Menon–Hadamard difference set with $u = 2^s$. Let $G = \langle a^2, c: c^2 = b\rangle$ be an Abelian group of type $(2^s)\ (2^{s+2})$. If $A = D \cap \langle a^2\rangle\ \langle b\rangle$ and $B = a^{-1}(D \backslash A)$, then $A \cup cB$ is a Menon–Hadamard difference set with $u = 2^s$ in G.

Theorem 3 is due to Dillon (28). Our next family is contained in the very important "unifying" work of Davis and Jedwab (29).

Family VIII: Davis–Jedwab Difference Sets. A difference set with parameters

$$\left(2^{2d+4}\left(\frac{2^{2d+2}-1}{3}\right), 2^{2d+1}\left(\frac{2^{2d+3}+1}{3}\right)\right.$$
$$\left. 2^{2d+1}\left(\frac{2^{2d+1}+1}{3}\right); 2^{4d+2}\right)$$

is called a Davis–Jedwab difference set. (Here d is any nonnegative integer).

These difference sets exist in all Abelian groups of order $2^{2d+4}\ [(2^{2d+2} - 1)/3]$ that have a Sylow 2-subgroup S_2 of exponent at most 4, with the single exception $d = 1$ and $S_2 \cong \mathbb{Z}_4^3$.

The most recent family due to Chen (30) is as follows.

Family IX: Chen Difference Sets. A difference set with parameters

$$\left(4q^{2d+2}\left(\frac{q^{2d+2}-1}{q^2-1}\right), q^{2d+1}\left(\frac{2(q^{2d+2}-1)}{q+1}+1\right)\right.$$
$$\left. q^{2d+1}(q-1)\left(\frac{q^{2d+1}+1}{q+1}\right); q^{4d+2}\right)$$

is called a Chen difference set. (Here d is a nonnegative integer and q a prime power.) The Chen family with $d = 0$ corresponds to the Menon–Hadamard family; the Chen family with $q = 2$ corresponds to the Davis–Jedwab family; and the Chen family with $q = 3$ corresponds to the Spence family. We distinguish these series for historical reasons.

Looking at the families mentioned previously we have two major questions. The first question is about nonexistence. What happens, for instance, if we replace the prime power q or the values of u in the Menon–Hadamard series by some other integer? Can we prove that for these different parameters no difference set can exist? More specifically, the following two questions have attracted a lot of attention:

Prime-Power Conjecture (PPC). Determine the parameters n for which an $(n^2 + n + 1, n + 1, 1; n)$ difference set can exist. The Singer examples with $d = 2$ show that examples exist whenever n is a prime power.

Menon–Hadamard Conjecture (MHC). Determine the possible values for u such that a Menon–Hadamard difference set of order u^2 exists. Is it true that u must be of the form in Theorem 2?

Similar questions can be asked about the Davis–Jedwab–Chen difference sets.

Another question addresses the groups that might carry difference sets. Although we know that difference sets with the parameters mentioned previously in the series do exist, it is not at all clear which Abelian groups contain these sets. In the description of our families, we have always described the groups for which it is known that they contain difference sets. In general, it is not at all clear whether other groups are also possible. Several partial nonexistence results in this direction are known for all the series mentioned previously. One of the most satisfying theorems in this direction is the following.

Theorem 4 (31). Let G be an Abelian group of order 2^{2d+2}. Then G contains a Menon–Hadamard difference set if and only if G satisfies Turyn's exponent bound $\exp(G) \leq 2^{d+2}$.

Finally, it would be interesting to obtain classification results saying that the only difference sets with certain parameters are the known ones. For instance, the only known planar difference sets correspond to the classical Desarguesian planes. A classification result would say that this must be the case.

MULTIPLIERS

Let D be a (v, k, λ) difference set in a group G. An automorphism α of G is said to be a multiplier of D if $\alpha(D) = Dg$ for some $g \in G$. If G is Abelian and if α is given by multiplication by an integer t relatively prime to the order of G, we say that t is a numerical multiplier, or simply, multiplier of D. The parameters of a hypothetical Abelian difference set D would sometimes imply the existence of numerical multipliers, which could then be used to investigate the existence of D. These ideas are due to Hall (32), who considered these for the case $\lambda = 1$. An easy extension of Hall's result obtained by Chowla and Ryser (20) is given in the following.

Theorem 5 (First Multiplier Theorem). Let D be an Abelian (v, k, λ) difference set. Let p be a prime dividing $n = k - \lambda$, but not v. If $p > \lambda$, then p is a multiplier of D.

To use the multipliers, we also need a result of McFarland and Rice (33).

Theorem 6. Let D be an Abelian (v, k, λ) difference set in G. Then there exists a translate of D that is fixed by every numerical multiplier of D.

Example 2. Consider a (21, 5, 1) difference set in $\mathbb{Z}_{21}$. Here 2 is a multiplier by Theorem 5. We may assume D consists of orbits of $\mathbb{Z}_{21}$ under $x \to 2x$, by Theorem 6. Since $k = 5$, D must be formed from the orbits $\{0\}$, $\{7, 14\}$, $\{3, 6, 12\}$, $\{9, 18, 15\}$. $D_1 = \{7, 14, 3, 6, 12\}$ and $D_2 = \{7, 14, 9, 18, 15\}$ both work.

Similarly, we can obtain $\{0, 1, 3, 9\}$ as a (13, 4, 1) difference set in $\mathbb{Z}_{13}$, and $\{1, 2, 4, 8, 16, 32, 64, 5, 37\}$ as a (73, 9, 1) difference set in $\mathbb{Z}_{73}$.

Example 2. Consider a hypothetical (31, 10, 3) difference set D in $\mathbb{Z}_{31}$. 7 is a multiplier of D by Theorem 5. But the orbits of $\mathbb{Z}_{31}$ under $x \to 7x$ have sizes 1, 15, 15. Hence D cannot exist.

The Multiplier Conjecture. Theorem 4 holds without the assumption that $p > \lambda$. All known multiplier theorems may be viewed as an attempt to eliminate conditions such as $p > \lambda$.

Theorem 7 (Second Multiplier Theorem) (34). Let D be an Abelian (v, k, λ)-difference set in G, and let $m > \lambda$ be a divisor of n that is co-prime with v. Moreover, let t be an integer co-prime with v satisfying the following condition: For every prime p dividing m there exists a nonnegative integer f with $t \equiv p^f \pmod{v^*}$, where v^* denotes the exponent of G. Then t is a numerical multiplier for D.

We next state another multiplier theorem due to McFarland (35). We first define a function M as follows:

$$M(2) = 2 \times 7, \quad M(3) = 2 \times 3 \times 11 \times 13$$
$$M(4) = 2 \times 3 \times 7 \times 31$$

recursively, $M(z)$ for $z \geq 5$ is defined as the product of the distinct prime factors of the numbers

$$z, M\left(\frac{z^2}{p^{2e}}\right), p - 1, p^2 - 1, \ldots, p^{u(z)} - 1$$

where p is a prime dividing m with $p^e \parallel m$ and where $u(z) = (z^2 - z)/2$. (The notation $p^a \parallel m$ means that $p^a \mid m$ but $p^{a+1} \nmid m$; we then say that p^a strictly divides m.)

Theorem 8. Theorem 7 remains true if the assumption $m > \lambda$ is replaced by $M(n/m)$ and v are co-prime.

NONEXISTENCE RESULTS VIA SELF-CONJUGACY

Multipliers provide nonexistence results, as we saw earlier. But most of the multiplier theorems for $(v, k, \lambda; n)$ difference sets have been proved when $(v, n) = 1$. An extension of known multiplier theorems to cover a few cases with $(v, n) > 1$ can be found in Arasu and Xiang (36), but these results are difficult to apply. Almost all results on difference sets with $(v, n) > 1$ pertain to exponent bounds and rely on character theoretic ideas introduced by Turyn (17) in his seminal paper. The notion of self-conjugacy is introduced by Turyn. A prime p is said to be self-conjugate modulo a positive integer m, if there exists an integer j, such that

$$p^j \equiv -1 (\mathrm{mod}\, m')$$

where m' is the p-free part of m. In the study of Abelian difference sets, we say that the self-conjugacy assumption is satisfied if every prime divisor of $n = k - \lambda$ is self-conjugate modulo $\exp(G)$.

The self-conjugacy assumption can be better understood via Abelian characters. For a $(v, k, \lambda; n)$ difference set D, viewing D as an element of the group ring $\mathbb{Z}G$ we obtain

$$\chi(D)\overline{\chi(D)} = n \tag{2}$$

for all nonprincipal characters χ of G. We note that $\chi(D)$ is an algebraic integer in a suitable cyclotomic field.

This idea of studying difference sets using character sums is due to Turyn (17). If n is self-conjugate modulo $\exp(G)$, then it can be shown: Equation (2) implies that n is a square, say $n = u^2$, and $\chi(D) = u\xi$, where ξ is a root of unity. Such solutions are called trivial solutions. Thus $\chi(D)$ is determined completely from Eq. (2) under the self-conjugacy assumption. This would then impose necessary conditions on the existence of D.

In the absence of self-conjugacy $\chi(D)$ cannot be easily determined from Eq. (2). This difficulty is the primary reason why McFarland's investigation (37) of Abelian Hadamard difference sets in groups of order $4p^2$, p a prime, was rather tedious and quite involved in the $p \equiv 1 (\mathrm{mod}\, 4)$ case (where self-conjugacy is absent), whereas the case $p \equiv 3 (\mathrm{mod}\, 4)$ in which self-conjugacy was present was easily disposed of (38).

Chan (39) introduced a new approach to deal with the situation without self-conjugacy. In some special situations, Chan showed that Eq. (2) has only the trivial solutions, even when there was no self-conjugacy. Using that, he was able to obtain further restrictions on Abelian groups of the form $\mathbb{Z}_{2pq} \times \mathbb{Z}_{2pq}$, where p, q are distinct primes, that contain Hadamard difference sets. In particular, he showed that Abelian Hadamard difference set in $\mathbb{Z}_{6p} \times \mathbb{Z}_{6p}$ can exist only if $p = 3$ or $p = 13$. Several useful theorems for studying difference sets without self-conjugacy can be found in Ma's work (40) on relative $(n, n, n, 1)$ difference sets.

We have seen that the existence of a difference set D yields the existence of an algebraic integer of a certain absolute value. This gives number theoretic conditions that are the basis for most nonexistence results on difference sets. In this section, we cannot survey even the most important nonexistence results, but we hope that the reader gets an impression how algebraic number theory can be used to obtain necessary conditions for the existence of difference sets.

To start with, let us look at the condition

$$\chi(D)\overline{\chi(D)} = n$$

more closely. If χ is a character of order ω, then this equation holds in $Z[\zeta_\omega]$, the ring of algebraic integers in $Q(\zeta_\omega)$; here $\zeta_\omega = e^{2\pi i/\omega}$ is a primitive ωth root of unity. The ring $Z[\zeta_\omega]$ is a Dedekind domain, that is, we can decompose the ideals $\chi(D)$, $\overline{(\chi(D))}$, and (n) uniquely into prime ideals and obtain

$$\chi(D)\overline{\chi(D)} = \prod_{i=1}^{m} P_i^{e_i}$$

where the P_is are distinct prime ideals. The prime ideal decomposition of n in $Z[\zeta_\omega]$ as well as the action of Galois automorphisms of $Q(\zeta_\omega)$ on these ideals is known:

Result 1. Let p be a prime and ζ_ω a primitive complex ωth root of unity; write $\omega = p^e w'$ where ω' is an integer relatively prime to p. The multiplicative order of p modulo ω' is denoted by f. Let $\Phi(x)$ be the number of positive integers $< x$ that are relatively prime to x. Then the following identity for ideals holds in $Z[\zeta_\omega]$:

$$(p) = (P_1 \cdots P_g)^{\Phi(p^e)}$$

where the P_i's are distinct prime ideals and $g = \Phi(\omega')/f$. If t is an integer relatively prime to p such that $t \equiv p^s \bmod \omega'$, then the Galois automorphism $\zeta_\omega \mapsto \zeta_\omega^t$ fixes the ideals P_i. If $\omega' = 1$ then $g = 1$ and $P_1 = (1 - \zeta_\omega)$.

Result 1 shows that the self-conjugacy of a prime p modulo w implies that all prime ideal divisors of p in $\mathbb{Z}[\zeta_\omega]$ are fixed by complex conjugation. This is basically the content of the so called Mann-test.

Corollary 1. Let p be self-conjugate modulo w, and let D be a difference set of order n in a group G whose exponent is divisible by w. Then p cannot divide the square-free part of n, that is, p^{2a} is the exact p power dividing n. In particular, for each character χ of order w, we have $\chi(D) \equiv \bmod p^a$.

As an example, there are no Abelian (25, 9, 3; 6) difference sets: We take $w = 5$ and $p = 3$, then $p^2 \equiv -1 \bmod w$ and hence the Galois automorphism $\zeta_5 \mapsto \zeta_5^3 = \overline{\zeta_5}$ fixes the ideal divisors of (3) in $Z[\zeta_5]$.

Note that the proof of this corollary just uses the prime ideal factorization of (p) in $Z[\zeta_\omega]$. To obtain stronger results, we must exploit the condition $\chi(D) \equiv 0 \bmod p^a$ more carefully. In this context, the following lemma is useful (41).

Lemma 3. Let p be a prime and let G be an Abelian group with a cyclic Sylow p-subgroup of order p^s. If $Y \in Z[G]$ is an element such that $\chi(Y) \equiv 0 \bmod p^a$ for all characters, then we can write

$$Y = p^a X_0 + p^{a-1} P_1 X_1 + \cdots + p^{a-r} P_r X_r$$

$r = \min(a, s)$, where P_i denotes the unique subgroup of order p^i. Moreover, if the coefficients of Y are nonnegative, the coefficients of the X_i can be chosen to be nonnegative, too.

Here is an application: Let D be an Abelian $(4u^2, 2u^2 - u, u^2 - u)$ difference set. Let $u = 2^a$ and assume that G contains a cyclic subgroup of order 2^b. Projection onto this subgroup yields a group ring element Y with $\chi(Y) \equiv 0 \bmod 2^a$ for all characters. The lemma shows that the coefficients of Y are constant modulo 2^a on cosets of subgroup P_1 of order 2. On the other hand, the coefficients are bounded by 2^{2a+2-b}. Since Y cannot be constant on cosets of N, we get $2a + 2 - b \geq a$. This bound is part of Turyn's famous exponent bound for Hadamard difference sets (17).

Another illustration is that there are no Abelian Hadamard difference sets of order p^2 in groups of order $4p^2$ if $p \equiv 3 \bmod 4$ or if the Sylow 2-subgroup is elementary abelian. Note that p is self-conjugate modulo the exponent of G; hence $\chi(D) \equiv 0 \bmod p$ for a putative difference set D. Projection onto the homomorphic image of order $4p$ yields a contradiction similar to the argument above. This (easy) proof is in remarkable contrast to the case $p \equiv 1 \bmod 4$: The nonexistence for those difference sets have been ruled out by McFarland (37) in a long, detailed paper as mentioned earlier. It is one of the first nonexistence results without using self-conjugacy.

Many more nonexistence results are variations of the approach that we have just described: Project the difference sets D in $Z[G]$ onto a group ring $Z[H]$ where H contains a cyclic Sylow p-subgroup and where p is self-conjugate modulo the exponent of H. However, in many situations (such as, for instance, for McFarland difference sets), this approach is only of limited use. The point is that elements in $Z[H]$ with the "correct" character values and "correct" coefficient sizes do exist. In other words, there are elements that "look like" images of difference sets (although the difference set might not exist). But, in general, there are many different subgroups N such that $G/N \cong H$, and the approach described earlier yields information about the image of a putative difference sets under all these projections. Several authors, notably Ma and Schmidt (3), developed some combinatorial group-theoretic tools in order to exploit the information about these different images simultaneously. This method has been applied successfully both to relative and McFarland difference sets.

Recently, two new approaches to prove nonexistence results without any self-conjugacy assumptions have been devised. Schmidt focused his attention on cyclic Hadamard difference sets and Eq. (2). He proved the following.

Result 2. Let Q be a finite set of primes. Then there are (at most) finitely many elements $(a_q)_{q\in Q} \in N^{|Q|}$ such that a cyclic Hadamard difference set of order $\prod_{q\in Q} q^{2aq}$ exists.

A slightly different idea to overcome the self-conjugacy assumption is given by Ma (40). His idea is to get as much information as possible about elements D satisfying Eq. (2). He applies his technique to relative difference sets; in particular he obtains the following strong result on planar functions (we discuss relative difference sets and planar functions later):

Result 3. Given two primes p and q, there are no planar functions on cyclic groups of order pq.

The special case that p is self-conjugate modulo q is comparatively easy.

Arasu and Ma (42) used similar methods to investigate McFarland difference sets without the self-conjugacy assumption.

Schmidt (43) introduces further techniques to deal with difference sets without self-conjugacy. He uses properties of the decomposition group of the prime ideal divisors of the or-

der of the difference set, coupled with ideas similar to those of McFarland (37, Sec. 4), to find restrictions on the solutions of Eq. (2). An example of such a result is given in the following special case.

Theorem 9. Let $d = p^a m$, where p is an odd prime and $d > 0$ is an odd integer relatively prime to p. If $X \in \mathbb{Z}[\zeta_d]$ satisfies

$$X\overline{X} = p$$

then with suitable j either $\zeta_d^j X \in \mathbb{Z}[\zeta_m]$ or $X = \pm\zeta_d^j Y$, where Y is a generalized Gauss sum (44).

With the aid of Theorem 9 it is often possible to find all the solutions of Eq. (2). These solutions can then be further examined to obtain necessary conditions on the existence of an Abelian difference set.

RELATIVE DIFFERENCE SETS

Relative difference sets are a generalization of difference sets. Relative difference sets provide constructions of Hadamard matrices and generalized Hadamard matrices that are of interest in various branches of mathematics. It turns out that group-invariant Hadamard matrices (equivalently Hadamard difference sets) are basically the same objects as certain relative difference sets. Similar to ordinary difference sets, relative differences sets yield sequences with interesting autocorrelation properties (45). Certain types of relative difference sets give rise to perfect ternary sequences (46).

Relative difference sets were introduced by Bose (47), although he did not use the term *relative difference sets.* The term *relative difference sets* was coined by Butson (48). Systematic investigations of these are due to Elliott and Butson (49) and Lam (50). A recent survey of these objects can be found in Pott (51). The interplay of relative difference sets, finite geometry, and character theory is the subject matter of the monograph by Pott (16).

A relative (m, n, k, λ) difference set R in a group of G of order mn relative to a normal subgroup N of order n is a k-subset of G with the following properties: the list of quotients $r(r')^{-1}$ with distinct elements $r, r' \in R$ contains each element of $G\backslash N$ exactly λ times. Moreover, no element in N has such a representation. N will be referred to as the forbidden subgroup. Note that each coset of N contains at most one element from R. (The more general divisible difference sets are subsets of G where the number of representations of elements in N is not necessarily 0, but another constant μ.) Easy counting yields

$$k(k-1) = \lambda n(m-1)$$

The obvious inequality $k \leq m$ follows, for otherwise at least one coset of N would contain more than just one element from R.

If $n = 1$, the relative difference sets become ordinary difference sets. A relative difference set is called Abelian, cyclic, etc., if the underlying group G has the respective property. All our results and examples would concern Abelian relative difference sets. A relative difference set R is said to be splitting if $G \cong H \times N$ for some subgroup H (i.e., the forbidden subgroup N must be a direct factor of G).

Example 3. The set of $\{0, 1, 3\}$ is a $(4, 2, 3, 1)$ relative difference set in $\mathbb{Z}_8$ relative to $N = \{0, 4\}$.

Example 4. The set $\{(0, 0), (1, 1), (2, 1)\}$ is a $(3, 3, 3, 1)$ relative set in $\mathbb{Z}_3 \times \mathbb{Z}_3$ relative to $N = \{0\} \times \mathbb{Z}_3$.

A relative difference set is said to be semiregular if $k = n\lambda$; otherwise it is called regular.

The following result on the parameters m, n, k, and λ of relative difference sets follows from the work of Bose and Connor (52). The symbol $(a, b)_p$ is the Hilbert symbol, which takes values $+1$ or -1 according to whether the congruence $ax^2 + by^2 \equiv 1 \pmod{p^r}$ has or has not for every value of r, rational solutions x_r and y_r.

Result 4. Let D be a regular (m, n, k, λ) relative difference set. Then the following holds:

1. If m is even, then $k - n\lambda$ is a square. If moreover $m \equiv 2 \pmod 4$ and n is even, then k is the sum of two squares.
2. If m is odd and n is even, then k is a square and

$$(k - n\lambda, (-1)^{(m-1)/2} n\lambda)_p = 1$$

   for all odd primes p.
3. If both m and n are odd, then

$$(k, (-1)^{(n-1)/2} n)_p (k - n\lambda, (-1)^{(m-1/2} n\lambda)_p = 1$$

   for all odd primes p.

Using the group ring notation we introduced earlier, the definition of relative (m, n, k, λ) difference sets R can be translated into a group ring equation in $\mathbb{Z}_G$:

$$RR^{(-1)} = n + \lambda(G - N) \tag{3}$$

If U is a normal subgroup of G contained in N, we consider the canonical epimorphism from G into G/U. Extending this epimorphism by linearity from $\mathbb{Z}G$ to $\mathbb{Z}[G/U]$ and applying to Eq. (3), we obtain the following.

Lemma 4. Let R be a relative (m, n, k, λ) difference set in G. If U is a normal subgroup of G contained in N, then there exists an $(m, n/u, k, \lambda_u)$ difference set in G/U relative to N/U. In particular, G/N contains an $(m, k, \lambda n)$ difference set.

KNOWN FAMILIES OF RELATIVE DIFFERENCE SETS

Extension of (m, m, m) Difference Sets

There exists a $(p^a, p^a, p^a, 1)$ relative difference set in $(\mathbb{Z}_p)^{2a}$ if p is an odd prime; in $(\mathbb{Z}_4)^a$ relative to $(\mathbb{Z}_2)^a$ if $p = 2$. This gives the following series of relative difference sets which are extensions of (m, m, m) difference sets.

Family I. Relative (p^a, p^b, p^a, p^{a-b}) relative difference sets exist whenever p is a prime.

Splitting relative difference sets with parameters $(n, n, n, 1)$ in $H \times N$ are equivalent to the so-called planar functions $f: H \to N$; see Dembowski and Ostrom (53). The existence of a planar function implies the existence of a projective plane with a certain automorphism group (semiregular automorphism group). In contrast to the case of planar difference sets, planar functions describing non-Desarguesian planes are known. However, in all known cases H and N are elementary Abelian (provided n is odd). It is one of the open problems concerning planar functions, whether this has to be the case. Finally, we mention that the case of even n has been settled completely (at least in the Abelian case): In this case, n has to be a power of 2 and the group has to be as mentioned before.

Menon–Hadamard difference sets of order u^2 give rise to the following two series of relative difference sets.

Family II. Relative $(4u^2, 2, 4u^2, 2u^2)$ difference sets exist whenever difference sets with parameters $(4u^2, 2u^2 \pm u, u^2 \pm u)$ exist. These are known to exist if $u = 2^a 3^b m^2$, where m is any odd integer (see Theorem 2).

Family III. Relative $(8u^2, 2, 8u^2, 4u^2)$ difference sets exist whenever difference sets with parameters $(4u^2, 2u^2 \pm u, u^2 \pm u)$ exist.

Note: Family III contains only nonsplitting examples, because otherwise a group of order $8u^2$ would contain a Hadamard difference set, which is impossible. New examples of semiregular relative difference sets in groups whose order can contain more than two distinct prime factors are explored by Davis, Jedwab, and Mowbray (54) and Arasu and deLauney (55).

Extensions of $(m, m-1, m-2)$ Difference Sets

Any Desarguesian projective plane of order q gives rise to a cyclic relative $(q+1, q-1, q, 1)$ difference set. Thus we get:

Family IV. For any prime power q and any divisor d of $q-1$, relative $(q+1, (q-1)/d, q, d)$ difference sets exist.

Note: Relative difference sets of Family IV will be considered later.

Extension of $((q^{d+1}-1)/(q-1), q^d, q^d - q^{d-1})$ Difference Sets

Complements of the Singer difference sets are difference sets with parameters

$$\left(\frac{q^{d+1}-1}{q-1}, q^d, q^d - q^{d-1}\right)$$

As observed by Bose (47), the above difference sets lift to cyclic relative difference sets, given in the following.

Family V. If q is a prime power, relative difference sets with parameters

$$\left(\frac{q^{d+1}-1}{q-1}, \frac{q-1}{t}, q^d, q^{d-1}t\right)$$

exist for each divisor t of $q-1$.

Note: The case $d = 2$ reduces to Family IV. We have decided to separate these since geometers usually distinguish the planar case (dimension 2) and the general case. If q is even, Family V does not include relative difference sets if the forbidden subgroup has order 2. Arasu et al. (56) later obtained in the following:

Family VI. If q is a power of 2, and d is even, relative difference sets with parameters

$$\left(\frac{q^{d+1}-1}{q-1}, 2, q^d, \frac{q^d - q^{d-1}}{2}\right)$$

exist.

Arasu, Leung, and Ma (57) obtain the following.

Family VII. If q is a power of 2, relative difference sets with parameters

$$\left(q^2+q+1, 2(q-1)q^2, \frac{q}{2}\right)$$

exist. A few further sporadic examples obtained using a computer search led Arasu, Leung, and Ma (57) to the following.

Conjecture. Cyclic $((q^{d+1}-1)/q-1, 2(q-1), q^d, (q^d - q^{d-1})/2(q-1))$ relative difference sets exist if q is a power of 2 and d is a positive odd integer.

Remark. In a forthcoming paper (in preparation), Arasu, Dillon, Leung, and Ma have proved this conjecture.

DIFFERENCE SETS AND PERFECT SEQUENCES

We summarize a few results that relate difference sets and their generalization such as relative/divisible difference sets to perfect and almost perfect sequences. For recent results on this topic, we refer the reader to Jungnickel and Pott (58).

A sequence $(a_i)_{i=0,1,2,\ldots}$ is said to be periodic with period v if $a_i = a_{i+v}$ for all i. A sequence all of whose entries are either $+1$ or -1 is called binary. The (periodic) autocorrelation function C of $(a_i)_{i=0,1,2,\ldots}$ is defined by

$$C(t) = \sum_{i=0}^{v-1} a_i a_{i+t}$$

Since $(C(t))_{t=0,1,2,\ldots}$ is also periodic (if (a_i) is periodic), it suffices to consider the autocorrelation coefficients $C(t)$ for $t = 0, 1, \ldots, (v-1)$. The autocorrelation function measures how much the original sequence differs from its translates. In the binary case, $C(t)$ is the number of agreements of $(a_i)_{i=0,1,\ldots}$ with its translate by a shift of t minus the number of disagreements. Obviously $C(0) = v$. The other autocorrelation coefficients $C(t)$, with $t \neq 0$, are called nontrivial or the off-peak autocorrelation coefficients. We let k denote the number of $+1$ entries in one period of a periodic binary sequence under consideration.

Periodic sequences with good autocorrelation properties are applicable in engineering. The following is easy to prove.

Lemma 5. A periodic binary sequence with period v, k entries $+1$ per period, and a two level autocorrelation function (with all nontrivial autocorrelation coefficients equal to γ) is equivalent to a cyclic $(v, k, \lambda; n)$ difference set, where $\gamma = v - 4(k-\lambda) = v - 4n$. A ± 1-sequence (a_i) of period v is said to be perfect if it has a two-level autocorrelation function, where the off-peak autocorrelation coefficients α are as small in magnitude as possible.

By Lemma 5, sequences with two-level autocorrelation functions correspond to cyclic difference sets. As shown by Jungnickel and Pott (58), the cases $\gamma = 0, \pm 1, \pm 2$ give rise to the following classes of cyclic (v, k, λ) difference sets of order $n = k - \lambda$:

Class I. $(v, v - \sqrt{v}/2, v - 2\sqrt{v}/4)$ of order $v/4$ corresponding to $\gamma = 0$

Class II. $(v, (v - 2\sqrt{v-1})2, (v + 1 - 2\sqrt{2v-1})4)$ of order $(v - 1/4)$ corresponding to $\gamma = 1$

Class III(a). $(v, (v - \sqrt{2-v})2, (v - 2 - 2\sqrt{2-v})4)$ of order $(v + 2)4$ corresponding to $\gamma = -2$

Class III(b). $(v, (v - \sqrt{3v-2})2, (v + 2 - 2\sqrt{3v-2})4)$ of order $(v - 2)4)$ corresponding to $\gamma = -2$

Class IV. $(v, (v - 1)2, (v - 3)4)$ of order $(v + 1)/4$ corresponding to $\gamma = -1$

Lemma 5 shows that the autocorrelation coefficients are always congruent 4 modulo v. Therefore, in order to determine in absolute value the smallest coefficients, we have to distinguish v modulo 4. This yields the four series I–IV: However, in the case v congruent 2 modulo 4, it is possible that the off-peak autocorrelation coefficient is 2 or -2 [which explains the two classes III(a) and III(b)]. Class III(a) provides only the trivial (2, 1, 0) difference set, corresponding to $\gamma = 2$.

Difference sets in Class I are Hadamard difference sets (Family VII) with parameters $(4u^2, 2u^2 - u, u^2 - u)$. The only known cyclic example of such a difference set is the trivial (4, 1, 0) difference set. It is conjectured that there cannot be any others. Turyn (17) ruled out the existence of cyclic Hadamard difference sets of size $4u^2$, for $1 < u < 55$. Schmidt's recent work (5), establishes the following results:

Theorem 10. Assume the existence of a cyclic Hadamard difference set of order $u^2 = \prod_{i=1}^{s} q_i^{e_i}$ where the q_is are distinct odd primes (note that u must be odd). Let

$$b_j = \min\{b : q_i^{\omega_i} \not\equiv \bmod q_j^b \text{ for all } i \neq j\}$$

where ω_i is the multiplicative order of q_i modulo $\prod_{j \neq i} q_j$. For $j = 1, \ldots, s$, define $c_j = \min\{2a_j, b_j - 1\}$. Moreover, let $u' = \prod_{j=1}^{s} q_j^{c_j}$. Then

$$u \leq \sqrt{2}[\sin(\pi/2u')]^{-1}$$

Corollary 2. Let Q be any finite set of odd primes. Then there are only finitely many cyclic Hadamard difference sets of order u^2, where all prime divisors of u are in Q.

Remark: Corollary 2 is already contained in Result 2.

Corollary 3. Cyclic Hadamard difference sets of order $1 < u \leq 2000$ with $u \notin \{165, 231, 1155\}$ do not exist.

The only known Abelian example of difference as in Series II is the (13, 4, 1) difference set. Parameters in series II can be rewritten as $(2u^2 + 2u + 1, u^2, u(u - 1)/2)$. Results by Broughton (59) and Eliahou and Kervaire (60) imply the following.

Result 5. For $3 \leq u \leq 100$, no Abelian difference sets with parameters

$$(2u^2 + 2u + 1, u^2, u(u-1)/2)$$

exist. Hence perfect sequences of Class II and period v do not exist for $13 < v \leq 20201$.

Difference sets with parameters in Class III(b) require $3v - 2$ and $v - 2$ both to be squares. This series contains the trivial (6, 1, 0) difference set. The next two candidates (66, 26, 10) and (902, 425, 200) are both ruled by Theorem 4.18 of Lander (14). The next permissible value of v is 12546. Thus we obtain the following.

Theorem 11. Perfect sequences of Class III(b) and period v do not exist for $6 < v \leq 12545$.

Difference sets of Class IV are called Paley–Hadamard difference sets. Known examples (parametrically) are given by Singer difference sets (Family I, with $q = 2$), the Paley difference sets of Family V, and the examples given in Family VI.

Song and Golomb (61) and Golomb and Song (62) systematically investigate cyclic Paley–Hadamard difference sets. It is conjectured that every such difference set has parameters as in one of the three series earlier above. Golomb and Song (62) prove the following.

Result 6. Assume the existence of a Paley–Hadamard difference set D in a cyclic group of order v, where $v < 10{,}000$. Then v is either of the form $2^m - 1$, or a prime $\equiv 3 \bmod 4$, or the product of two twin primes, with the possible exceptions of $v = 1295, 1599, 1935, 3135, 3439, 4355, 4623, 5775, 7395, 7743, 8227, 8463, 8591, 8835, 9135, 9215, 9423$.

Corollary 4. A perfect sequence of Class IV and period $v < 10{,}000$ exists if and only if v is either of the form $2^m - 1$ or a prime $\equiv 3 \bmod 4$, or the product of two twin primes, with the possible exceptions given in Result 6.

A concrete application of the perfect sequences corresponding to the twin-prime power difference sets to applied optics is explained in Jungnickel and Pott (58).

ALMOST PERFECT SEQUENCES AND DIVISIBLE DIFFERENCE SETS

As we saw in earlier, perfect ± 1-sequences with off-peak autocorrelation value 0 are quite rare. To remedy this situation, one studies the so-called almost perfect sequences (a concept due to Wolfmann (63) in an attempt to obtain further classes of sequences with good correlation properties.

An almost perfect sequence is a ± 1-sequence in which all the off-peak autocorrelation coefficients are as small as possible—with exactly one exception, say $C(g)$. Since $C(g)$ is the only exceptional autocorrelation coefficient, it follows that $g = -g$, forcing the period v to be even and $g = v/2$. The subset of $\mathbb{Z}_v$ that corresponds to an almost perfect sequence is a divisible difference set.

A k-element subset D of a group G of order v relative to a subgroup N of G of order u is called a $(v/u, u, k, \lambda_1, \lambda_2)$ divisible difference set if the list of all differences

$$(d_1 - d_2 : d_1, d_2 \in D, d_1 \neq d_2)$$

contains every element of $N\backslash\{0\}$ exactly λ_1 times and every element of $G\backslash N$ exactly λ_2 times. If $\lambda_1 = 0$, these reduce to relative difference sets of presented earlier.

Using group ring notations, D is a $(v/u, u, k, \lambda_1, \lambda_2)$ divisible difference set in G relative to N if and only if

$$DD^{(-1)} = (k - \lambda_1) + \lambda_1 N + \lambda_2 (G - N) \text{ in } \mathbb{Z}G$$

Bradley and Pott (64) show: almost perfect sequences are equivalent to cyclic divisible difference sets with $u = 2$, of certain types. Let f be the exceptional correlation coefficient $C(v/2)$. Since v is even, we obtain three possible series of almost perfect sequences corresponding to the Classes I, III(a), and III(b) described earlier. Class I:

$$\text{Class I: } v \equiv 0 \bmod 4 : \left(\frac{v}{2}, 2, \frac{v}{2} - \theta, \frac{f+v}{4} - \theta, \frac{v}{4} - \theta\right)$$
$$\text{where } \theta = \sqrt{v+f}/2$$

Class III:

$$v \equiv 2 \bmod 4$$

Class III(a), off-peak autocorrelation 2:

$$\left(\frac{v}{2}, 2, \frac{v}{2} - \theta, \frac{f+v}{4} - \theta, \frac{v-2}{4} - \theta\right), \text{ where } \theta = \tfrac{1}{2}\sqrt{-v+f+4}$$

Class III(b); off-peak autocorrelation -2:

$$\left(\frac{v}{2}, 2, \frac{v}{2} - \theta, \frac{f+v}{4} - \theta, \frac{v+2}{4} - \theta\right), \text{ where } \theta = \tfrac{1}{2}\sqrt{3v+f-4}$$

Let us first consider the Class I. Only the cases $\theta = 0$, 1, and 2 have been investigated systematically so far. For the case $\theta = 0$, we use the following result of Jungnickel (65).

Result 7. If there exists a cyclic $(v/2, 2, v/2, 0, v/4)$ difference set, then $v = 4$.

Hence, we obtain

Theorem 12. If there exists an almost perfect sequence of type I and $\theta = 0$, then $v = 4$.

In case $\theta = 1$ of type I, an infinite family of almost perfect sequences exists: Let D consist of these elements d in the multiplicative group G of $GF(q^2)$, satisfying $\text{tr}(d + d^q) = 1$. Then D is a cyclic relative difference set in G with parameters $(q + 1, q - 1, q, 1)$. Such relative difference are called affine difference sets (9). Projection yields a cyclic relative difference set with parameters

$$\left(q + 1, 2, q, \frac{q-1}{2}\right)$$

if q is odd.

Hence we obtain the following.

Theorem 13. If $v = 2(q + 1)$, where q is a power of an odd prime, then almost perfect sequences of class I with period v and $\theta = 1$ exist.

It is widely conjectured that $(m + 1, 2, m, (m - 1)/2)$ relative difference sets exist only if m is a prime power. This has been verified for m up to 425 by Reuschling (66). Tools required to establish their nonexistence are listed in the following:

Result 8 (16,36). The following integers are multipliers of any cyclic relative difference sets with parameters $(m + 1, 2, m, (m - 1)/2)$:

- m
- If $m = p^k$ is a power of prime p, then p is a multiplier
- If $m = p^i q^j$ is the product of powers of two distinct primes p and q, then p^i and q^j are multipliers

Result 9 (67). Let G be an Abelian group of order $2(m + 1)$. Let t be a multiplier of a putative $(m + 1, 2, m, (m - 1)/2)$ difference set relative to N. If $G\backslash N$ contains elements x and y with $x^t = x$ and $y^q = (m + 1)y$, then G cannot contain a difference set with these parameters.

Result 10 (Mann Test). Let D be a divisible $(v/u, u, k, \lambda_1, \lambda_2)$ difference set in the Abelian group G relative to N. Moreover, let t be a multiplier of D, and let U be a subgroup of G such that G/U has exponent w $(U \neq G)$. Let p be a prime not dividing w such that $tp^f \equiv -1 \bmod w$. Then the following holds:

- If N is not contained in U, then p does not divide the square-free part of $k - \lambda_1$
- If N is contained in U, then p does not divide the square-free part of $k^2 - mn\lambda_2$

Using these tests for $m \leq 1000$ (m odd), nonexistence of cyclic $(m + 1, 2, m, (m - 1)/2)$ relative difference sets has been established for composite m (58), except for the following values of $m = 425, 531, 545, 549, 629, 867, 909$.

The case $m = 425$ has been recently settled by Arasu and Voss (68), using multipliers and intersection numbers.

Almost perfect sequences of Class I with $\theta = 2$ correspond to cyclic divisible difference sets with parameters

$$\left(\frac{v}{2}, 2, \frac{v-4}{2} 2, \frac{v-8}{4}\right)$$

Examples are known for $v = 8$, 12 and 28. Leung et al. (69) show the following.

Result 11. Almost perfect sequences of Class I with $\theta = 2$ and period v exist if and only if $v = 8$, 12, or 28.

Next we consider Class III(a). Here two possibilities arise: $\theta = 0$ when $f = v - 4$ and $\theta = 1$ when $f = v$ [note that it can be shown that $f \equiv v \pmod 4$]. The following parameters series are obtained:

$$\theta = 0 \Rightarrow \left(\frac{v}{2}, 2, \frac{v}{2}, \frac{v-2}{2}, \frac{v-2}{4}\right)$$
$$\theta = 1 \Rightarrow \left(\frac{v}{2}, 2, \frac{v-2}{2}, \frac{v-2}{2}, \frac{v-6}{4}\right)$$

If $\theta = 0$, $k - \lambda_1 = 1$. Arasu et al. (70) studied divisible difference sets with these parameters. The cyclic case has been settled completely in their work, which gives the following.

Result 12. Let D be a cyclic divisible difference set in $\mathbb{Z}_v$ with parameters

$$\left(\frac{v}{2}, 2, \frac{v}{2}, \frac{v-2}{2}, \frac{v-2}{4}\right)$$

Then $v/2$ must be an odd prime p, i.e. $\mathbb{Z}_v \cong \mathbb{Z}_p \times \mathbb{Z}_2$. The set D is, up to complementation and equivalence,

$$D = \{(x, y) \colon x \text{ is a nonzero square in } \mathbb{Z}_p\} \cup \{(0, 0)\}$$

Corollary 5. Almost perfect sequences of Class III(a) and period v with $\theta = 0$ exist if and only if $v/2$ is an odd prime.

When $\theta = 1$, we have $k = \lambda_1$; using the classification by Bose and Connor (52) of these "divisible designs" and projection arguments, Jungnickel and Pott (58) show the following.

Theorem 14. A divisible difference set D with parameters

$$\left(\frac{v}{2}, 2, \frac{v-2}{2}, \frac{v-2}{2}, \frac{v-6}{4}\right)$$

exists in a group G relative to a normal subgroup N if and only if G/N contains a Paley–Hadamard difference set D'. If ϕ denotes the projection epimorphism $G \to G/N$, then the preimage of D' under ϕ is the desired divisible difference set.

Corollary 6. An almost perfect sequence of Class III(a) and period v with $\theta = 1$ exists if and only if a perfect sequence of period $v/2$ with $v \equiv 6 \bmod 8$ exists.

Finally we consider Class III(b). The parameter series looks quite messy; for instance, it contains divisible difference sets with $k - \lambda_1 = 0$, $k - \lambda_1 = 1$, or $\lambda_1 = \lambda_2$. The first two series have been investigated by Bose and Connor (52) and Arasu et al. (70), respectively. The last case corresponds to ordinary difference sets. This series also contains affine difference sets of Bose (47). Further parameters sets are also covered by Class III(b), which apparently do not fall into an infinite class, although they all correspond to almost perfect sequences.

BIBLIOGRAPHY

1. J. Singer, A theorem in finite projective geometry and some applications to number theory, *Trans. Amer. Math. Soc.,* **43**: 377–385, 1938.
2. H. B. Mann, *Addition Theorems,* New York: Wiley, 1965.
3. S. L. Ma and B. Schmidt, A sharp exponent bound for McFarland difference sets with $p = 2$, *J. Combinatorial Theory A,* **80**: 347–352, 1997.
4. S. L. Ma and B. Schmidt, Difference sets corresponding to a class of symmetric designs. *Des., Codes Cryptogr.,* **10**: 223–236, 1997.
5. B. Schmidt, Circulant Hadamard matrices: Overcoming non-self-conjugacy (submitted for publication).
6. B. Schmidt, Cyclotomic integers of prescribed absolute value and the class group (submitted for publication).
7. B. Schmidt, Non-existence results on Chen and Davis-Jedwab difference sets (submitted for publication).
8. D. Jungnickel, Difference sets, in J. H. Dinitz and D. R. Stinson (eds.), *Contemporary Design Theory: A Collection of Surveys,* New York: Wiley, 1992, pp. 241–324.
9. D. Jungnickel, On affine difference sets, *Sankhya A,* **54**: 219–240, 1992.
10. J. A. Davis and J. Jedwab, A survey of Hadamard difference sets, in K. T. Arasu et al. (eds.), *Groups, Difference Sets, and the Monster,* Berlin and New York: de Gruyter, 1996, pp. 145–156.
11. D. Jungnickel and A. Pott, Difference sets: Abelian, in C. J. Colbourn and J. Dinitz (eds.), *The CRC Handbook of Combinatorial Designs,* Boca Raton, FL: CRC Press, 1996, pp. 297–307.
12. T. Beth, D. Jungnickel, and H. Lenz, *Design Theory,* 2nd ed. Cambridge, UK: Cambridge Univ. Press, 1998.
13. M. Hall, Jr., *Combinatorial Theory,* 2nd ed. New York: Wiley, 1986.
14. E. S. Lander, *Symmetric Designs: An Algebraic Approach,* Cambridge, UK: Cambridge Univ. Press, 1983.
15. L. D. Baumert, *Cyclic Difference Sets,* Vol. 182, Lect. Notes Math., New York: Springer, 1971.
16. A. Pott, *Finite Geometry and Character Theory,* Vol. 1601, Lect. Notes Math., Berlin: Springer-Verlag, 1995.
17. R. J. Turyn, Character sums and difference sets, *Pac. J. Math.,* **15**: 319–346, 1965.
18. R. H. Bruck, Difference sets in a finite group, *Trans. Amer. Math. Soc.,* **78**: 464–481, 1955.
19. R. H. Bruck and H. J. Ryser, The non-existence of certain finite projective planes, *Can. J. Math.,* **1**: 88–93, 1949.
20. S. Chowla and H. J. Ryser, Combinatorial problems, *Can. J. Math.,* **2**: 93–99, 1950.
21. M. P. Schutzenberger, A non-existence theorem for an infinite family of symmetrical block designs, *Ann. Eugen.,* **14**: 286–287, 1949.
22. C. J. Colbourn and J. H. Dinitz (eds.), *The CRC Handbook of Combinatorial Designs,* Boca Raton, FL: CRC Press, 1996.
23. B. Gordon, W. H. Mills, and L. R. Welsch, Some new difference sets, *Can. J. Math.,* **14**: 614–625, 1962.
24. R. G. Stanton and D. A. Sprott, A family of difference sets, *Can. J. Math.,* **11**: 73–77, 1958.
25. R. L. McFarland, A family of difference sets in non-cyclic abelian groups, *J. Combinatorial Theory A,* **15**: 1–10, 1973.
26. J. A. Davis and J. Jedwab, Nested Hadamard difference sets, *J. Stat. Plann. Inference,* **62**: 13–20, 1997.
27. T. Spence, A family of difference sets in non-cyclic groups, *J. Combinatorial Theory A,* **22**: 103–106, 1977.

28. J. F. Dillon, Difference sets in 2-groups, in R. L. Ward (ed.), *NSA Mathematical Sciences Meeting,* Ft. George Meade, MD, 1987, pp. 165–172.
29. J. A. Davis and J. Jedwab, A unifying construction of difference sets, *J. Combinatorial Theory, A,* **80**: 13–78, 1997.
30. Y. Q. Chen, On the existence of abelian Hadamard difference sets and a new family of difference sets, *Finite Fields Appl.,* **3**: 234–256, 1997.
31. R. G. Kraemer, Proof of a conjecture on Hadamard 2-groups, *J. Combinatorial Theory A,* **63**: 1–10, 1993.
32. M. Hall, Jr., Cyclic projective planes, *Duke J. Math.,* **14**: 1079–1090, 1947.
33. R. L. McFarland and B. F. Rice, Translates and multipliers of abelian difference sets, *Proc. Amer. Math. Soc.,* **68**: 375–379, 1978.
34. P. K. Menon, Difference sets in abelian groups, *Proc. Amer. Math. Soc.,* **11**: 368–376, 1960.
35. R. L. McFarland, On multipliers of abelian difference sets, Ph.D. thesis, Ohio State Univ., Columbus, 1970.
36. K. T. Arasu and Q. Xiang, Multiplier theorems, *J. Comb. Des.,* **3**: 257–267, 1995.
37. R. L. McFarland, Difference sets in abelian groups of order $4p^2$, *Mitt. Math. Sem. Giessen,* **192**: 1–70, 1989.
38. H. B. Mann and R. L. McFarland, On Hadamard difference sets, in J. N. Srivastava et al. (eds.), *A Survey of Combinatorial Theory,* Amsterdam: North-Holland, 1973, pp. 333–334.
39. W. K. Chan, Necessary conditions for Menon difference sets, *Des., Codes Cryptogr.,* **3**: 147–154, 1993.
40. S. L. Ma, Planar functions, relative difference sets and character theory, *J. Algebra,* **185**: 342–356, 1996.
41. K. T. Arasu et al., Exponent bounds for a family of abelian difference sets, in K. T.Arasu et al. (eds.), *Groups, Difference Sets, and the Monster,* Berlin and New York: de Gruyter, 1996, pp. 129–143.
42. K. T. Arasu and S. L. Ma, Abelian difference sets without self-conjugacy, *Des., Codes Cryptogr.* (to be published).
43. B. Schmidt, On (p^a, p^b, p^a, p^{a-b})-relative difference sets, *J. Algebraic Comb.,* **6**: 279–297, 1997.
44. K. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory,* New York: Springer, 1982.
45. P. V. Kumar, On the existence of square dot-matrix patterns having a specified three-valued periodic correlation function, *IEEE Trans. Inf. Theory,* **34**: 271–277, 1988.
46. K. T. Arasu et al., Relative difference sets with $n = 2$, *Discrete Math.,* **147** (1–3): 1–17, 1995.
47. R. C. Bose, An affine analogue of Singer's theorem, *J. Indian Math. Soc.,* **6**: 1–15, 1942.
48. A. T. Butson, Relations among generalized Hadamard matrices, *Can. J. Math.,* **15**: 42–48, 1963.
49. J. E. H. Elliott and A. T. Butson, Relative difference sets, *Ill. J. Math.,* **10**: 517–531, 1966.
50. C. W. H. Lam, On relative difference sets, *Proc. 7th Manitoba Conf. Numer. Math. Comput.,* 1977, pp. 445–474.
51. A. Pott, A survey on relative difference sets, in K. T. Arasu et al. (eds.), *Groups, Difference Sets and the Monster,* Berlin and New York: de Gruyter, 1996, pp. 195–232.
52. R. C. Bose and W. S. Connor, Combinatorial properties of group divisible incomplete block designs, *Ann. Math. Stat.,* **23**: 367–383, 1952.
53. P. Dembowski and T. G. Ostrom, Planes of order n with collineation groups of order n^2, *Math. Z.,* **103**: 239–258, 1968.
54. J. A. Davis, J. Jedwab, and M. Mowbray, New families of semiregular relative difference sets, *Des., Codes Cryptogr.,* **13**: 131–146, 1993.
55. K. T. Arasu and W. deLauney, Two dimensional perfect quaternary arrays, manuscript, 1998.
56. K. T. Arasu et al., The solution of the Waterloo problem, *J. Combinatorial Theory A,* **17**: 316–331, 1995.
57. K. T. Arasu, K. H. Leung, and S. L. Ma, Cyclic relative difference sets with classical parameters, manuscript, 1998.
58. D. Jungnickel and A. Pott, Perfect and almost perfect sequences, *Discrete Appl. Math.,* 1998 (to be published).
59. W. J. Broughton, A note on table 1 of "Barker sequences and difference sets," *L'Ens. Math.,* **50**: 105–107, 1994.
60. S. Eliahou and M. Kervaire, Barker sequences and difference sets, *L'Ens. Math.,* **38**: 345–382, 1992.
61. H Y. Song and S. W. Golomb, On the existence of cyclic Hadamard difference sets, *IEEE Trans. Inf. Theory,* **40**: 1266–1268, 1994.
62. S. W. Golomb and H. Y. Song, A conjecture on the existence of cyclic Hadamard difference sets, *J. Stat. Plann. Inference,* **62**: 39–42, 1997.
63. J. Wolfmann, Almost perfect autocorrelation sequences, *IEEE Trans. Inf. Theory,* **38**: 1412–1418, 1992.
64. S. P. Bradley and A. Pott, Existence and non-existence of almost-perfect autocorrelation sequences, *IEEE Trans. Inf. Theory,* **41**: 301–304, 1995.
65. D. Jungnickel, On a theorem of Ganley, *Graphs Comb.,* **3** (2): 141–143, 1987.
66. D. Reuschling, Zur Existenz von ω-zirkulanten Konferenz-Matrizen, Diplomarbeit, Universität Augsburg, 1994.
67. P. Langevin, Almost perfect binary functions. *Applicable Algebra, Algorithms and Error-Correcting Codes,* **4** (2): 95–102, 1993.
68. K. T. Arasu and N. J. Voss, Answering a question of Pott on almost perfect sequences, submitted, 1998.
69. K. H. Leung et al., Almost perfect sequences with $\theta = 2$, *Arch. Math.,* **70** (2): 128–131, 1998.
70. K. T. Arasu, D. Jungnickel, and A. Pott, Symmetric divisible designs with $k - \lambda_1 = 1$, *Discrete Math.,* **97** (1–3): 25–38, 1991.

Reading List

K. T. Arasu, S. L. Ma, and N. J. Voss, On a class of almost perfect sequences, *J. Algebra,* **192**: 641–650, 1997.

J. A. Davis and J. Jedwab, Some recent developments in difference sets, in K. Quinn et al. (eds.), *Combinatorical Designs and Applications,* London: Addison-Wesley (to be published).

J. F. Dillon, Difference sets in 2-groups. *Contemp. Math.,* **111**: 65–72, 1990.

S. Eliahou, M. Kervaire, and B. Saffari, A new restriction on the lengths of Golay complementary sequences, *J. Combinatorial Theory A,* **55**: 49–59, 1990.

J. Jedwab, Generalized perfect arrays and Menon difference sets, *Des., Codes Cryptogr.,* **2**: 19–68, 1992.

J. Jedwab et al., Perfect binary arrays and difference sets, *Discrete Math.,* **125** (1–3): 241–254, 1994.

D. Jungnickel, On automorphism groups of divisible designs, *Can. J. Math.,* **24**: 257–297, 1982.

D. Jungnickel and B. Schmidt, Difference sets: An update, *London Math. Soc. Lect. Notes,* **245**: 89–112, 1997.

R. E. Kibler, A summary of non-cyclic difference sets, $k \leq 20$, *J. Combinatorial Theory A,* **25**: 62–67, 1978.

L. E. Kopilovich, Difference sets in non-cyclic abelian groups, *Kibernetika,* **2**: 20–23, 1989.

P. K. Menon, On difference sets whose parameters satisfy a certain relation, *Proc. Amer. Math. Soc.,* **13**: 739–745, 1962.

K. W. Smith, A table no non-abelian difference sets, in C. J. Colbourn and J. H. Dinitz (eds.), *CRC Handbook of Combinatorial Designs,* Boca Raton, FL: CRC Press, 1996, pp. 308–312.

J. Storer, *Cyclotomy and Difference Sets,* Chicago: Markham, 1967.

R. Wilson and Q. Xiang, Constructions of Hadamard difference sets, *J. Combinatorial Theory A,* **77**: 148–160, 1997.

M. Y. Xia, Some infinite classes of special Williamson matrices and difference sets, *J. Combinatorial Theory A,* **61**: 230–242, 1992.

K. Yamamoto, Decomposition fields of difference sets, *Pac. J. Math.,* **13**: 337–352, 1963.

K. T. ARASU
Wright State University
ALEXANDER POTT
University of Augsburg

THEORY OF FILTERING. See FILTERING THEORY.
THEORY OF NUMBERS. See NUMBER THEORY.
THEORY OF PROGRAMMING. See PROGRAMMING THEORY.
THEORY OF RELIABILITY. See RELIABILITY THEORY.
THERAPY, ABLATION. See HYPOTHERMIA THERAPY.
THERAPY, HYPOTHERMIA. See HYPOTHERMIA THERAPY.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Time-Domain Analysis
Standard Article

F. L. Teixeira[1]
[1]Massachusetts Institute of Technology, Cambridge, MA

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2459
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (418K)

Abstract

The sections in this article are

Electromagnetic Wave Propagation

Time-Domain Differential-Equation Modeling

Time Integration Schemes

Convolution and Recursive Functions

Time-Domain Integral-Equation Methods

Current Issues

| | |

Copyright © 1999-2008 All Rights Reserved.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*
Copyright © 1999 John Wiley & Sons, Inc.

TIME-DOMAIN ANALYSIS

The study of many linear physical phenomena and processes, including electromagnetic wave radiation, propagation, and scattering, can in principle be done either in the *time domain* or in the *frequency domain*.

In time-domain analysis, the study is done by considering the time as an independent variable. Fields, signals, and systems are considered as explicit functions of time and mathematically described through their natural evolution from the past towards the future. In such time-domain description, time can be treated either as a continuous variable (*continuous time*) or as a discrete variable (*discrete time*) (1,2). Traditionally, continuous-time problems are usually associated with electric circuits, communication, and physics problems, while discrete-time problems are usually associated with time-series analysis, statistical problems, and numerical analysis. However, with the advent of high-speed digital computers, it has been increasingly advantageous to treat phenomena that occur in the time continuum as discrete-time processes through some process of sampling. This has led to an increasing connection between continuous time-domain and discrete time-domain analysis.

Frequency-domain analysis, on the other hand, is done by treating the frequency as an explicit variable. This is because physical processes and signals described in the time domain can often (advantageously) be *spectrally* (Fourier) decomposed, i.e., decomposed into a (possibly infinite) sum of elementary modes or signals, each one with a definite frequency (periodicity). These elementary components are called spectral (Fourier) components. The relative weights of these spectral components in the sum correspond to an alternative (dual) description of the original signal, where now the frequency is taken as an explicit variable. Such spectral decomposition is useful because many signals allow for a much simpler representation in the frequency domain than in the time domain.

Mathematically, the passage from the time domain to the frequency domain is done through the a Fourier transform pair [1,2,3,4,5]. Given a time-domain function $\phi(t)$, its Fourier transform $\tilde{\Phi}(\omega)$ is defined by

$$\tilde{\phi}(\omega) = \mathcal{F}[\phi(t)] = \int_{-\infty}^{+\infty} dt\, e^{i\omega t}\phi(t) \tag{1a}$$

Furthermore, the original time-domain function $\phi(t)$ can be easily reconstructed from $\tilde{\Phi}$ through the use of the *inverse* Fourier transform

$$\phi(t) = \mathcal{F}^{-1}[\tilde{\phi}(\omega)] = \frac{1}{2\pi}\int_{-\infty}^{+\infty} d\omega\, e^{-i\omega t}\tilde{\phi}(\omega) \tag{1b}$$

In the above expressions, we have used the so-called $e^{-i\omega t}$ convention (3,4,5), common in optics and physics. An alternative convention is the $e^{j\omega t}$ (1,2), common in circuit analysis and signal processing. One can go from one convention to the other by simply replacing i with $-j$ in all expressions or vice versa.

2 TIME-DOMAIN ANALYSIS

The Fourier transform pair establishes a one-to-one relationship between the time-domain and frequency-domain descriptions. There are some mathematical conditions to observe on the function $\phi(t)$ for its Fourier transform to exist. These conditions are related to the convergence of the integrals in Eqs. (1a). Discussion of those aspects is beyond the scope of this article. The interested reader may consult, for example, Refs. 1,2,3. Suffice is to say that, for most functions of interest in practice, a Fourier transform pair is well defined.

Many physical processes of interest are described by linear time-invariant (LTI) differential equations. For instance, Maxwell's equations, which govern the behavior of the electromagnetic fields, are written as (4,5)

$$\mu(\boldsymbol{r})\frac{\partial \boldsymbol{H}}{\partial t} = -\nabla \times \boldsymbol{E} \tag{2a}$$

$$\epsilon(\boldsymbol{r})\frac{\partial \boldsymbol{E}}{\partial t} + \boldsymbol{J} = \nabla \times \boldsymbol{H} \tag{2b}$$

$$\nabla \cdot \boldsymbol{E} = \frac{\rho}{\epsilon}(\boldsymbol{r}) \tag{2c}$$

$$\nabla \cdot \boldsymbol{H} = 0 \tag{2d}$$

where $\boldsymbol{E}$ is the electric field, $\boldsymbol{H}$ is the magnetic field, $\boldsymbol{J}$ is the electric current density, and ρ the electric charge density. If $\epsilon(\boldsymbol{r})$ and $\mu(\boldsymbol{r})$ are functions only of position, the above equations constitute a linear, time-invariant system. *Linearity* means that if we take two set of possible field solutions of Eq. (2a), say $\{\boldsymbol{E}_1, \boldsymbol{H}_1\}$ and $\{\boldsymbol{E}_2, \boldsymbol{H}_2\}$, then any linear combination $\{\alpha\boldsymbol{E}_1+\beta\boldsymbol{E}_2, \alpha\boldsymbol{H}_1+\beta\boldsymbol{H}_2\}$ will still be a solution of those equations (principle of superposition). Time invariance means that if we consider a set of excitations (sources) $\{\boldsymbol{J}(t), \rho(t)\}$ and corresponding solutions (responses) $\{\boldsymbol{E}(t), \boldsymbol{H}(t)\}$, and shift the excitations (sources) by an arbitrary amount of time τ, that is, $\{\boldsymbol{J}(t-\tau), \rho(t-\tau)\}$, then the new corresponding solutions are the original solutions shifted by the same amount of time, that is, $\{\boldsymbol{E}(t-\tau), \boldsymbol{H}(t-\tau)\}$.

Differential equations describing LTI systems in the time domain can be easily translated to the frequency domain by replacing the temporal derivative with the algebraic operator $-i\omega$. Time integrals can be similarly replaced by algebraic operators. As a result, the analytical treatment of LTI differential equations (such as Maxwell's equations) or LTI integral equations is significantly simplified when described in the frequency domain. This constitutes a major reason for the popularity for analytical treatments in the frequency domain.

For a discrete-time signal $\phi(n)$, $\infty < n < \infty$, where n is an integer, a Fourier representation is also possible, in the form (2)

$$\phi(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \tilde{\phi}(e^{j\omega})e^{j\omega n}\,d\omega \tag{3a}$$

where we have used the $e^{j\omega t}$ ($e^{j\omega n}$ for the discrete time-domain case) convention mentioned previously, instead. In the above, $\tilde{\Phi}$) is given by

$$\tilde{\phi}(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \phi(n)e^{-j\omega n} \tag{3b}$$

Equations (3a) constitute the *discrete Fourier transform* (*DFT*) pair. In this case, the Fourier transform $\tilde{\Phi}(e^{j\omega})$ is a periodic function with period 2π. The DFT plays an important role in the analysis and design of discrete-time signal-processing algorithms and systems, analogous to the role of the Fourier transform for continuous-time systems. Also important is the fact that very efficient algorithms, collectively known as *fast Fourier transforms* (*FFT*s), exist to compute the DFT (2). It is interesting to observe the correspondence with the continuous-time case: the summation in Eq. (5b) is simply the Fourier *series* of a periodic signal $\tilde{\Phi}(e^{j\omega})$, whereas the integral for the "coefficients" $\phi(n)$ in Eq. (5a) is just the integral that would be used to obtain the coefficients of the Fourier series. Of course, the discrete-time treatment can nevertheless be carried out independently of this correspondence. Another connection between the continuous-time representation and the discrete-time representation is given by the celebrated *Nyquist sampling theorem*, which states that a band-limited signal $\phi(t)$ whose Fourier transform $\tilde{\Phi}(\omega) = 0$ for $\omega > \omega_{\max}$ is uniquely determined by its samples (i.e., a discrete-time representation), $\phi(n) = \phi(n\ \Delta t)$, $-\infty < n < \infty$, if $\omega_s = 2\ \pi/\ \Delta t > 2\omega_{\max}$. The frequency $\omega_{\max}$ is usually referred as the *Nyquist frequency*, and the frequency ω_s as the Nyquist rate. Therefore, a band-limited signal can be reconstructed exactly from discrete samples. It is possible to generalize both the Fourier transform and the DFT to fully exploit the theory of complex variables in the analysis of the time-domain signals. In the case of continuous-time systems, this generalization is given by the *Laplace* transform (1), and for discrete-time systems it is given by the z transform (2).

Apart from its advantages in analytical treatment, the popularity of frequency-domain analysis also stems from the fact that many measurement apparatuses are confined to frequency-domain measurements. With the advent of high-speed digital computers, however, the strong prevalence of frequency-domain analysis in comparison with time-domain analysis has declined. Digital computers altered the way problems can be solved and paved the way to new algorithms and increasing popularity of time-domain analysis (6,7,8,9,10). Moreover, they also promoted dramatic changes in the measurement hardware. Some of the specific reasons for the increasing popularity of time-domain analysis in the digital age are:

(1) Many problems of interest are actually nonlinear or time-variant. In those cases, time-domain analysis provides a more direct and straightforward modeling.
(2) In some cases, fewer arithmetic operations are required in the time domain than in the frequency domain. For instance, for broadband problems, time-domain analysis is more efficient because the problem is localized in the time domain (e.g., short duration times) but not in the frequency domain (large bandwidths).
(3) It is often easier to get frequency-domain results from time-domain data than vice versa.
(4) Time-domain analysis is conceptually closer to our intuition, which develops in the space–time arena.

In this article, we will describe the basics of time-domain analysis as applied to Maxwell's equations, which govern the behavior of electromagnetic waves. Because of their pervasiveness in electrical engineering, Maxwell's equations will provide a general ground for the discussion of time-domain algorithms. A more detailed discussion of time-domain algorithms in relation to electric circuit and network analysis is presented in the article Time-Domain Network Analysis

Electromagnetic Wave Propagation

The equations governing the electromagnetic field are given by Eq. (2a). If we take a source-free case, $\boldsymbol{J} = \boldsymbol{0}$ and $\rho = 0$, substitute Eq. (2a) into Eq. (2b), and solve for $\boldsymbol{E}$, we obtain

$$\nabla \times \nabla \times \boldsymbol{E} - \mu\epsilon \frac{\partial^2}{\partial t^2} \boldsymbol{E} = \boldsymbol{0} \tag{4}$$

By making use of the identity

$$\nabla \times \nabla \times \boldsymbol{E} = \nabla^2 \boldsymbol{E} - \nabla \nabla \cdot \boldsymbol{E} \tag{5}$$

and of the divergence-free condition for the electric field in a source-free region,

$$\nabla \cdot \boldsymbol{E} = 0 \tag{6}$$

we obtain

$$\nabla^2 \boldsymbol{E} - \mu\epsilon \frac{\partial^2}{\partial t^2} \boldsymbol{E} = \boldsymbol{0} \tag{7}$$

which is know as the *vector wave equation*. The speed of propagation is given by $v = (\mu\epsilon)^{-1/2}$ and depends on the background medium. In vacuum, $v = c = (\mu_0\epsilon_0)^{-1/2} \approx 2.998 \times 10^8$ m/s.

In Cartesian coordinates, the Laplacian operator, ∇^2, is written as

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \tag{8}$$

Furthermore, we can write $\boldsymbol{E}$ in terms of its field components, i.e., $\boldsymbol{E} = \hat{\boldsymbol{x}} E_x + \hat{\boldsymbol{y}} E_y + \hat{\boldsymbol{z}} E_z$, where $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$ are the Cartesian unit vectors. The vector wave equation can therefore be written as three *scalar wave equations*:

$$\nabla^2 E_x - \frac{1}{v^2} \frac{\partial^2}{\partial t^2} E_x = 0 \tag{9}$$

and the same for E_y and E_z. Although the field components appear decoupled in these three equations, they are nevertheless coupled through the divergence-free condition, $\nabla \cdot \boldsymbol{E} = 0$.

The wave-propagating nature of the solutions of Eq. (9) can be easily understood by considering, for simplicity, the one-dimensional case

$$\frac{\partial^2}{\partial x^2} E_x - \frac{1}{v^2} \frac{\partial^2}{\partial t^2} E_x = 0 \tag{10}$$

and considering the *trial* solutions $E_x(x,t) = F^+(x - vt) + F^-(x + vt)$, where $F^\pm$ are twice differentiable functions. It is a simple exercise to verify that this functions satisfy Eq. (10), regardless of the specific functional choice for $F^\pm$. The functions $F^\pm$ are known as propagating wave functions (also called D'Alembert solutions) because they represent a traveling function propagating with speed v in the $+x$ (F^+) and $-x$ (F^-) directions, respectively. The above solutions establishes the propagating-wave nature of the solutions of Maxwell's equations.

The wave equation is a linear second-order partial differential equation: only derivatives up to second order are present. Linear second-order partial differential equations can be classified into three classes, according to the coefficients that multiply the highest-order terms (11). The wave equation is the prototypical *hyperbolic* differential equation. The other classes are the *elliptic* and the *parabolic*. Elliptic equations are associated with steady-state phenomena (boundary-value problems), do not involve time evolution, and therefore will not be

considered here. The prototypical elliptic equation is the Laplace equation, i.e., $\nabla^2\phi = 0$. Parabolic equations are associated with diffusion phenomena, such as heat diffusion (heat equation). They can also occur in the context of electromagnetics, as for instance, in a medium with high conductivity. In that case, Eq. (16b) becomes

$$\frac{\partial^2}{\partial x^2}E_x - \frac{1}{v^2}\frac{\partial^2}{\partial t^2}E_x - \mu\sigma\frac{\partial}{\partial t}E_x = 0 \tag{11a}$$

If the conductivity is sufficiently large so that displacement currents can be neglected, the second term above can be dropped and we have

$$\mu\sigma\frac{\partial}{\partial t}E_x = \frac{\partial^2}{\partial x^2}E_x \tag{11b}$$

which is the diffusion equation, the same parabolic equation that governs heat flow.

Both parabolic and hyperbolic equations are evolutionary equations, which undergo change as a function of time and to which time-domain analysis applies. Parabolic and hyperbolic equations are usually solved through similar techniques, as opposed to elliptic equations.

Note that the passage from the time domain to the frequency domain corresponds to replacing the wave equation (4) by the so-called *Helmholtz equation,*

$$\nabla^2 E_x + \omega^2\mu\epsilon E_x = 0 \tag{12}$$

which is an elliptic equation with no time evolution present.

Time-Domain Differential-Equation Modeling

The time-domain analysis of differential equations using numerical methods on a computer involves a *discretization* procedure, whereby the infinite number of degrees of freedom of the original model is reduced to a finite number, tractable by the computer. For instance, the passage from a continuous-time representation to a discrete-time representation (sampling) mentioned before is an example of a discretization procedure. The spatial discretization of partial differential equations requires a meshing of the space, whereby the spatial domain is replaced by a lattice of discrete points or a set of elementary discrete domains (facets, volumes, etc.). The discretization constitutes the central step in the modeling of differential equations.

When referring to the spatial discretization, three major families of techniques can be identified for differential-equation time-domain modeling. The first are the *finite-difference* techniques, which consists in replacing the derivatives (both temporal and spatial) present in the differential equations with finite-difference approximations. Finite-difference methods are simple to implement and well suited for modeling time-domain problems in moderately complex domains. The second family are the *finite-volume* techniques, where (spatial) local integral relations are derived for the field quantities and discretized through the use of elementary contours, surfaces, and volumes. The third family are the *finite-element* techniques, usually associated with variational formulations and where the fields are projected on the space of some compactly supported basis (interpolatory) functions, usually defined over finite domains (so-called *elements*). Finite elements are better suited for problems involving very complex geometries. They can also provide very accurate error estimates.

There are strong conceptual links between finite-difference, finite-volume, and finite-element techniques (12,13), and in some instances the distinctions between them can become quite blurred. For example, finite-

difference techniques can be recast as particular (point-matched) finite-element techniques with some specific choices for the basis functions. Moreover, the finite-element method can be used as a systematic way to produce more complex finite-difference methods with sharp error estimates. The finite-volume technique can often be also reinterpreted as a finite-difference technique over irregular grids.

Besides these three families of techniques, other families can also be identified. (Pseudo) *spectral methods* (14,15), for example, can be identified as a class of finite methods with global, smooth (for example, Chebyshev or sinusoidal) basis functions, that is, functions defined over the entire spatial domain, instead of compactly supported as in the finite-element method. Pseudospectral methods can be seen as limiting cases of increasing-order finite-difference methods. Another popular time-domain method for Maxwell's equation is the *transmission-line method* (TLM) (16), closely related to the finite-difference method.

When referring to the temporal discretization of hyperbolic equations, the most relevant distinction is between *explicit* and *implicit* techniques. In explicit techniques, the values of the fields at a given instant of time depend only on the previous instant of time. Therefore, they can be written *explicitly* as a function of previously known values in a time-update scheme. In implicit methods, the field values at some instant of time depend not only on field values at previous instants, but also on field values at the same instant of time (and distinct points of space). Therefore, they cannot be written explicitly as a function of previously known field values. Implicit methods require the solution of a linear system at each time step, as opposed to explicit methods. Although for differential-equation modeling the associated matrix is usually sparse, this can be a considerable computational burden. On the other hand, implicit methods are superior in their numerical stability, allowing for larger time discretization steps. This will be discussed in more detail later on.

Finite-Difference Time-Domain Methods. The finite-difference time-domain (FDTD) modeling (8,9, 16) of differential equations replaces derivatives by finite differences. It starts by assuming some field quantity, say $\psi(x,t)$, to exist only at discrete points of space and time separated by fixed intervals Δx and Δt and labeled by the indices m, n that is, $\psi^{m,n} = \psi(m\ \Delta x, n\ \Delta t)$. In general, the intervals do not need to be of equal length, but here they will be assumed so for the sake of simplicity. If $\psi(x,t)$ satisfies a one-dimensional scalar wave equation of the form

$$\frac{1}{v^2}\frac{\partial^2 \psi}{\partial t^2} = \frac{\partial^2 \psi}{\partial x^2} \tag{13}$$

then the spatial derivative can be approximated as

$$\left.\frac{\partial \psi}{\partial x}\right|_{x=m\,\Delta x} \approx \frac{1}{\Delta x}\left(\psi^{m+1,n} - \psi^{m,n}\right) \tag{14a}$$

which corresponds to forward differencing. Alternatively, the spatial derivative can also be approximated as

$$\left.\frac{\partial \psi}{\partial x}\right|_{x=m\,\Delta x} \approx \frac{1}{\Delta x}\left(\psi^{m,n} - \psi^{m-1,n}\right) \tag{14b}$$

or

$$\left.\frac{\partial \psi}{\partial x}\right|_{x=m\,\Delta x} \approx \frac{1}{2\,\Delta x}\left(\psi^{m+1,n} - \psi^{m-1,n}\right) \tag{14c}$$

which correspond to backward and central differencing, respectively. For a second derivative, central differencing can be applied twice,

$$\left.\frac{\partial^2\psi}{\partial x^2}\right|_{x=m\,\Delta x} \approx \frac{1}{\Delta x^2}\left(\psi^{m+1,n} - 2\psi^{m,n} + \psi^{m-1,n}\right) \tag{15}$$

A more systematic way to derive the finite-difference approximation is to expand the functions using a Taylor expansion for $\psi^{m+1,n}$ and $\psi^{m-1,n}$ around $x = m\ \Delta x$, that is, in terms of $\psi^{m,n}$ and its derivatives:

$$\psi^{m+1,n} = \psi^{m,n} + \Delta x\ \frac{\partial\psi}{\partial x}\bigg|^{m,n} + \frac{1}{2}\,\Delta x^2\ \frac{\partial^2\psi}{\partial x^2}\bigg|^{m,n} + \frac{1}{6}\,\Delta x^3\ \frac{\partial^3\psi}{\partial x^3}\bigg|^{m,n} + O(\Delta x^4) \tag{16a}$$

$$\psi^{m-1,n} = \psi^{m,n} - \Delta x\ \frac{\partial\psi}{\partial x}\bigg|^{m,n} + \frac{1}{2}\,\Delta x^2\ \frac{\partial^2\psi}{\partial x^2}\bigg|^{m,n} - \frac{1}{6}\,\Delta x^3\ \frac{\partial^3\psi}{\partial x^3}\bigg|^{m,n} + O(\Delta x^4) \tag{16b}$$

By adding the above equations, we obtain the expression for the second derivative,

$$\left.\frac{\partial^2\psi}{\partial x^2}\right|_{x=m\,\Delta x} = \frac{1}{\Delta x^2}\left(\psi^{m+1,n} - 2\psi^{m,n} + \psi^{m-1,n}\right) + O(\Delta x^2) \tag{17}$$

which shows that the approximation in Eq. (15) is an approximation of second order, that is, with error $O(\Delta x^2)$. On the other hand, backward differencing and forward differencing both have error $O(\Delta x)$.

In a hyperbolic equation, the solution region in the (x,t) domain is open in the coordinate t, so that a solution advances towards positive t from prescribed *initial conditions* at some specified initial time $t = t_0$. On the other hand, in the spatial domain, at any given instant of time, the solution is known for all x and should satisfy prescribed *boundary conditions*. This is illustrated in Fig. 1. Because we are dealing with an initial-value problem in the temporal domain and with a boundary-value problem in the spatial domain, there is a fundamental difference between the time discretization and the spatial discretization.

If we apply the discretization given by Eq. (15) to the time derivative, and substitute in the wave equation in Eq. (13), we arrive at

$$\psi^{m,n+1} = 2\psi^{m,n} - \psi^{m,n-1} + v^2\,\Delta t^2\ \left.\frac{\partial^2\psi}{\partial x^2}\right|_{t=n\,\Delta t} \tag{18}$$

which is a time-stepping formula, that is, given the knowledge of $\psi^{m,n-1}$ and $\psi^{m,n}$, one can find $\psi^{m,n+1}$. In particular, given the values of $\psi^{m,0}$ and $\psi^{m,1}$, the values of $\psi^{m,n}$ at all subsequent time steps can be determined, as long as the time-stepping scheme is stable. The stability of the time-stepping scheme is related to the relative values of Δx, Δt, and v and will be discussed later on. By combining the time and spatial finite differences, the resulting discrete equation becomes

$$\psi^{m,n+1} = 2\psi^{m,n} - \psi^{m,n-1} + v^2\frac{\Delta t^2}{\Delta x^2}\left(\psi^{m+1,n} - 2\psi^{m,n} + \psi^{m-1,n}\right) \tag{19}$$

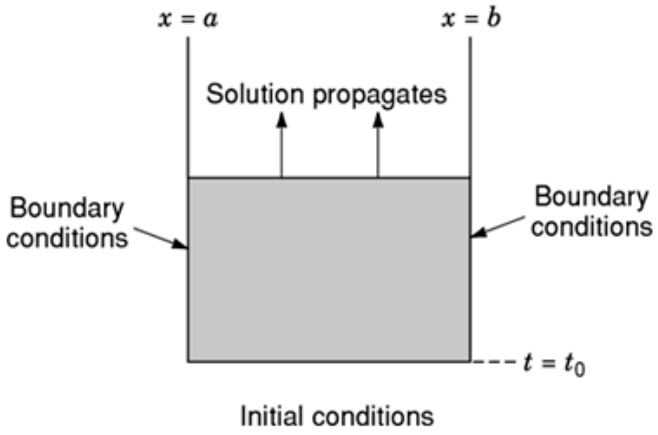


Fig. 1. Time evolution of a hyperbolic partial differential equation in the (x, t) domain, with boundary conditions on x and initial conditions on t.

which can be solved on a computer for some prescribed initial and boundary values.

In the case of Maxwell's equations, we can solve for one of the fields first and discretize the resulting equation, for example the second-order wave equation for $\boldsymbol{E}$, Eq. (12). Otherwise, we can work directly with the first-order curl equations, Eqs. (2a) and (2b), and both fields, $\boldsymbol{E}$ and $\boldsymbol{H}$, simultaneously. Yee's scheme (8,9) is a very popular FDTD scheme to discretize the first-order Maxwell's curl equations in Cartesian coordinates. In terms of components, Eq. (2a) is written as

$$-\mu \frac{\partial H_x}{\partial t} = \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \tag{20a}$$

$$-\mu \frac{\partial H_y}{\partial t} = \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \tag{20b}$$

$$-\mu \frac{\partial H_z}{\partial t} = \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \tag{20c}$$

Yee's FDTD discretization scheme for the above equations starts by spatially staggering the electric and magnetic field components and replacing the spatial derivatives by central differences. The staggering of electric and magnetic fields and the location of each field component are illustrated in Fig. 2. This is equivalent to defining the electric and magnetic field components over different (dual) grids, staggered with respect to each other. A temporal staggering is also used for the electric and magnetic fields in the time evolution. The temporal staggering is such the magnetic field at a time $t = (n + \frac{1}{2})\,\Delta t$ is obtained from the electric field at an instant of time $t = n\,\Delta t$. The electric field at $t = (n + 1)\,\Delta t$ is then obtained from the magnetic field at $t = (n + \frac{1}{2})\,\Delta t$, and the scheme is iterated. Such a time update scheme is usually known as a *leapfrog* scheme. By denoting the field components using $\psi(i\,\Delta x, j\,\Delta y, k\,\Delta z, n\,\Delta t) = \psi^n{}_{i,j,k}$, then the FDTD discretization for Eqs.

(20a) become

$$-\mu\frac{1}{\Delta t}\left(H^{n+\frac{1}{2}}_{x,i,j+\frac{1}{2},k+\frac{1}{2}} - H^{n-\frac{1}{2}}_{x,i,j+\frac{1}{2},k+\frac{1}{2}}\right) = \frac{1}{\Delta y}\left(E^{n}_{z,i,j+1,k+\frac{1}{2}} - E^{n}_{z,i,j,k+\frac{1}{2}}\right) - \frac{1}{\Delta z}\left(E^{n}_{y,i,j+\frac{1}{2},k+1} - E^{n}_{y,i,j+\frac{1}{2},k}\right) \tag{21a}$$

$$-\mu\frac{1}{\Delta t}\left(H^{n+\frac{1}{2}}_{y,i+\frac{1}{2},j,k+\frac{1}{2}} - H^{n-\frac{1}{2}}_{y,i+\frac{1}{2},j,k+\frac{1}{2}}\right) = \frac{1}{\Delta z}\left(E^{n}_{x,i+\frac{1}{2},j,k+1} - E^{n}_{x,i+\frac{1}{2},j,k}\right) - \frac{1}{\Delta x}\left(E^{n}_{z,i+1,j,k+\frac{1}{2}} - E^{n}_{z,i,j,k+\frac{1}{2}}\right) \tag{21b}$$

$$-\mu\frac{1}{\Delta t}\left(H^{n+\frac{1}{2}}_{z,i+\frac{1}{2},j+\frac{1}{2},k} - H^{n-\frac{1}{2}}_{z,i+\frac{1}{2},j+\frac{1}{2},k}\right) = \frac{1}{\Delta x}\left(E^{n}_{y,i+1,j+\frac{1}{2},k} - E^{n}_{y,i,j+\frac{1}{2},k}\right) - \frac{1}{\Delta y}\left(E^{n}_{x,i+\frac{1}{2},j+1,k} - E^{n}_{x,i+\frac{1}{2},j,k}\right) \tag{21c}$$

An analogous discretization applies for the other curl equation (Ampere's law) by duality (4). Once the $\boldsymbol{E}$ field at $t = 0$ and the $\boldsymbol{H}$ field at $t = -\frac{1}{2}\,\Delta t$ are known (initial conditions), the update equations can be used to find the fields at all future time steps.

Yee's FDTD scheme is an explicit scheme, and for a given choice of Δx, Δy, and Δz, a basic condition applies to the maximum admissible value of Δt to obtain numerically stable time stepping. This is given by the Courant-Friedrichs-Levy (CFL) criterion (4,8,16),

$$v\,\Delta t \leq \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right)^{-1/2} \tag{22}$$

For an inhomogeneous medium, v is a function of position, and in the above, v should be taken as its maximum value in the computational domain. The CFL criteria can be derived through a von Neumann stability analysis (8). We note that this criterion is also related to the causality condition. It essentially states that the time step Δt should be smaller than the shortest travel time for waves between the lattice planes, a requirement of causality. However, implicit schemes, which include the same time-step interactions at different spatial grid points, do allow for the relaxation of the CFL condition.

In the case of a parabolic equation such as Eq. (11b), if forward differencing is used for the first-order temporal derivative and central differencing for the second-order spatial derivative (Euler scheme), the resulting scheme will be stable if $2\,\Delta t \leq (\mu\sigma\,\Delta s^2/\,n)$, where n is the dimension of the problem, and Δs is the

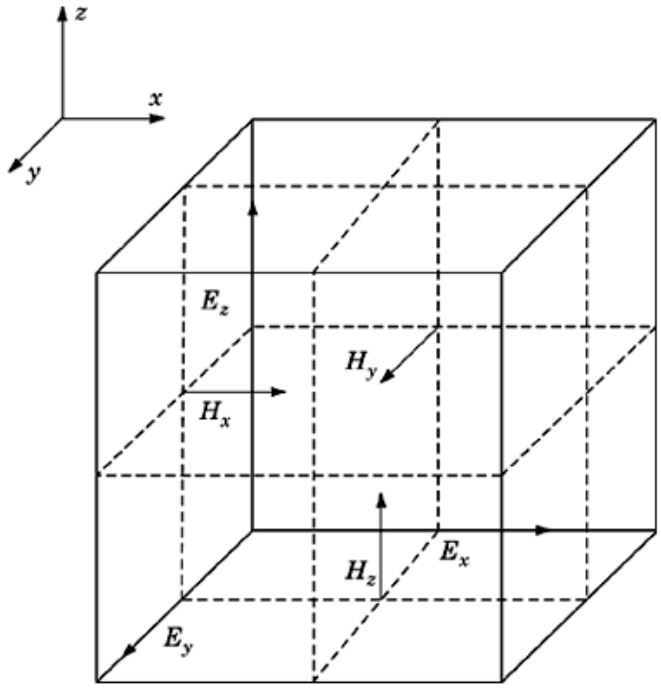


Fig. 2. Yee's elementary staggered FDTD cell, depicting the location of the electric and magnetic field components.

spatial discretization size (assumed uniform) (4). (If central differencing is used for both temporal and spatial derivatives, the update is always unstable.) The right-hand side of this condition is just the time for the field to diffuse between successive planes in the numerical grid. This condition therefore just states that the time step should be smaller than half this diffusion time. In contrast to the discretization of the wave equation, we observe that the computation time for the diffusion equation modeled using FDTD grows as $N^{(n+2)/n}$, where N is the total number of grid points. The computational effort to solve a diffusion problem grows faster with the size of the problem than that in solving the wave equation (the maximum temporal step depends quadratically on the spatial step). This can be physically understood in connection with the difference in how the wavelength (wave equation) and the skin depth (diffusion equation) scale with frequency. One possible way to increase the efficiency of the solution is to use an implicit scheme instead of an explicit one. Another is to add a small wavelike term to the diffusion equation, making it possible to employ a central-differencing scheme for the temporal derivative in which the CFL criterion is always satisfied (unconditionally stable). This must be done without altering much of the physics of the problem (4).

Apart from numerical stability, a second general criterion to be observed in the choice of the discretization size is related to the numerical (or grid) dispersion effects. Numerical dispersion refers to the fact that plane waves do not all propagate with the same phase velocity on the lattice (4,8). Plane waves with different frequencies will have different phase velocities. Moreover, plane waves with the same frequency but different propagation *directions* will also have different phase velocities (due to the anisotropy of the discretized space), this last effect, of course, being present only in two- or three-dimensional problems. As a result of numerical dispersion, a time-domain pulse, which is a linear superposition of plane waves, will be distorted as it propagates through the lattice. To minimize the numerical dispersion error, the spatial discretization size should be chosen small compared to the wavelength (usually between $\lambda/10$ and $\lambda/20$). Alternatively, higher-order finite-difference methods can be employed to reduce the dispersion error (8). Higher-order schemes utilize larger stencils in the

finite-difference approximation for the derivatives. Both alternatives lead to larger computational requirements in memory and CPU time.

In many cases, it is not the numerical dispersion effects but the fine geometrical features of the problem that dictate the maximum values of the spatial discretization steps. In those cases, the use of nonuniform grids, where the spatial discretization size is *locally* reduced to accommodate the fine geometrical details, is often advantageous.

To speed up the time-domain simulation, it is convenient to chose Δt to be as large as possible yet satisfying Eq. (22). As a bonus, it can be shown that, for Yee's scheme, this choice minimizes the numerical dispersion error (8).

Finite-Element Time-Domain Methods. Another popular discretization scheme is the finite-element method (*FEM*) (10,17,18). Although more commonly used for frequency-domain problems (elliptical equations), it has nevertheless been applied with success for time-domain analysis (10). Its major attractiveness is that it is well suited for use with unstructured spatial grids. Finite-element methods can often also be seen as a convenient way to generate complex finite-difference schemes and obtain accurate error estimates. Finite-element schemes are based on general analytical expansion techniques to obtain approximate solutions. They are also usually known as the *Rayleigh–Ritz* method for stationary problems and the method of *weighted residuals* for problems that are posed directly as differential equations (18). Contrary to finite-difference methods, which are based on a finite-point-set approximation to a differential equations, finite-element methods most often utilize piecewise continuous polynomials to expand the unknown functions and generate the discretized version of the equations (in the method of weighted residuals). Hence, instead of discretizing the spatial domain directly, finite-element methods first discretize the function used to represent the solution.

The *point-matched*, or *collocation*, time-domain finite-element method is chosen in the short exposition that follows because of its simplicity and because it already incorporates some of the main features of the time-domain finite-element method. In this scheme, the spatial domain is first subdivided into *finite elements*. In the two-dimensional case, the elements may correspond, for example, to triangles as depicted in Fig. 3. We assume the $\boldsymbol{E}$ and $\boldsymbol{H}$ fields to have the following functional forms:

$$\boldsymbol{E}(\boldsymbol{r}, t) \approx \sum_{i=1}^{M} \phi_i(\boldsymbol{r}) \boldsymbol{E}_i(t) \tag{23a}$$

$$\boldsymbol{H}(\boldsymbol{r}, t) \approx \sum_{j=1}^{N} \psi_j(\boldsymbol{r}) \boldsymbol{H}_j(t) \tag{23b}$$

The functions $\phi_i(\boldsymbol{r})$ and $\psi_j(\boldsymbol{r})$ are called *basis functions* and interpolate the fields within each element using the values of the nodes constituting the element (nodal elements). Alternatively, the degrees of freedom can be associated with edges instead of nodes. In that case, the interpolatory elements are called edge elements. Although nodal elements are simpler to implement, they may produce spurious modes in a finite-element implementation. Proper care should be taken to avoid that when using nodal elements to interpolate $\boldsymbol{H}$ or $\boldsymbol{E}$ (17,18). Edge elements avoid spurious modes because they better mimic the physics of the problem. Using the language of differential forms, $\boldsymbol{H}$ and $\boldsymbol{E}$ are one-forms, while $\boldsymbol{B}$ and $\boldsymbol{D}$ are two-forms. The natural interpolants for one-forms are edge (Whitney) elements. Nodal elements are the natural interpolants for zero-forms (e.g., scalar potentials) only.

The expansion in Eq. (23a) is required to be complete, that is, it can represent any function up to the order of approximation, but the basis functions are not required to be orthogonal. By substituting the above

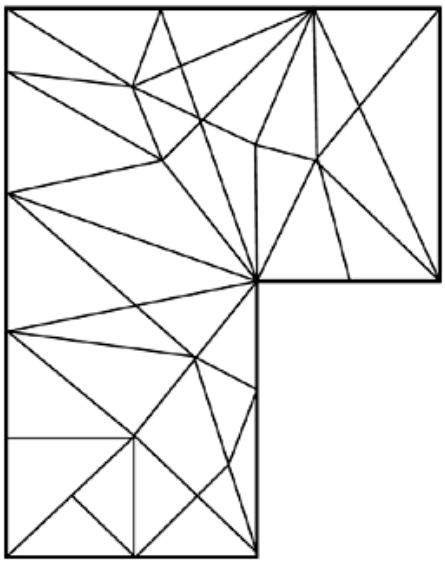


Fig. 3. Typical finite-element meshing of an L-shaped domain using triangular elements.

expressions into Maxwell's equations, we obtain

$$\sum_{j=1}^{N} \psi_j(\boldsymbol{r}) \frac{d\boldsymbol{H}_j(t)}{dt} = -\frac{1}{\mu} \sum_{i=1}^{M} \nabla\phi_i(\boldsymbol{r}) \times \boldsymbol{E}_i(t) \tag{24a}$$

$$\sum_{i=1}^{M} \phi_i(\boldsymbol{r}) \frac{d\boldsymbol{E}_i(t)}{dt} = \frac{1}{\epsilon} \sum_{j=1}^{N} \nabla\psi_j(\boldsymbol{r}) \times \boldsymbol{H}_j(t) \tag{24b}$$

Because both $\phi_i(\boldsymbol{r})$ and $\psi_j(\boldsymbol{r})$ are known functions, the only unknowns in Eqs. (24a) are the time-dependent nodal values of the electric and magnetic fields $\boldsymbol{E}_i(t)$ and $\boldsymbol{H}_j(t)$ at the nodal points $\boldsymbol{r}_i$. By conveniently normalizing the basis functions so that $\phi_i(\boldsymbol{r}_j) = \psi_i(\boldsymbol{r}_j) = \delta_{ij}$, and enforcing Eqs. (24a) at each nodal point (point matching), we obtain

$$\frac{d\boldsymbol{H}_j(t)}{dt} = -\frac{1}{\mu} \sum_{i=1}^{M} \nabla\phi_i(\boldsymbol{r}_j) \times \boldsymbol{E}_i(t) \tag{25a}$$

$$\frac{d\boldsymbol{E}_i(t)}{dt} = \frac{1}{\epsilon} \sum_{j=1}^{N} \nabla\psi_j(\boldsymbol{r}_i) \times \boldsymbol{H}_j(t) \tag{25b}$$

This corresponds to the semidiscrete system of the finite-element method (i.e., discretization in space only). Because both $\psi_j(\boldsymbol{r})$ and $\psi_j(\boldsymbol{r})$ have finite support in space, only a few terms in the summation in Eq. (25a)

will contribute, and, because of the point matching, an explicit scheme for the time update is obtained. The leapfrog scheme can then be used similarly to the FDTD method. In this manner we have

$$\boldsymbol{H}_j^{n+\frac{1}{2}} = \boldsymbol{H}_j^{n-\frac{1}{2}} - \frac{\Delta t}{\mu} \sum_{i=1}^{M} \nabla \phi_i(\boldsymbol{r}_j) \times \boldsymbol{E}_i^n \tag{26a}$$

$$\boldsymbol{E}_i^{n+1} = \boldsymbol{E}_i^n + \frac{\Delta t}{\epsilon} \sum_{j=1}^{N} \nabla \psi_j(\boldsymbol{r}_i) \times \boldsymbol{H}_j^{n+\frac{1}{2}} \tag{26b}$$

Other strategies are possible to obtain the expansion coefficients in Eq. (23a). In general, instead of a point matching, testing functions (spanning the *test* space) are utilized in conjunction with the basis functions (spanning the *trial* space), and the resultant semidiscrete system produces a implicit scheme. These are known as *Galerkin* methods and are perhaps the most popular. Point matching can be shown to be equivalent to a special case of a Galerkin method with Dirac delta test functions (distributions) associated with the electric and magnetic field nodes. Another strategy is the *least-squares* method of weighted residuals, where the square of residuals is integrated over the domain of the problem and the expansion coefficients in Eq. (23a) are obtained by minimizing the resulting integral. Note that in contrast to the FDTD scheme described previously, all three components of the electric field are placed at the same node for Eqs. (26a). The same is true for the magnetic field components.

There are many important issues connected with the finite-element method that are beyond the scope of this article, such as variational formulations, choice of meshing and elements utilized, bases and test functions, nodal versus edge elements, and mass lumping (a procedure used to convert an implicit time-update scheme to an explicit one). For a detailed discussion of those issues, the reader is referred to Refs. 17,18.

Finite-Volume Time-Domain Methods. The term *finite-volume* has a loose meaning in the literature, and it often can refer to different discretization methods. For instance, it sometimes synonymous with three-dimensional FDTD schemes on irregular, unstructured grids. Here we will use the classification employed in Ref. 19. The finite-volume time-domain method starts by subdividing the computational domain into elementary volumes (most often irregular). Upon integration of Maxwell's curl equations in each elementary volume, we obtain

$$-\mu \int_V \frac{\partial \boldsymbol{H}}{\partial t} dv = \int_V \nabla \times \boldsymbol{E} dv = \int_{\partial V} \hat{\boldsymbol{n}} \times \boldsymbol{E} ds \tag{27a}$$

$$\epsilon \int_V \frac{\partial \boldsymbol{E}}{\partial t} dv = \int_V \nabla \times \boldsymbol{H} dv = \int_{\partial V} \hat{\boldsymbol{n}} \times \boldsymbol{H} ds \tag{27b}$$

Many time-domain algorithms can be obtained through different choices of elementary volumes and surfaces. As with the finite-difference and finite-element methods, the discretization scale should resolve the wavelength well to minimize dispersion error. The grid may consist of cubes, distorted cubes, tetrahedrons, prisms, or their combination. By using a leapfrog scheme similar to the previously discussed finite-difference

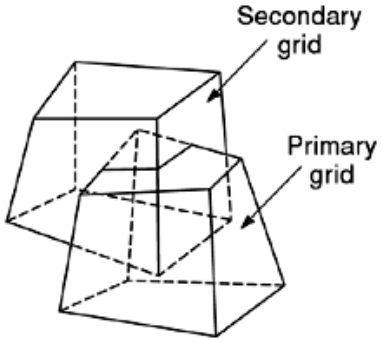


Fig. 4. Illustration of the dual grid construction used in the finite-volume method with cubical elements.

and finite-element discretizations, the time update equations for the above become

$$-\frac{\mu}{\Delta t}\int_V \left(\boldsymbol{H}^{n+\frac{1}{2}} - \boldsymbol{H}^{n-\frac{1}{2}}\right) dv = \int_{\partial V} \hat{\boldsymbol{n}} \times \boldsymbol{E}\, ds \tag{28a}$$

$$\frac{\epsilon}{\Delta t}\int_V \left(\boldsymbol{E}^{n+1} - \boldsymbol{E}^{n}\right) dv = \int_{\partial V} \hat{\boldsymbol{n}} \times \boldsymbol{H}\, ds \tag{28b}$$

Similarly to Yee's FDTD case, a dual grid is often utilized as illustrated in Fig. 4, where the electric field components are located on the primary grid and the magnetic field components are located on the secondary grid. At each time step a double interpolation should be used to interpolate field values from one grid to another. There is a close association of the finite-volume time-domain method with the FDTD method, in that it can be shown that the FDTD can be derived through application of the integral relations

$$-\mu \int_A \frac{\partial \boldsymbol{H}}{\partial t} \cdot \hat{\boldsymbol{n}} dS = \int_{\partial A} \boldsymbol{E} \cdot d\boldsymbol{l} \tag{29a}$$

$$\epsilon \int_A \frac{\partial \boldsymbol{E}}{\partial t} \cdot \hat{\boldsymbol{n}} dS = \int_{\partial A} \boldsymbol{H} \cdot d\boldsymbol{l} \tag{29b}$$

Therefore, while the finite-volume time-domain method is derived by using the curl of the electric field over a surface to find the magnetic vector at the center of the enclosed volume as in Eq. (27a) [and vice versa on the dual grid, Eq. (27b)], the FDTD can be derived through using the circulation of the electric field around a contour to update the magnetic field over the enclosed area as in Eq. (29a) [and vice versa on the dual-grid, Eq. (29b)]. For an additional discussion of the finite-volume time-domain method, the reader may consult Ref. 19.

Time Integration Schemes

In the previous sections, we have illustrated the main features of different discretization methods using leapfrog schemes. The leapfrog scheme is convenient for a system of two first-order equations such as Maxwell's curl equations, because it achieves overall second-order accuracy in time with only first-order time differencing. However, many other time-stepping schemes are possible. In this section, we will illustrate some of those.

Single-Step Methods. Starting from a problem already discretized in the spatial domain, the resulting semidiscrete problem can be written as

$$\frac{d\boldsymbol{v}(t)}{dt} = \boldsymbol{F}(\boldsymbol{v}, t) \tag{30}$$

where $\boldsymbol{v}$ and $\boldsymbol{F}$ are vectors, and $\boldsymbol{v}$ is subject to some initial condition, $\boldsymbol{v}(t_0) = \boldsymbol{v}_0$. In general $\boldsymbol{F}$ can be a nonlinear function. The algorithms for solving first-order equations such as Eq. (30) immediately generalize to systems of first-order equations, and because a higher-order equation can be cast as a system of first-order equations, the methods discussed here apply to differential equations of arbitrary order. In the case of Maxwell's equations, $\boldsymbol{F}$ is linear and Eq. (30) can be decomposed as

$$\frac{d\boldsymbol{v}_1(t)}{dt} = \boldsymbol{F}_1(\boldsymbol{v}_2) \tag{31a}$$

$$\frac{d\boldsymbol{v}_2(t)}{dt} = \boldsymbol{F}_2(\boldsymbol{v}_1) \tag{31b}$$

In this case, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ represent the field components at all grid points, while $\boldsymbol{F}_1(\boldsymbol{v}_2) = \boldsymbol{F}_1 \cdot \boldsymbol{v}_2$ and $\boldsymbol{F}_2(\boldsymbol{v}_1) = \boldsymbol{F}_2 \cdot \boldsymbol{v}_1$ are (sparse) matrices representing the curl operators on the grid.

A time-step update scheme for the prototypical Eq. (30) can be written as

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \Delta t \left[\theta \boldsymbol{F}(\boldsymbol{v}^n, t^n) + (1-\theta)\boldsymbol{F}(\boldsymbol{v}^{n+1}, t^{n+1})\right] \tag{32}$$

with $t^n = n\ \Delta t$, and $0 \le \theta \le 1$. This is called a *theta method*. If $\theta = 0$, we arrive at *Euler's method* (explicit), and if $\theta = \frac{1}{2}$, we arrive at the *trapezoidal rule.*

We can interpret Eq. (32) geometrically, by assuming the slope of the solution to be piecewise constant and given by a linear combination, weighted through the variable θ, of the derivatives at the endpoint of each discretization interval. Alternatively, we can also use a Taylor expansion as in Eqs. (16a) to arrive at the above approximation. By doing so, it can be shown that the theta approximation is of first order, except for $\theta = 1/2$, when it is of second order (trapezoidal rule). Moreover, if $\theta = 0$ the method is explicit, and otherwise implicit.

Multistep Methods. The theta method and the leapfrog scheme described before are examples of *step-by-step* methods, that is, once the solution for $\boldsymbol{v}^{n+1}$ is obtained, the value of $\boldsymbol{v}^n$ is discarded for future updates. However, past values at more than one time step can be used to compute the value at next time step. This gives rise to *multistep* methods, which can be written in general form as

$$\sum_{m=0}^{s} a_m \boldsymbol{v}_{n+m} = \Delta t \sum_{m=0}^{s} b_m \boldsymbol{F}(\boldsymbol{v}^{n+m}, t^{n+m}) \tag{33}$$

If $b_s = 0$, the method is explicit; otherwise, it is implicit. Of course, the efficacy of the method depends on the values of the coefficients a_m and b_m. The method above has $2s + 1$ degrees of freedom (coefficients), and in principle an optimal scheme could be obtained by adjusting the maximum number of coefficients. However, it can be shown that an implicit method of order $2s$ does not converge for $s \geq 3$. It is also possible to show that the maximum order for a convergent s-step method, as in Eq. (33), is equal to s for explicit schemes and to 2 int$[(s + 2)/2]$ for implicit schemes. This result is known as the *Dahlquist first barrier* (11).

Convergence is a most important characteristic of a time-stepping scheme. A scheme is said to be convergent if, for every equation of the form in Eq. (30), $|\boldsymbol{v}^n{}_{\Delta t} - \boldsymbol{u}(t^n)| \to 0$ as $\Delta t \to 0$ for all n, where $\boldsymbol{u}$ denotes the actual solution (i.e., exact) in the case of an ordinary-differential-equation problem or a semidiscrete solution in the case of a partial-differential-equation problem, and $\boldsymbol{v}^n{}_{\Delta t}$ denotes the numerical solution obtained by using a time step of size Δt. The numerical solution of a convergent scheme tends to the analytical or semidiscrete solution as the discretization step size approaches zero. Note that, because the semidiscrete solution is itself an approximation to the exact solution, a time-stepping convergent scheme may not converge to the exact solution of the problem if the spatial discretization is not consistent (13). In that case, to study the convergence properties of the overall numerical scheme, we must consider the discretization with respect to time and space conjointly.

A popular multistep scheme is the two-step *predictor–corrector* scheme. We again consider Eq. (30). In the first step (predictor), we predict the function value at the middle of the discretization interval:

$$\boldsymbol{v}^{n+\frac{1}{2}} = \boldsymbol{v}^n + \frac{\Delta t}{2}\boldsymbol{F}(\boldsymbol{v}^n, t^n) \tag{34}$$

and this is followed by a corrector step, where the next step value is estimated through

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \Delta t\, \boldsymbol{F}(\boldsymbol{v}^{n+\frac{1}{2}}, t^{n+\frac{1}{2}}) \tag{35}$$

The predictor–corrector scheme can be related to the popular *Lax–Wendroff* scheme when $\boldsymbol{F}$ is a linear time-invariant function. The Lax–Wendroff scheme uses the following identity from a Taylor expansion:

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \Delta t \left.\frac{d\boldsymbol{v}}{dt}\right|^n + \frac{\Delta t^2}{2}\left.\frac{d^2\boldsymbol{v}}{dt^2}\right|^n + O(\Delta t^3) \tag{36}$$

to estimate the derivative at t^n as

$$\left.\frac{d\boldsymbol{v}}{dt}\right|^n \approx \frac{1}{\Delta t}\left(\boldsymbol{v}^{n+1} - \boldsymbol{v}^n\right) - \frac{\Delta t}{2}\left.\frac{d^2\boldsymbol{v}}{dt^2}\right|^n \tag{37}$$

The second derivative of $\boldsymbol{v}$ in Eq. (37) can be obtained from Eq. (30):

$$\left.\frac{d^2\boldsymbol{v}}{dt^2}\right|^n = \left.\frac{d\boldsymbol{F}(\boldsymbol{v}, t)}{dt}\right|^n = \left.\frac{d\boldsymbol{F}(\boldsymbol{v}, t)}{d\boldsymbol{v}} \cdot \boldsymbol{F}(\boldsymbol{v}, t)\right|^n + \left.\frac{\partial \boldsymbol{F}(\boldsymbol{v}, t)}{\partial t}\right|^n \tag{38}$$

and, if $\boldsymbol{F}(\boldsymbol{v},t) = \boldsymbol{F} \cdot \boldsymbol{v}$ (linear, time-invariant), Eq. (36) then becomes

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \Delta t\, \boldsymbol{F} \cdot \boldsymbol{v}^n + \frac{\Delta t^2}{2} \boldsymbol{F}^2 \cdot \boldsymbol{v}^n \tag{39}$$

which is a second-order scheme.

Runge–Kutta (*RK*) methods form another popular class of time-stepping schemes. They are essentially based on (estimated) numerical quadrature rules to approximate the integral

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \int_{t^n}^{t^{n+1}} \boldsymbol{F}\left(\boldsymbol{v}(t'), t'\right) dt' \tag{40}$$

through a sum like

$$\boldsymbol{v}^{n+1} \approx \boldsymbol{v}^n + \Delta t \sum_{k=1}^{K} c_k \boldsymbol{F}\left(\boldsymbol{v}(t_k), t_k\right) \tag{41}$$

where $t_k \in (t^n, t^{n+1})$, $k = 1, \ldots, K$. Because the values of $\boldsymbol{F}\left(\boldsymbol{v}(t_k), t_k\right)$ are not known *a priori*, they also need to be approximated. The idea is to use an estimate $\boldsymbol{F}\left(\boldsymbol{v}\left(t_l\right), t_l\right) \approx \zeta_l$, given as a linear combination of the previous estimates ζ_k, $k = 0, \ldots, l-1$, namely,

$$\begin{aligned}
\zeta_1 &= \boldsymbol{v}^n \\
\zeta_2 &= \boldsymbol{v}^n + \Delta t\, A_{21} \boldsymbol{F}\left(\zeta_1, t^n\right) \\
&\vdots \\
\zeta_l &= \boldsymbol{v}^n + \Delta t \sum_{k=1}^{l-1} A_{lk} \boldsymbol{F}\left(\zeta_k, t^n + \alpha_k\, \Delta t\right) \\
\boldsymbol{v}^{n+1} &= \boldsymbol{v}^n + \Delta t \sum_{k=1}^{K} c_k \boldsymbol{F}\left(\zeta_k, t^n + \alpha_k\, \Delta t\right)
\end{aligned} \tag{42}$$

The matrix (A_{lk}) is called the RK matrix, c_k are the RK weights, and α_k are the RK nodes. The number K denotes the stage of the RK scheme (and usually, but not always, also corresponds to its order). Popular third-order schemes are the classical RK scheme, with $K = 3$ and $\alpha_1 = 0$, $\alpha_2 = \frac{1}{2}$, $\alpha_3 = 1$, $c_1 = \frac{1}{6}$, $c_2 = \frac{2}{3}$, $c_3 = \frac{1}{6}$, $A_{21} = \frac{1}{2}$, $A_{31} = -1$, $A_{32} = 2$, and $A_{kl} = 0$ for $k \leq l$; and the Nystrom RK scheme, also with $K = 3$, and $\alpha_1 = 0$, $\alpha_2 = \alpha_3 = \frac{2}{3}$, $c_1 = \frac{1}{4}$, $c_2 = c_3 = \frac{3}{8}$, $A_{21} = A_{32} = \frac{2}{3}$, and all other $A_{kl} = 0$.

The best-known fourth-order Runge–Kutta method has $K = 4$, and $\alpha_1 = 0$, $\alpha_2 = \alpha_3 = \frac{1}{2}$, $\alpha_4 = 1$, $c_1 = c_4 = \frac{1}{6}$, $c_2 = c_3 = \frac{1}{3}$, $A_{21} = A_{32} = \frac{1}{2}$, $A_{43} = 1$, and all other $A_{kl=0}$.

One of the reasons for the popularity of the fourth-order RK scheme is that, to obtain a fifth-order scheme, we need at least six stages. All these RK schemes correspond to *explicit* schemes, but *implicit* RK schemes can also be derived (11). Implicit schemes require the solution of a linear system at each time step, but present superior stability properties.

Convolution and Recursive Functions

In (time-)dispersive media, the relationship between the electric field vectors $\boldsymbol{D}$ and $\boldsymbol{E}$ in the time domain (constitutive relation) is given by a convolutional operator of the form

$$\boldsymbol{D}(t) = \int_0^t dt'\, \boldsymbol{E}(t-t')\epsilon(t') = \epsilon_\infty \epsilon_0 \boldsymbol{E}(t) + \epsilon_0 \int_0^t dt'\, \boldsymbol{E}(t-t')\chi(t') \tag{43}$$

Translated to the frequency domain, the above relation reads

$$\boldsymbol{D}(\omega) = \epsilon(\omega)\boldsymbol{E}(\omega) \tag{44}$$

where $\epsilon(\omega)$ is the (frequency-dependent) permittivity, ϵ_0 is the free-space permittivity, ϵ_∞ is the infinite-frequency permittivity (instantaneous response), and $\chi(t)$ is the time-domain susceptibility. In Eq. (43) causality was invoked by having $\epsilon(t) = 0$ for $t < 0$.

A naive implementation of the convolution in Eq. (43) for a time-stepping scheme requires the storage of the whole previous time history of $\boldsymbol{E}(t)$ in order to obtain $\boldsymbol{D}(t)$ at each successive time step. However, this can be avoided through the use of a *recursive convolution* algorithm (20,21), as described below.

Letting $t^n = n\ \Delta t$, we have

$$\boldsymbol{D}(t) \approx \boldsymbol{D}(t^n) = \boldsymbol{D}^n = \epsilon_\infty \epsilon_0 \boldsymbol{E}^n + \epsilon_0 \int_0^{n\,\Delta t} dt'\, \boldsymbol{E}(n\,\Delta t - t')\chi(t') \tag{45}$$

If the time interval Δt is sufficiently small, we may approximate the field quantities as constants over each interval (piecewise-constant approximation) so that the above integration becomes a summation of the form

$$\boldsymbol{D}^n = \epsilon_\infty \epsilon_0 \boldsymbol{E}^n + \epsilon_0 \sum_{m=0}^{n-1} \boldsymbol{E}^{n-m} \int_{m\,\Delta t}^{(m+1)\,\Delta t} dt'\ \chi(t') \tag{46}$$

The objective is to incorporate the summation above in a explicit time-domain update at minimal computational cost. A discretized form of the constitutive relation can be written as

$$\boldsymbol{D}^{n+1} - \boldsymbol{D}^n = \epsilon_\infty \epsilon_0 (\boldsymbol{E}^{n+1} - \boldsymbol{E}^n) + \epsilon_0 (\boldsymbol{P}^{n+1} - \boldsymbol{P}^n) \tag{47}$$

with

$$\boldsymbol{P}^n = \sum_{m=0}^{n-1} \boldsymbol{E}^{n-m} \int_{m\,\Delta t}^{(m+1)\,\Delta t} dt'\, \chi(t') = \sum_{m=0}^{n-1} \boldsymbol{E}^{n-m} \tilde{\chi}^m \tag{48}$$

where we define

$$\tilde{\chi}^m = \int_{m\,\Delta t}^{(m+1)\,\Delta t} dt'\ \chi(t') \tag{49}$$

We assume the susceptibility function $\chi(t)$ to be an exponential function in time. As we will see later, this assumption is not as restrictive as it may seem.

At first, it appears that the summation in Eq. (48) (convolution) requires the storage of all the past values of $\boldsymbol{E}^n$. In order to see how, on assuming an exponential dependence for $\chi(t')$, this storage is actually not necessary, we use Eq. (48) and rewrite the last term in Eq. (47) as

$$\boldsymbol{P}^{n+1} - \boldsymbol{P}^n = \boldsymbol{E}^{n+1}\tilde{\chi}^0 + \sum_{m=0}^{n-1} \boldsymbol{E}^{n-m}\left(\tilde{\chi}^{m+1} - \tilde{\chi}^m\right) \tag{50}$$

We also define

$$\Delta\tilde{\chi}^n = \left(\tilde{\chi}^{n+1} - \tilde{\chi}^n\right) \tag{51}$$

and the recursive accumulator

$$\boldsymbol{Q}^n = \sum_{m=0}^{n-1} \boldsymbol{E}^{n-m}\Delta\tilde{\chi}^n \tag{52}$$

For a susceptibility function of the form

$$\chi(t) = A_0 e^{-t/\tau_0} u(t) \tag{53}$$

for $t > 0$ and zero otherwise, Eq. (52) can then be written recursively as

$$\boldsymbol{Q}^n = \boldsymbol{E}^n\tilde{\chi}^0 + e^{-\Delta t/\tau_0}\boldsymbol{Q}^{n-1} \tag{54}$$

with $\boldsymbol{Q}^0 = \boldsymbol{Q}^1 = 0$. From the above, we observe that only the previous value of the recursive accumulator is needed.

The recursive convolution algorithm can be easily extended to the case where the susceptibility function $\chi(t)$ displays sinusoidal or damped sinusoidal behavior, by equating it to the real part of a complex exponential susceptibility function (22). Moreover, the restriction of χ to a exponential function above is not problematic, because the susceptibility function from a more arbitrary frequency-dependent material $\epsilon(\omega)$ can be modeled as a sum of exponentials, for example, via Prony's method.

Other variants of the recursive convolution algorithm exist. For instance, instead of approximating the electric field as a constant at each time interval in Eq. (46), one can approximate it as a linear function over each interval. This gives rise to the so-called *piecewise-linear recursive convolution* algorithm (23).

Moreover, it is also possible to efficiently incorporate the dispersive behavior of Eq. (43) into update equations through entirely different approaches, such as the z-transform method or the auxiliary differential equation (*ADE*) method. In the ADE approach, for instance, the frequency-dependent dispersion relation is transformed into a ordinary differential equation in the time variable, which relates $\boldsymbol{E}$ and $\boldsymbol{D}$. To illustrate the approach, we take $\epsilon(\omega)$ having the form (Debye dispersion)

$$\epsilon(\omega) = \frac{\alpha}{1 - i\omega\tau} \tag{55}$$

If we substitute this expression into Eq. (44), we arrive at

$$\left(1 - i\omega\tau\right) \boldsymbol{D} = \alpha \boldsymbol{E} \tag{56}$$

In the time domain, the above becomes

$$\left(1 + \tau \frac{\partial}{\partial t}\right) \boldsymbol{D} = \alpha \boldsymbol{E} \tag{57}$$

In a update scheme, the above equation can be easily updated concomitantly with Maxwell's curl equations. A more detailed description of the ADE and z-transform approaches is beyond the scope of this article, however. The interested reader may consult Refs. 24,25,26,27.

Time-Domain Integral-Equation Methods

Time-domain methods can be used not only in conjunction with differential equations, but also in conjunction with integral equations (time-domain integral equations). Models based on time-domain integral equations provide, in general, a more efficient formulation of, for example, surface-scattering phenomena or, in general, of problems where a Green's function is available (6,7,16,28).

A typical time-domain integral equation can be written as

$$g(\boldsymbol{r}, t) = \int_D d\boldsymbol{r}' \int dt' \; f(\boldsymbol{r}', t') K(\boldsymbol{r}, \boldsymbol{r}', t - t') \tag{58}$$

where $g(\boldsymbol{r},t)$ is the excitation function, $K(\boldsymbol{r},\boldsymbol{r}',t)$ is the kernel of the integral equation (often the Green's function of an associated differential equation), $f(\boldsymbol{r},t)$ is the unknown solution, for $\boldsymbol{r} \in D$, and D is some integration volume or surface. To solve Eq. (58) numerically, the unknown is represented in terms of suitably chosen spatiotemporal basis functions:

$$f(\boldsymbol{r}, t) = \sum_{n=1}^{N_s} \sum_{i=0}^{N_t} a_{n,i} b_n(\boldsymbol{r}) T_i(t) \tag{59}$$

where $a_{n,i}$ are the unknown coefficients. Spatial and the temporal basis functions having local support are often employed. Upon substituting Eq. (59) into Eq. (58) and taking the inner product (also called a *testing procedure*) with each of the spatial basis functions $b_m(\boldsymbol{r})$ at discrete times $t = t_j = j\,\Delta t$, the following matrix equation is obtained:

$$\bar{\boldsymbol{Z}}_0 \cdot \bar{\boldsymbol{A}}_j = \bar{\boldsymbol{G}}_j - \sum_{l=1}^{j-1} \bar{\boldsymbol{Z}}_l \cdot \bar{\boldsymbol{A}}_{j-l} \tag{60}$$

where the elements of the vectors $\bar{\boldsymbol{A}}_j$ and $\bar{\boldsymbol{G}}_j$ are given by

$$[\bar{\boldsymbol{A}}_i]_m = a_{m,i} \tag{61}$$

and

$$[\bar{\boldsymbol{G}}_i]_m = \int_D d\boldsymbol{r}\, b_m(\boldsymbol{r}) g(\boldsymbol{r}, t_i) \tag{62}$$

The elements of the (sparse) interaction matrix are given by

$$[\bar{\boldsymbol{Z}}_i]_{mn} = \int_D d\boldsymbol{r}\, b_m(\boldsymbol{r}) \int_D d\boldsymbol{r}'\; b_n(\boldsymbol{r}') \int dt'\, K(\boldsymbol{r}, \boldsymbol{r}', t - t') T_{j-i}(t')|_{t=t_j} \tag{63}$$

Equation (60) relates the expansion coefficients at the jth time step to the excitation and the expansion coefficients at previous time steps. This algorithm is often termed the marching-on-time (*MOT*) scheme (28). Numerical methods based on integral equations are often termed *boundary element methods* because, from the knowledge of the Green's function of the problem, the discretization needs to be performed only on the boundary of the domain. In electromagnetics, for historical reasons, the general procedure of projecting an integral equation into a matrix equation is often termed *method of moments* (29).

Current Issues

Time-domain algorithms continue to be a topic of active research. Because of the need to model large-scale broadband and nonlinear systems and with the ever increasing advance in the computational resources available, time-domain techniques have become an indispensable tool. Among the topics of current research in time-domain analysis are (a) higher-order methods to reduce the inherent dispersion error caused in finite-difference, finite-element, and finite-volume methods (30), (b) the use of irregular grids (structured or unstructured) to better model irregular geometries and to avoid the staircasing approximation of regular grids (31), (c) the use of (pseudo)spectral methods to reduce the number of nodes per wavelength necessary for a given accuracy (15), and (d) multigrid (32), adaptative meshing (33), or subgridding (34) schemes. In case of time-domain integral equations, the use of time-domain fast multipole methods (28) to reduce the computational complexity of the algorithms is also a topic under current investigation. Moreover, asymptotic techniques, such as the geometric theory of diffraction (*GTD*), originally developed in the frequency domain, have also been translated in recent years to the time domain (35). For a more detailed discussion of modern aspects of time-domain analysis, the reader is referred to Refs. 6,7,8 and references therein.

BIBLIOGRAPHY

1. A. V. Oppenheim,A. W. Willsky,I. T. Young, *Signals and Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1983.
2. A. V. OppenheimR. W. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.
3. E. Butkov, *Mathematical Physics*, New York: Addison-Wesley, 1968.
4. W. C. Chew, *Waves and Fields in Inhomogeneous Media*, Piscataway, NJ: IEEE Press, 1995.
5. J. A. Kong, *Electromagnetic Wave Theory*, Cambridge, MA: EMW Publishing, 1999.
6. E. K. Miller, Time domain modeling in electromagnetics, *J. Electromag. Waves Appl.*, **8** (9): 1125–1172, 1994.
7. S. M. Rao (ed.), *Time Domain Electromagnetics*, Academic Press, 1999.
8. A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Boston: Artech House, 1995.
9. K. S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. Antennas Propag.*, **14**: 302–307, 1966.

10. A. C. Cangellaris,C.-C. Li,K. K. Mei, "Point-matched time domain finite element methods for electromagnetic radiation and scattering," IEEE Trans. Antennas Propag., **35**: 1160–1173, 1987.
11. A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, Cambridge: Cambridge University Press, 1996.
12. C. Matiussi, An analysis of finite volume, finite element, and finite difference methods using concepts form algebraic topology, *J. Comput. Phys.*, **33** (2): 289–309, 1997.
13. F. L. TeixeiraW. C. Chew, Lattice electromagnetic theory from a topological viewpoint, *J. Math. Phys.*, **40** (1): 169–187, 1999.
14. B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge: Cambridge University Press, 1998.
15. Q. H. Liu, Large scale simulations of electromagnetic and acoustic measurements using the pseudospectral time-domain (PSTD), *IEEE Trans. Geosci. Remote Sens.*, **37**: 917–926, 1999.
16. M. Sadiku, *Numerical Techniques in Electromagnetics*, Boca Raton, FL: CRC Press, 1992.
17. J. M. Jin, *The Finite Element Method in Electromagnetics*, New York: Wiley, 1993.
18. P. P. SilvesterR. L. Ferrari, *Finite Elements for Electrical Engineers*, 3rd ed., Cambridge: Cambridge University Press, 1996.
19. K. S. YeeJ. S. Chen, The finite-difference (FDTD) and the finite-volume time-domain (FVTD) methods in solving Maxwell's equations, *IEEE Trans. Antennas Propag.*, **45**: 354–363, 1997.
20. R. J. Luebbers, *et al.* A frequency-dependent finite-difference time-domain formulation for dispersive materials, *IEEE Trans. Electromagn. Compat.*, **32**: 222–227, 1990.
21. R. SiushansianJ. LoVetri, Efficient evaluation of convolution integrals arising in FDTD formulations of electromagnetic dispersive media, *J. Electromagn. Waves Appl.*, **11**: 101–117, 1997.
22. R. J. LuebbersF. Hunsberger, FDTD for *n*-th order dispersive media, *IEEE Trans. Antennas Propag.*, **40**: 1297–1301, 1992.
23. D. F. KelleyR. J. Luebbers, Piecewise linear recursive convolution for dispersive media using FDTD, *IEEE Trans. Antennas Propag.*, **44**: 792–797, 1996.
24. D. M. Sullivan, *Z*-transform theory and the FDTD method, *IEEE Trans. Antennas Propag.*, **44**: 28–34, 1996.
25. D. M. Sullivan, Frequency-dependent FDTD methods using *Z* transforms, *IEEE Trans. Antennas Propag.*, **40**: 1223–1230, 1992.
26. T. KashiwaI. Fukai, A treatment by the FDTD method of the dispersive characteristics associated with electronic polarization, *Microw. Opt. Technol. Lett.*, **3** (6): 203–205, 1990.
27. W. H. WeedonC. M. Rappaport, A general method for FDTD modeling of wave propagation in arbitrary frequency dispersive media, *IEEE Trans. Antennas Propag.*, **45**: 401–410, 1997.
28. A. A. Ergin,B. Shanker,E. Michielssen, "Fast evaluation of three-dimensional transient wave fields using diagonal translation operators," *J. Comput. Phys.*, **146** (1): 157–180, 1998.
29. R. F. Harrington, *Field Computations by Moment Methods*, Piscataway, NJ: IEEE Press, 1993.
30. R. D. Gralia,D. R. Wilton,A. F. Peterson, Higher-order interpolatory vector bases for computational electromagnetics, *IEEE Trans. Antennas Propag.*, **45**: 329–342, 1997.
31. E. A. Navarro, *et al.* Some considerations about the finite difference time domain method in general curvilinear coordinates, *IEEE Microw. Guided Wave Lett.*, **4**: 396–398, 1994.
32. M. J. WhiteM. F. Iskander, Development of a multigrid FDTD code for three-dimensional applications, *IEEE Trans. Antennas Propag.*, **45**: 1512–1517, 1997.
33. I. S. KimW. J. R. Hoefer, A local mesh refinement algorithm for the time domain finite difference method using Maxwell's curl equations, *IEEE Trans. Microw. Theory Tech.*, **38**: 812–815, 1990.
34. P. Monk, Sub-gridding FDTD schemes, *Appl. Comput. Electromagn. Soc. J.*, **11**: 37–46, 1996.
35. T. W. Veruttipong,Time domain version of the uniform GTD, IEEE Trans. Antennas Propag., **38**: 1757–1764, 1990.

F. L. TEIXEIRA
Massachusetts Institute of Technology

/ /

Transfer Functions

Standard Article

Duane Mattern[1]

[1]Mattern Engineering, Controls and Hardware, Columbus, OH

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2461

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (201K)

Abstract

The sections in this article are

Continuous Time Transfer Functions

Transfer Function Models of First- and Second-Order Linear Systems

Cascading Transfer Functions and the Loading Assumption

Block Diagrams

Properties of Transfer Functions

State-Space Methods

Simulation of Linear Dynamic Systems

Control System Design and Analysis

Experimental Identification of Discrete Transfer Functions

| | |

Copyright © 1999-2008 All Rights Reserved.

TRANSFER FUNCTIONS

This article considers continuous time systems based on differential equations and discrete time systems based on difference equations. In the following sections, we will look at continuous time transfer function models for the resistor, inductor, and capacitor elements using the Laplace transform. We will use these models to construct first-order and second-order transfer function models and will discuss the construction of complex transfer functions from these subsystems. We will examine the properties of transfer functions and introduce special transfer functions like sensitivity and complementary sensitivity. We will also show how the single-input, single-output (SISO) transfer function can be extended to the multiinput, multioutput (MIMO) transfer function matrix. Also, we will take an abbreviated look at state-space representations of these systems. Finally, we will examine discrete time transfer functions and the Z transform and we will introduce model identification and parameter estimation.

A transfer function is a method for representing a dynamic mathematical model of a system. It is an algebraic expression that models the outputs of a system as a function of the system inputs. The input/output system is defined by the user of the model. Typically, a transfer function is used to model a *physical* system, which is something that can be described by the laws of physics. For example, consider the physical system defined by the resistor shown in Fig. 1(a). We define the system input to be the voltage, V, across the resistor, and we define the system output to be the electrical current, I, through the resistor. For systems with only one input and one output we can express the transfer function model as the ratio of the output divided by the input, which is the slope of the line shown in Fig. 1(b).

$$\frac{I}{V} = \frac{1}{R} \tag{1}$$

By definition, R is the resistance of the resistor. Transfer functions have a number of assumptions associated with them. For example, we have assumed that the resistance is constant in Eq. (1). As current passes through the resistor, it could cause self-heating from power dissipation (I^2R). This would cause a change in the temperature and the resistance of the resistor, violating the constant resistance assumption and possibly causing a modeling error. Thus the assumptions are an essential part of the model.

The model of the resistor described by Eq. (1) is a static, nondynamic transfer function that is a trivial example. A transfer function typically represents the dynamic characteristics of a system by parameterizing the transfer function with an operator that is indicative of the dynamics. The operator is usually the Laplace variable s, which results from con-

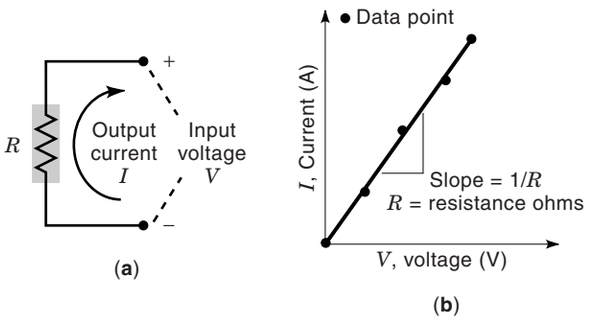
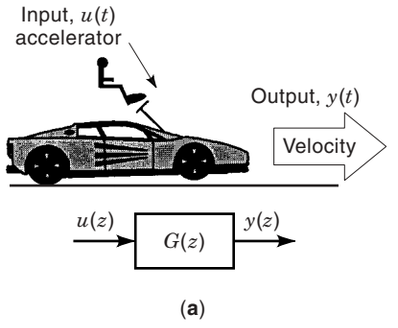
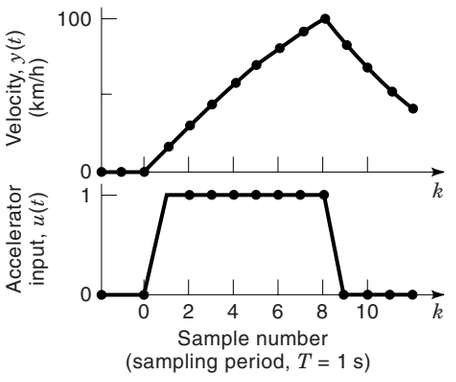


Figure 1. (a) Electrical resistor model showing components of a simple, static, transfer function with input voltage and output current. (b) Nondynamic, linear relationship between input voltage and output current in resistor model.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.



(a)



(b)

Figure 2. (a) Model of an automotive system with accelerator input and velocity output. (b) Dynamic velocity response of the model of the automotive system to a step input in the accelerator.

verting a differential equation to an algebraic equation using the Laplace transform, or the operator is the z variable, which results from converting a difference equation to an algebraic equation using the Z transform. We will use the automotive system sketched in Fig. 2(a) to introduce the dynamic transfer function. We can model the automobile with the accelerator as the system input and the velocity as the system output. We can then use the mathematical model to calculate the velocity as a function of the accelerator value and time. We can use this response to calculate how long it will take the vehicle to accelerate to some velocity from zero velocity after the driver provides an accelerator input, thus predicting the automobile acceleration performance. We will use a difference equation for this model with a fixed sampling timestep, T, equal to one second. The difference equation calculates the velocity at time = kT based on the accelerator position at time = kT and the previous value of the velocity at time = $(k - 1)T$ and is shown in Eq. (2):

$$y(k) = 0.93 \cdot y(k-1) + 16 \cdot u(k) \tag{2}$$

where u is the accelerator input ranging from zero to one (0 to 100%); y is the automobile velocity in kilometers per hour; k is the integer index where time = kT; and $y(k - 1)$ represents the velocity at the previous timestep. Table 1 shows the values for Eq. (1), starting at time equal to zero ($k = 0$) and the initial velocity equal to zero. The data in this table are plotted in Fig. 2(b). The top plot in Fig. 2(b) shows the exponential velocity response, and the bottom figure shows the step input to the accelerator. If we tune and validate the model in Eq. (2) against data taken from a real automobile, then we can use this model to represent the velocity response of that automobile. We could use this model in the design of a cruise control system for the automobile. The difference equation in Eq. (2) can be converted to an algebraic equation using the Z transform. A detailed discussion of the Z transform is outside the scope of this article, but it just requires a simple modification to Eq. (2) in this example. The transformation results in Eq. (3), which is in the z domain.

$$y(z) = 0.93 \cdot z^{-1}y(z) + 16 \cdot u(z) \tag{3}$$

where $u(z)$ is the Z transform of $u(k)$; $y(z)$ is the Z transform of $y(k)$; and $z^{-1}y(z)$ is the Z transform of $y(k - 1)$, where z^{-1} is the unit time delay operator. Note that Eq. (3) is an algebraic equation parameterized by z. The only variables in Eq. (3) are the input, $u(z)$, and the output, $y(z)$, so Eq. (3) can be manipulated to obtain the output over input ratio:

$$\frac{y(z)}{u(z)} = G(z) = \frac{16}{1 - 0.93 \cdot z^{-1}} = \frac{16z}{z - 0.93} \tag{4}$$

Table 1. Automotive Discrete Model Velocity Response as a Function of Time to a Step Input in the Accelerator

Sample Index, k and Time = kT	Accelerator Input $u(k)$ $(0 - 1)$	Vehicle Velocity $y(k)$ km/hour	One Step Delayed Vehicle Velocity $y(k - 1)$ km/hour
0	0	0	0
1	1	16	0
2	1	30.9	16
3	1	44.7	30.9
4	1	57.6	44.7
5	1	69.6	57.6
6	1	80.7	69.6
7	1	91.0	80.7
8	1	100.7	91.0
9	0	93.6	100.7
10	0	87.0	93.6
11	0	81.0	87.0
12	0	75.3	81.0

and $G(z)$ defines the discrete time, SISO dynamic transfer function model of the automotive system with a fixed sampling period, $T = 1$ second. Once we have validated the model by comparing it to the response of the real automobile, then we can use the model in place of the real system to perform the desired numerical studies, subject to the model assumptions. (An example of an assumption for a specific make and model of automobile might be the number of passengers and weight of the cargo that the vehicle was carrying.)

Note that $G(z)$ in Eq. (4) is comprised of a numerator and a denominator polynomial in the operator variable, z. The roots of the numerator polynomial are called system zeros, and the roots of the denominator polynomial are called system poles. For Eq. (4), there is one zero, $z = 0$, and one pole, $z = 0.93$. The pole and zero locations convey characteristics about the transfer function model.

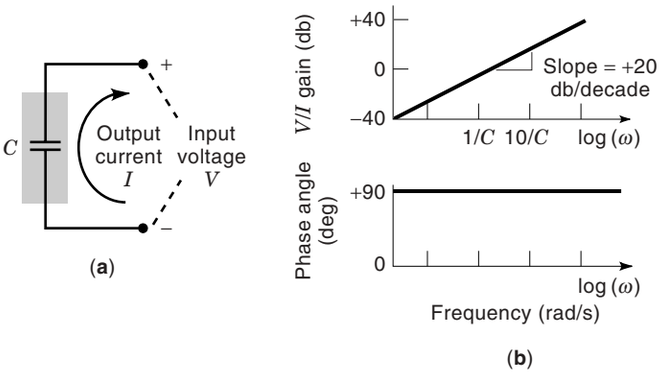


Figure 4. (a) Dynamic model of an electrical Capacitor with input voltage and output current. (b) Bode diagram frequency response of an electrical capacitor showing 20 db/decade slope of gain and 90 degree phase lead.

CONTINUOUS TIME TRANSFER FUNCTIONS

Three basic electrical components are the resistor, inductor, and capacitor. We have already seen the static transfer function model for the resistor in Eq. (1). We now consider continuous time dynamic transfer functions models for the inductor and capacitor using differential equations and the Laplace transform.

Continuous Time Transfer Function Model of an Inductor

Consider the integral equation of the system defined by the inductor shown in Fig. 3(a). The current through the inductor is a function of the initial current, $I(0)$, L, the inductance, and the time integral of the voltage across the inductor:

$$I(t) = \frac{1}{L}\int V(t)\,dt + I(0) \tag{5}$$

This equation is a dynamic model of the inductor. By using the Laplace transform, the operation of integration with respect to time can be replaced by the operation of division by the Laplace variable, s. Assuming that the initial current is zero, $I(0) = 0$, we can compute the output (current) over input (voltage) ratio resulting in the *sinusoidal* transfer function:

$$\frac{I(s)}{V(s)} = G_{\mathrm{L}}(s) = \frac{1}{Ls} \tag{6}$$

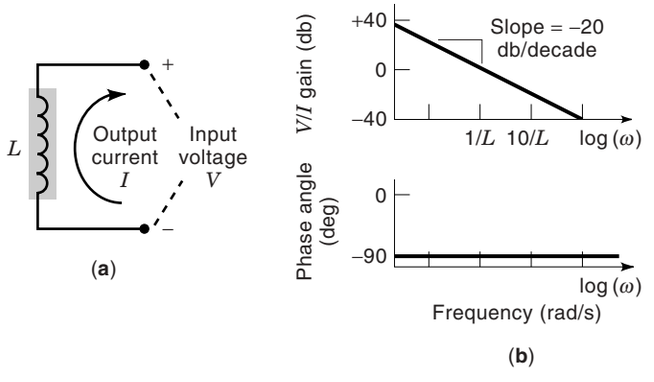


Figure 3. (a) Dynamic model of an electrical inductor with voltage input and current output. (b) Bode diagram frequency response of an electrical inductor showing −20 db/decade slope of gain and −90 degree phase lag.

Division by s, $1/s$, is indicative of an integration with respect to time. Equation (6) is a sinusoidal transfer function because the properties of the Laplace transform allow us to replace s with the complex term $s = j\omega$ to calculate the frequency response of the model. The term ω is the frequency in radians per second of a sinusoidal input to the system, and j is the complex variable used to represent $\sqrt{(-1)}$. Thus $G_{\mathrm{L}}(s = j\omega)$ is a complex function composed of real and imaginary terms. Note that $I(s)$ is the Laplace transform of $I(t)$. Similarly, $V(s)$ is the Laplace transform of $V(t)$. $G_{\mathrm{L}}(s = j\omega)$ is a complex function of frequency that we can represent with phasor notation. The phasor notation transforms the real and imaginary terms of a complex number to an amplitude (gain) and phase angle. At a specific frequency, ω_o, the complex value $G_{\mathrm{L}}(j\omega_o)$ can be converted to an amplitude ratio (gain from input to output) and a phase angle between the input and the output. The frequency response of the dynamic transfer function model of the inductor is shown in Fig. 3(b) using a Bode diagram. The amplitude ratio has a −20 db/decade slope because of the pure integration in the model. Conversion of the amplitude to decibels is obtained by taking the log-based 10 of the gain and multiplying by 20 (20 $\log_{10}$(gain). A decade is an order of magnitude change in frequency from ω to 10ω. The negative phase angle shown in Fig. 3(b) implies that the current output lags the input voltage, so if the input voltage were $V(t) = \cos(\omega t)$, then the output current would be $I(t) = \cos(\omega t - 90)/(L\omega)$.

Continuous Time Transfer Function Model of a Capacitor

We can also use the Laplace transform to obtain a dynamic transfer function model of a capacitor, as shown in Fig. 4(a). The current passing through a capacitor is the time derivative of the input voltage. The dynamic model of the capacitor is shown as a linear ordinary differential equation:

$$I(t) = C\frac{dV(t)}{dt} \tag{7}$$

Using the Laplace transform, Eq. (7) can be converted into an algebraic equation and then the output over the input ratio can be computed, resulting in the following:

$$\frac{I(s)}{V(s)} = G_c(s) = Cs \tag{8}$$

where s implies a time derivative operator. The resulting frequency response of this dynamic model of the capacitor is plotted in Fig. 4(b) using a Bode diagram to plot the input-output phasor information as an amplitude ratio (gain) and a phase angle between the input and the output. The positive phase angle implies that the current output leads the input voltage, so if the input voltage were $V(t) = \cos(\omega t)$, then the output current would be $I(t) = C\omega \cos(\omega t + 90)$.

Continuous Time Transfer Function Model of a General System

The transfer functions of individual components can be used to model interconnected devices. This offers a convenient way to construct models out of tested subsystems. The resulting single-input, single-output model may have many terms in the numerator and denominator polynomials, as shown in the general transfer function in Eq. (9):

$$\frac{y(s)}{u(s)} = G(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_2 s^2 + a_1 s + a_0} \tag{9}$$

where m is the highest order of the numerator polynomial and n is the highest order of the denominator polynomial. Causal models of physical systems require $n \geq m$, where causality means that only present inputs and past information are required to calculate the model output. Once there is a model of the system in the form $G(s)$, it can be used to estimate $y(t)$ given $u(t)$. For example, if a functional relationship is known for $u(t)$ (a step input, for example), it can be converted to $u(s)$ using the Laplace transform or a table of Laplace transformations. Then $y(s)$ can be calculated using the product $y(s) = G(s)u(s)$. Finally, $y(t)$ can be calculated from $y(s)$ using the inverse Laplace transform or a partial fraction expansion and the Laplace transformation tables. The section on Laplace transforms will present more details on this type of calculation.

TRANSFER FUNCTION MODELS OF FIRST- AND SECOND-ORDER LINEAR SYSTEMS

In Eq. (9) the mth order numerator polynomial has m roots and the nth order denominator polynomial has n roots. These numerator and denominator polynomials can be factored into the product of first-order and second-order polynomials as shown for a general case:

$$\frac{y(s)}{u(s)} = G(s) = \frac{K(s-z_1)(s-z_2)\ldots(s^2 + 2\zeta_{n1}w_{nn1}s + w_{nn1}^2)}{(s-p_1)(s-p_2)\ldots(s^2 + 2\zeta_{d1}w_{nd1}s + w_{nd1}^2)} \tag{10}$$

The first-order polynomial contains only a single real root. The second-order polynomial can contain two real roots or a complex pair of roots. The response of complex, high-order transfer functions can be obtained by adding up the contributions of all the first- and second-order polynomials in the frequency domain. To gain an understanding of how first- and second-order polynomials affect the response of a transfer function model, we will look at first- and second-order transfer functions.

First-Order Transfer Functions

Consider the electrical circuit of a low-pass filter comprising a resistor and capacitor, as shown in Fig. 5(a). A transfer function can be written from the current, I, to the output voltage V_o, and from the input voltage, V_i, to the current, I, as shown in Eq. (8) and in the following equations:

$$\frac{I(s)}{V_i(s)} = G_a(s) = \frac{1}{\left(R + \dfrac{1}{Cs}\right)} = \frac{Cs}{RCs + 1} \tag{11}$$

$$\frac{V_o(s)}{I(s)} = G_b(s) = \frac{1}{Cs} \tag{12}$$

These equations assume that no current is required to obtain the measurement of V_o [$I_o = 0$ in Fig. 5(a)]. We can combine Eqs. (11) and (12) and eliminate $I(s)$. The result is only a function of $V_o(s)$ and $V_i(s)$, which can be rewritten as the transfer function from the input voltage V_i to the output voltage V_o. The dynamic transfer function model of the low-pass filter is then

$$\frac{V_o(s)}{V_i(s)} = G_{1st}(s) = \frac{I(s)}{V_i(s)}\frac{V_o(s)}{I(s)} = G_a(s)G_b(s) = \frac{1}{\tau s + 1}, \text{ where } \tau = RC \tag{13}$$

By combining $G_a(s)$ and $G_b(s)$, we have demonstrated the multiplicative property of transfer functions. The frequency

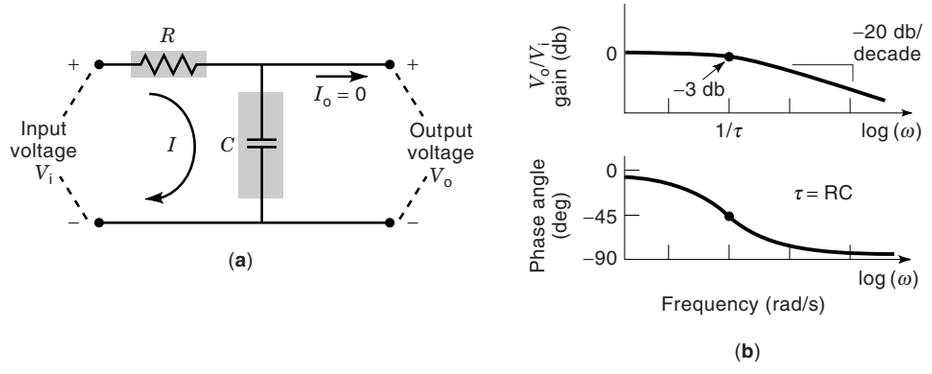


Figure 5. (a) Electrical first-order model of a passive low-pass filter comprised of a resistor and capacitor. (b) Bode diagram frequency response of a low-pass filter showing attenuation of gain at a frequencies greater than $1/\tau$. Gain is -3 dB and phase is -45 degrees at a frequency of $1/\tau$ rads/s.

response of this dynamic model is shown in Fig. 5(b) as a Bode diagram. The amplitude ratio is "flat" or approximately equal to 1 (0 dB) at low frequencies, and the amplitude decreases for frequencies greater than $1/\tau$. The gain equals -3 dB at a frequency $\omega = 1/\tau$. At higher frequencies the amplitude ratio decreases at the rate of -20 dB per decade (a decade is a range of frequency from ω to 10ω). This appears as a straight line on the plot of amplitude in decibels versus log frequency. Thus the system is called a low-pass filter because it allows low frequencies to pass but attenuates high frequencies. The phase angle starts at zero degrees, passes through -45 degrees at $\omega = 1/\tau$, and progresses to a phase angle of -90 degrees at high frequency. Note that the negative phase angle means that the output lags the input. The root of the polynomial in the denominator of this transfer function is $s = -1/\tau$. This value is the pole of the first-order transfer function, and it conveys information regarding the speed of response of the system. The time constant, τ is equal to the time it takes the system to respond to 63.2% of the final value when commanded with a step input.

Second-Order Transfer Functions

Figure 6(a) shows a second-order system comprised of a resistor, inductor, and a capacitor. Again we can write a transfer function from the input voltage to the current passing through all the components, shown in Eq. (14), and from the current to the output voltage, shown in Eq. (15).

$$\frac{I(s)}{V_i(s)} = G_1(s) = \frac{1}{\left(R + Ls + \dfrac{1}{Cs}\right)} = \frac{Cs}{LCs^2 + RCs + 1} \tag{14}$$

$$\frac{V_o(s)}{I(s)} = G_2(s) = \frac{1}{Cs} \tag{15}$$

By combining G_1(s) and G_2(s), eliminating the current, I, and rearranging the terms, we obtain the transfer function from the input voltage to the output voltage:

$$\frac{V_o(s)}{V_i(s)} = G_{2nd}(s) = G_1(s)G_2(s) = \frac{1}{\dfrac{s^2}{w_n^2} + \dfrac{2\zeta s}{w_n} + 1} \tag{16}$$

where $\omega_n = 1/\sqrt{LC}$ is the system natural frequency in radians per second and $\zeta = (R/2)\sqrt{C/L}$ is the system nondimensional damping coefficient. Note that Eq. (16) is a standard second-order transfer function. Depending on the value of ζ, the roots of the denominator polynomial can be real or both complex. Figure 6(b) shows the plot for $\zeta \approx 0.2$, which causes the amplitude of the frequency response to peak at a frequency near ω_n. There is an entire family of curves for the frequency response of a second-order system that vary with the value of ζ. When $\zeta > 1$, both of the roots of the denominator polynomial are real and the system is said to be overdamped. The amplitude response does not have a peak. When $\zeta = 1$, the system is said to be critically damped and the roots of the denominator polynomial are both real and repeated or identical. When $\zeta < 1$, the system is said to be underdamped. The two roots of the polynomial are a complex pair, $(a + jb)$ and $(a - jb)$, where a is the real part and b is the complex part of the root. As ζ approaches 0 from 1, the response becomes more oscillatory and the magnitude of the complex portion of the root increases. Figure 7 shows the second-order step response for four values of ζ when $\omega = 10$ rad/s. When $\zeta = 3$ the system is overdamped and responds slowly. When $\zeta = 1$ the system is critically damped and responds without an overshoot. When $\zeta = 0.4$ the system is underdamped and overshoots before settling in on the final value of 1.0 for a unit step response. When $\zeta = 0.1$ the system is oscillatory, and it takes several seconds for the oscillations to die out. When $\zeta = 0$ the system is said to be undamped and it will oscillate continuously because it is marginally stable (on the borderline between the mathematical definitions of stability and instability). We do not consider the case for $\zeta < 0$ because it implies a negative coefficient in the denominator polynomial, which indicates that the system is unstable from Routh stability criterion.

CASCADING TRANSFER FUNCTIONS AND THE LOADING ASSUMPTION

We have shown how transfer functions can be multiplied. But there is an assumption associated with this. When transfer functions are cascaded together, there can be energy trans-

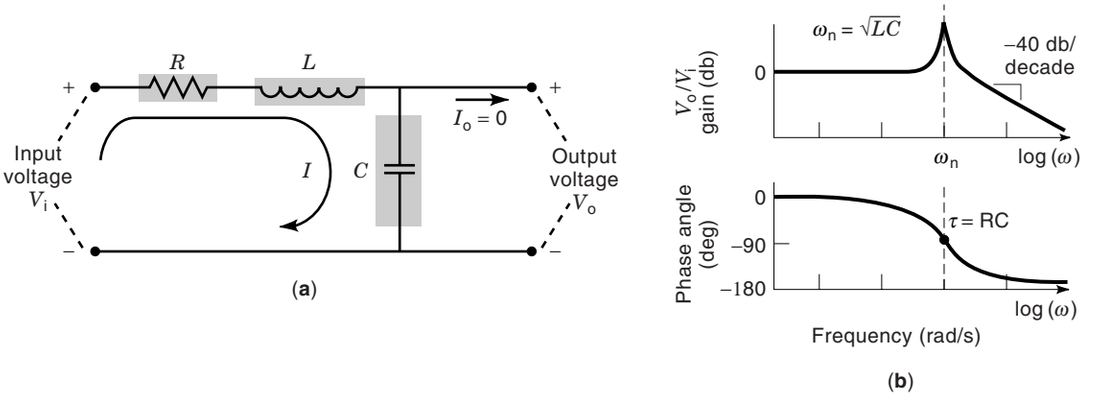


Figure 6. (a) Electrical second-order model comprised of inductor, resistor, and capacitor. (b) Bode diagram frequency response of a second-order system showing peak at natural frequency for system with low damping and high frequency attenuation of gain. The gain at ω_n depends on the damping coefficient, $\zeta = (R/2)\sqrt{C/L}$. The gain at ω_n is large when ζ is small (underdamped) and the gain is smaller when ζ is large (overdamped).

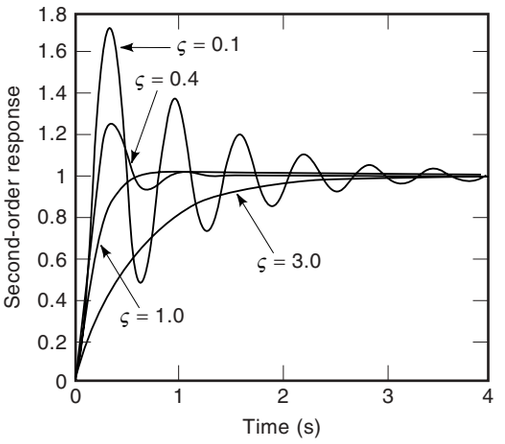


Figure 7. Time history response of second-order system with ω_n = 10 rads/s to a step input showing variation of the response with a range of damping coefficients, ζ. Note that the step response of the second-order system is sluggish for large values of ζ and oscillatory for small values of ζ.

ferred between the two systems represented by the transfer functions. By cascading transfer functions it is assumed that the energy extracted from one system does not significantly impact the response of that system. This energy flow is called loading. If there is significant loading from one system to the next, then multiplying the transfer functions violates an assumption and can lead to erroneous results, and the systems must be reanalyzed. Consider a system comprised of two cascaded low-pass first-order passive filters that were introduced in Fig. 5(a). If we multiply the two low-pass filter transfer functions from Eq. (13), we get

$$\frac{V_o(s)}{V_i(s)} = G_{\text{cascade}}(s) = G_1(s)G_2(s) = \frac{1}{\tau_1 s + 1} \cdot \frac{1}{\tau_2 s + 1} \tag{17}$$

where $\tau_1 = R_1C_1$ and $\tau_2 = R_2C_2$ are the time constants of the two cascaded filters. Equation (17) can be written as a second-order transfer function:

$$G_{\text{cascade}}(s) = \frac{1}{\tau_1\tau_2 s^2 + (\tau_1 + \tau_2)s + 1} \tag{18}$$

Note that by cascading two of the systems depicted in Fig. 5(a), we may have violated the assumptions that the output current draw is zero, ($I_{o1} = 0$). Reanalyzing the new system shown in Fig. 8 without this assumption results in the following transfer function for the two cascaded first-order filters:

$$G_{\text{cascade}}(s) = \frac{1}{\tau_1\tau_2 s^2 + (\tau_1 + R_1C_2 + \tau_2)s + 1} \tag{19}$$

The difference between Eq. (18) and Eq. (19) is the R_1C_2 term. So if R_1C_2 is small relative to τ_1 and τ_2, then cascading these two transfer functions is a good approximation. If these filters were active instead of passive filters, then the input impedance of the second filter would be high and the output impedance of the first filter would be low, so there would be little current drawn, and no loading effect.

BLOCK DIAGRAMS

Block diagrams and signal flow graphs are methods for visualizing systems constructed from subsystems, including transfer functions. We will concentrate on block diagrams because most of the computer-aided control system design software uses block diagrams for model construction. Block diagrams have a set of rules for manipulating the blocks in the diagram. These rules are identical to the rules of manipulating transfer functions. Figure 9(a) shows the product of two blocks representing the cascading of two transfer functions considered in the previous section in Eq. (17). The block diagram multiplication in Fig. 9(a) assumes a no-loading condition, and readers should be aware of this assumption when using the computer-aided simulation tools. Figure 9(b) shows the block diagram addition of two transfer functions.

Closed-Loop Block Diagrams

Figure 10 shows the concept of negative feedback in a closed-loop system using transfer functions $K(s)$ and $G(s)$. $K(s)$ represents a control system, and $G(s)$ represents a plant or controlled system. This figure represents the servomechanism control problem. There are a number of important transfer functions that will be examined in the next section using the closed-loop block diagram in Fig. 10.

PROPERTIES OF TRANSFER FUNCTIONS

Thus far in this article we have introduced the transfer function models for the resistor, inductor, and capacitor. It is important to note that these electrical elements have analogies in mechanical, thermal, and fluid systems. Transfer functions have applicability in a wide class of scientific and engineering fields. Equation (9) represents the Laplace transform of a constant coefficient, linear differential equation. Constant coefficient, linear differential equations adhere to the principle of superposition. Superposition states that the linear system input signal can be broken up into a sum of signals and the

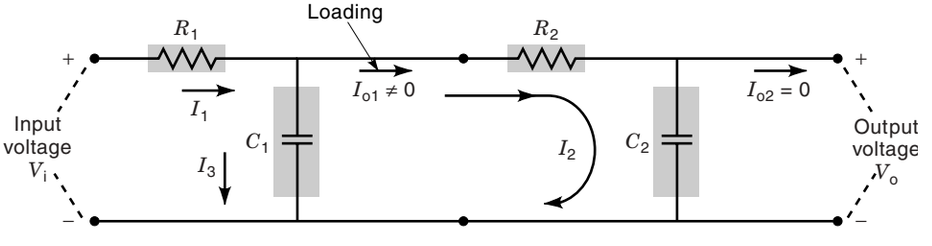


Figure 8. Cascaded first-order systems used to show the possible violation of a modeling assumption due to loading.

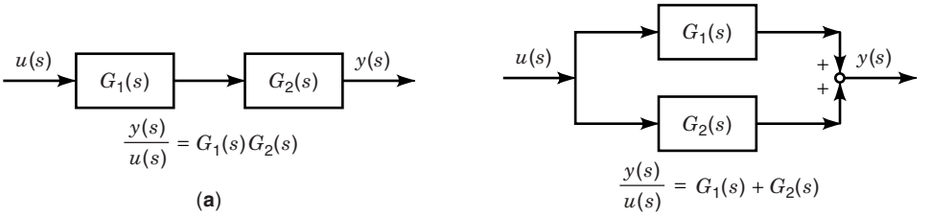


Figure 9. (a) Block diagram and transfer function multiplication. If $G_1(s)$ and $G_2(s)$ are transfer functions and not transfer function matrices, then the process of multiplication is commutative. (b) Block diagram and transfer function addition.

system output can be expressed as the sum of the system responses to each of the individual input signals. Consider the system $y(s) = G(s)u(s)$. We want to know what $y(t)$ is given by $u(t) = 6 + \cos(2\pi t) + \sin(3\pi t)$. We can define $u(t) = u_1(t) + u_2(t) + u_3(t) + u_4(t)$, where $u_1(t) = 3$, $u_2(t) = 3$, $u_3(t) = \cos(2\pi t)$, and $u_4(t) = \sin(3\pi t)$. Substituting for $u(t)$, we have

$$y(s) = G(s)u_1(s) + G(s)u_2(s) + G(s)u_3(s) + G(s)u_4(s) \quad (20)$$

where

$$y_1(s) = G(s)u_1(s), y_2(s) = G(s)u_2(s)$$
$$y_3(s) = G(s)u_3(s), y_4(s) = G(s)u_4(s) \quad (21)$$
$$y(s) = y_1(s) + y_2(s) + y_3(s) + y_4(s) \quad (22)$$

where

$$y_1(s) = G(s)u_1(s), y_2(s) = G(s)u_2(s)$$
$$y_3(s) = G(s)u_3(s), y_4(s) = G(s)u_4(s)$$

Using the inverse Laplace transform yields

$$y(t) = y_1(t) + y_2(t) + y_3(t) + y_4(t) \quad (23)$$

where $y_1(t)$, $y_2(t)$, $y_3(t)$, and $y_4(t)$, are the inverse Laplace transforms of $y_1(s)$, $y_2(s)$, $y_3(s)$, and $y_4(s)$, respectively. Thus the system output is equal to the sum of the system outputs corresponding to the individual portions of the input. Note that the numerical value of 6 was broken up into 3 and 3. This implies that scale factors pass through undisturbed, and Eq. (23) could be written as

$$y(t) = 2(y_1(t)) + y_2(t) + y_3(t) \quad (24)$$

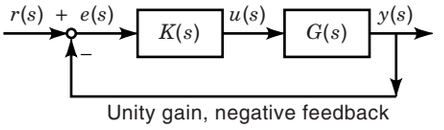


Figure 10. Block diagram of a unity gain, negative feedback system showing the control system transfer function, $K(s)$, and the controlled system transfer function, $G(s)$. The Laplace transform of the respective command signal, error, input, and output are $r(s)$, $e(s)$, $u(s)$, and $y(s)$. This block diagram is used to calculate the closed-loop, sensitivity, and complementary sensitivity transfer functions.

This is not true in general for nonlinear systems. The property of superposition holds for transfer functions because they are transformed from systems of linear, constant coefficient equations. Another property of transfer functions comes from the fact that the Laplace transform of a unit impulse is equal to one. Thus the unit impulse response of a system is just equal to the system transfer function since $y(s) = G(s)1 = G(s)$. Thus the inverse Laplace transform of the transfer function is the system time response to a unit impulse input.

Stability

We have not discussed the stability of the transfer functions because stability is covered elsewhere in this encyclopedia. We will just briefly mention that continuous time transfer functions in the Laplace domain are unstable if any denominator root, or transfer function pole, has a positive real portion. Thus, if the pole lies in the right-hand side of the y axis when plotted in the complex s plane, then the transfer function is unstable, as shown in Fig. 11(a). For discrete transfer functions parameterized with the Z transform variable, the transfer function is unstable if the complex pole lies outside the unit circle in the z plane, as shown in Fig. 11(b). (See Z TRANSFORMS for details.)

Sensitivity Transfer Function

Let us use transfer function algebra to solve for various transfer functions between variables. Consider the closed-loop block diagram shown in Fig. 10, where $G(s)$ is a transfer function of the system to be controlled (the plant) and $K(s)$ is a transfer function of the control system (the controller). The relationship from the commanded input, $r(s)$ to the controller error, $e(s)$, is

$$y(s) = G(s)u(s) = G(s)K(s)e(s) \quad (25)$$
$$e(s) = r(s) - y(s) \quad (26)$$

Substituting for $y(s)$ in Eq. (26) from Eq. (25) results in the following:

$$e(s) = r(s) - G(s)K(s)e(s) \quad (27)$$
$$e(s) = \frac{1}{1 + G(s)K(s)}\, r(s) \quad (28)$$
$$\frac{e(s)}{r(s)} = S(s) = \frac{1}{1 + G(s)K(s)} \quad (29)$$

$S(s)$ is called the sensitivity function, and it shows that as long as the product $G(s)K(s)$ is large relative to one, then the

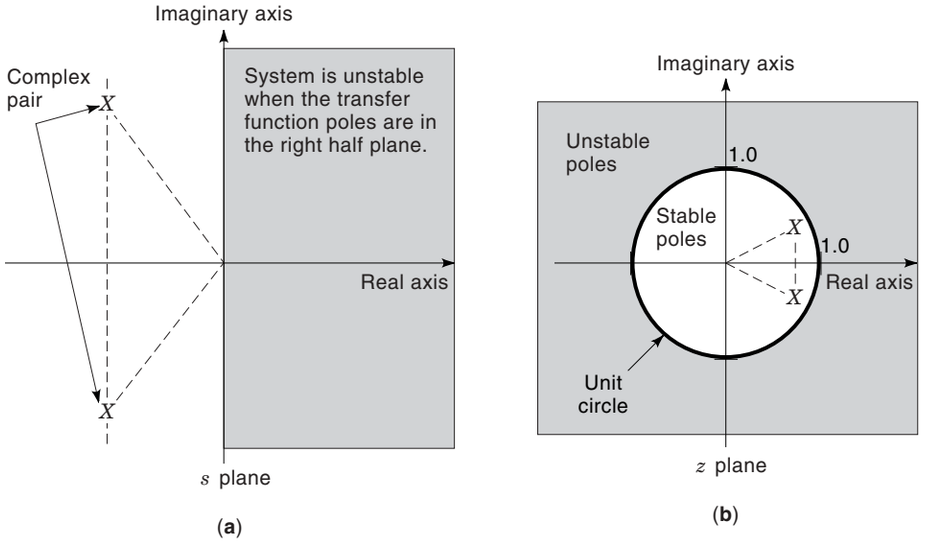


Figure 11. (a) An s domain plot showing the location of unstable poles for a continuous transfer function in the Laplace s variable. Two complex pair roots from a continuous second-order transfer function are shown. (b) A z domain plot showing the location of the unstable poles for a discrete transfer function in the z variable. Two complex pair roots from a discrete second-order transfer function are shown. (a) Stable poles of s domain transfer function are in the right half plane. (b) Stable poles of the z domain transfer function are within the unit circle.

error, $e(s)$, will be small. For large values of the product $G(s)K(s)$, Eq. (29) can be approximated as follows:

$$\text{when } G(s)K(s) \gg 1, \text{ then } S(s) \approx \frac{1}{G(s)K(s)} \tag{30}$$

The reason that Eq. (29) is called the sensitivity function will become apparent later in this discussion.

Complementary Sensitivity Transfer Function

In Fig. 10, consider the closed-loop transfer function relationship from the commanded input, $r(s)$, to the controlled output, $y(s)$. Using Eqs. (25) and (26) but substituting for $e(s)$ in Eq. (25) from Eq. (26) results in the following:

$$y(s) = G(s)K(s)r(s) - G(s)K(s)y(s) \tag{31}$$

$$y(s) = \frac{G(s)K(s)}{1 + G(s)K(s)}\, r(s) \tag{32}$$

$$\frac{y(s)}{r(s)} = T(s) = \frac{G(s)K(s)}{1 + G(s)K(s)} \tag{33}$$

$T(s)$ is called the complementary sensitivity transfer function. It is the transfer function from the commanded input, $r(s)$, to the controlled output, $y(s)$, and complements the sensitivity function because of the relationship between $T(s)$ and $S(s)$, shown as follows:

$$T(s) + S(s) = \frac{y(s)}{r(s)} + \frac{e(s)}{r(s)} = \frac{y(s) + (r(s) - y(s))}{r(s)} = 1 \tag{34}$$

So the sum of the sensitivity function and the complementary sensitivity function is equal to one.

The primary purpose of feedback is to reduce the sensitivity of the system to parameter variations and unwanted disturbances. Let us consider the block diagram in Fig. 10 without the feedback path. This would be an open-loop control system resulting in the following model:

$$y(s) = G(s)K(s)r(s) \tag{35}$$

where $G(s)$ represents a model of the system, which is just an approximation. If $G(s)$ is inaccurate, then the true system might be represented by

$$\begin{aligned} y(s) &= (G(s) + \Delta G(s))K(s)r(s) \\ &= G(s)K(s)r(s) + \Delta G(s)K(s)r(s) \end{aligned} \tag{36}$$

where $\Delta G(s)$ represents the modeling error. The resulting error in the output $y(s)$ is directly proportional to the modeling error. The closed-loop block diagram in Fig. 10 results in the transfer function in Eq. (33). If we introduce the plant uncertainty, $\Delta G(s)$, into Eq. (33) we have

$$\frac{y(s)}{r(s)} = \frac{(G(s) + \Delta G(s))K(s)}{1 + (G(s) + \Delta G(s))K(s)} = \frac{G(s)K(s) + \Delta G(s)K(s)}{1 + G(s)K(s) + \Delta G(s)K(s)} \tag{37}$$

Assuming that the model variation is $\Delta G(s)$ is small relative to $G(s)$, then Eq. (37) can be approximated as

$$\frac{y(s)}{r(s)} \approx \frac{G(s)K(s) + \Delta G(s)K(s)}{1 + G(s)K(s)} = T(s) + \frac{\Delta G(s)K(s)}{1 + G(s)K(s)} \tag{38}$$

So the change in the closed-loop, input/output transfer function, $T(s)$, due to the change in the open-loop transfer function, $G(s)$, is equal to $\Delta T(s)$, defined as follows:

$$\Delta T(s) = \frac{\Delta G(s)K(s)}{1 + G(s)K(s)} \tag{39}$$

Note that compared to the change in the open-loop equation in Eq. (36), the change in the closed-loop response is scaled by the denominator $(1 + G(s)K(s))$. Thus closing the loop with negative feedback reduces the affect of variations in the system plant. Also note that by the sensitivity function, $S(s)$ is essentially the reduction factor between Eqs. (36) and (39).

Control Sensitivity Transfer Function

Consider the relationship from the commanded input, $r(s)$, to the controller output or plant input, $u(s)$. Using Eqs. (25) and

(26) but substituting for $e(s)$ in Eq. (25) from Eq. (26), and using both pieces of Eq. (25), results in the following:

$$\begin{aligned} u(s) &= K(s)e(s) = K(s)(r(s) - y(s)) \\ &= K(s)r(s) - K(s)G(s)u(s) \end{aligned} \tag{40}$$

$$\frac{u(s)}{r(s)} = \frac{K(s)}{1 + K(s)G(s)} \tag{41}$$

Equation (41) is important in control system design because it gives the actuator response in a closed loop design. This allows the designer to take actuator rate and range limits into account by limiting the control sensitivity within the design procedure.

STATE-SPACE METHODS

The standard state-space representation is a set of four matrices, A, B, C, D, that make up a set of ordinary differential equations as follows:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{42}$$

where A is the system matrix, B is the input matrix, C is the output matrix, D is the feedforward matrix, x is a vector comprised of state variables, and $\dot{x}$ is the time derivative of the state vector. As before, u is the scalar input and y is the scalar output. State-space methods, state feedback, and state estimation are topics covered in other articles of this encyclopedia. We will discuss state-space representation briefly by saying that a transfer function can be converted to a state-space representation and vice versa. A state-space representation is not unique since the state of any representations can be transformed to another equivalent input-output representation and a new set of state variables. There are state-space representations and block diagrams that are standard representations of block diagrams. These standard forms represent specific state variable formulations. These forms are called the control, observer, and modal canonical forms, and they use only isolated integrators and gains as dynamic elements. The control and observer canonical forms are related to the concepts of observerability and controllability, which are discussed elsewhere in this encyclopedia. The following discussion holds for the transfer function of any single-input, single-output system. We will use a third-order system with a third-order numerator as an example, as follows:

$$\frac{y(s)}{u(s)} = G(s) = \frac{b_3 s^3 + b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \tag{43}$$

Control Canonical Form

The control canonical block diagram is shown in Fig. 12. The control canonical state-space representation is as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$
$$y = [b_0 - b_3 a_0 \quad b_1 - b_3 a_1 \quad b_2 - b_3 a_2] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [b_3] u \tag{44}$$

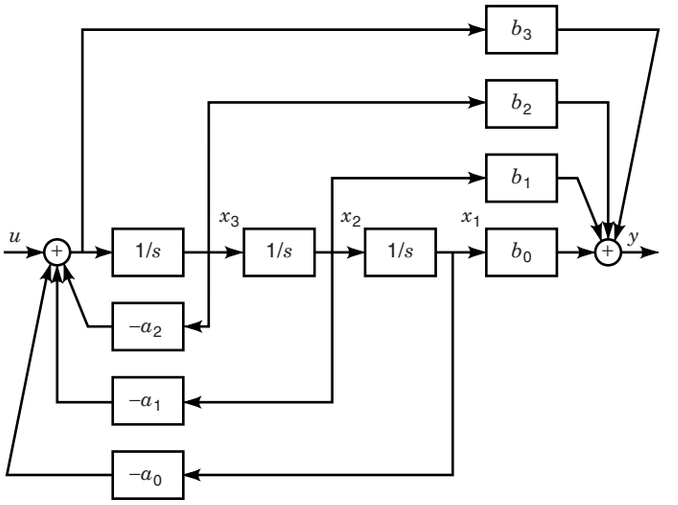


Figure 12. Block diagram showing the structure of the control canonical form.

Note that the block coefficients in Fig. 12 and the matrix scalar elements in Eq. (44) are the coefficients of the denominator in the transfer function in Eq. (43). Also, three new variables and the time derivatives of these variables were defined, x_1, x_2, x_3. These three variables make up the vector, $\boldsymbol{x}$, which is called the system state vector. This state variable is not unique, as we will see in a moment. The format in Eq. (44) and Fig. 12 is also called the phase variable form in some references.

Observer Canonical Form

The observer canonical block diagram is shown in Fig. 13. The observer canonical state space representation is as follows:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} -a_2 & 1 & 0 \\ -a_1 & 0 & 1 \\ -a_0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} b_2 - a_2 b_3 \\ b_1 - a_1 b_3 \\ b_0 - a_0 b_3 \end{bmatrix} u$$
$$y = [1 \quad 0 \quad 0] \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + [b_3] u \tag{45}$$

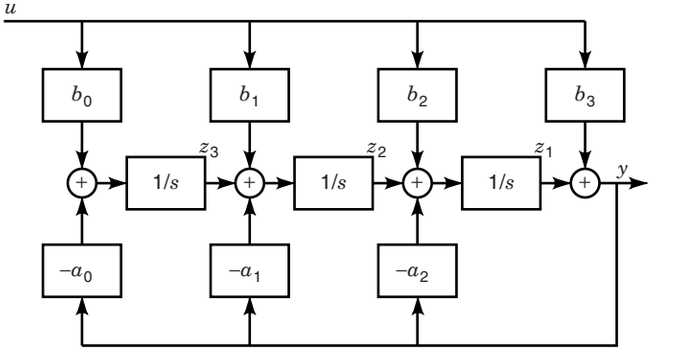


Figure 13. Block diagram showing the structure of the observer canonical form.

Note that the block coefficients in Fig. 13 and the matrix scalar elements in Eq. (45) are the coefficients of the denominator in the transfer function in Eq. (43). Also note that the state vector, $\boldsymbol{z}$, used in Eq. (45) is not the same state vector, $\boldsymbol{x}$, used in Eq. (44), even though the input, u, and output, y, variables are the same. The state vectors, $\boldsymbol{x}$ and $\boldsymbol{z}$, differ by a coordinate transformation. See the article in this encyclopedia on state space for more details. The form shown in Fig. 13 and in Eq. (45) is also called the rectangular form in some references because of the shape of the block diagram.

Modal Canonical Form

The block diagram for modal canonical form requires a discussion of residues and repeated roots and is outside of the scope of this article. We will just say that the modal canonical form results in a system matrix that is diagonal. The elements on the diagonal are made up of the roots of the denominator polynomial of the transfer function. For the modal canonical form, the elements on the diagonal of the A matrix could be complex, but there are methods for representing this A matrix with real values using a block diagonal A matrix.

Multivariable Systems

The advantage of the state-space system is that it can easily be extended to multivariable systems. If the system has nu inputs, nx state variables, and ny outputs, then $\boldsymbol{u}$ and $\boldsymbol{y}$ are nx by 1 and ny by 1 column vectors. x is the nx by 1 state vector. A, B, C, D are matrices of appropriate dimensions. A is a square nx by nx matrix. The dimensions of the B, C, and D matrices are nx by nu, ny by nx, ny by nu, respectively. Equation (46) shows the general format.

$$\begin{aligned}
\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_{nx} \end{bmatrix} &= \begin{bmatrix} a_{1,1} & \cdots & a_{1,nx} \\ \vdots & \cdots & \vdots \\ a_{nx,1} & \cdots & a_{nx,n} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{nx} \end{bmatrix} \\
&\quad + \begin{bmatrix} b_{1,1} & \cdots & b_{1,nu} \\ \vdots & \cdots & \vdots \\ b_{nx,1} & \cdots & b_{nx,nu} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_{nu} \end{bmatrix} \\
\begin{bmatrix} y_1 \\ \vdots \\ y_{ny} \end{bmatrix} &= \begin{bmatrix} c_{1,1} & \cdots & c_{1,nx} \\ \vdots & \cdots & \vdots \\ c_{ny,1} & \cdots & c_{ny,nx} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{nx} \end{bmatrix} \\
&\quad + \begin{bmatrix} d_{1,1} & \cdots & d_{1,nu} \\ \vdots & \cdots & \vdots \\ d_{ny,1} & \cdots & d_{ny,nu} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_{nu} \end{bmatrix}
\end{aligned} \tag{46}$$

There exists a transfer function for each input and output pair shown. The result is a transfer function matrix. We can no longer obtain the output over input ratio of $\boldsymbol{y}(s)/\boldsymbol{u}(s)$, since $\boldsymbol{u}(s)$ and $\boldsymbol{y}(s)$ are no longer scalars. They are column vectors. We can calculate the output over input ratio of each input-output pair. The transfer function matrix can be obtained as described for single-input, single-output systems. Take the Laplace transform that results in the replacement of $\dot{\boldsymbol{x}}$ with $s\boldsymbol{x}$, and solve for $\boldsymbol{y}(s)$ as a function of $\boldsymbol{u}(s)$. The following equation results:

$$\boldsymbol{y}(s) = [C[sI - A]^{-1}B + D]\boldsymbol{u}(s) \tag{47}$$

where I is an identity matrix of the appropriate dimension and $[sI - A]^{-1}$ is the matrix inverse of $[sI - A]$. For A matrices larger than 3 by 3, this is a difficult inverse to perform symbolically and is normally only performed numerically at discrete values of frequency ω, after substituting for $s = j\omega$. The result is a matrix of transfer functions:

$$\boldsymbol{y}(s) = \begin{bmatrix} \dfrac{y_1(s)}{u_1(s)} & \cdots & \dfrac{y_1(s)}{u_i(s)} & \cdots & \dfrac{y_1(s)}{u_{nu}(s)} \\ \vdots & & & & \vdots \\ \dfrac{y_j(s)}{u_1(s)} & \cdots & \dfrac{y_j(s)}{u_i(s)} & \cdots & \dfrac{y_j(s)}{u_{nu}(s)} \\ \vdots & & & & \vdots \\ \dfrac{y_{ny}(s)}{u_1(s)} & \cdots & \dfrac{y_{ny}(s)}{u_i(s)} & \cdots & \dfrac{y_{ny}(s)}{u_{nu}(s)} \end{bmatrix} \boldsymbol{u}(s) \tag{48}$$

where each element of the matrix is a transfer function.

SIMULATION OF LINEAR DYNAMIC SYSTEMS

A state-space system in Eq. (46) can be simulated with the addition of a numerical integration routine. The general nonlinear case is shown in Eq. (49). Equation (46) is obtained through the multivariable Taylor's series expansion of Eq. (49).

$$\begin{aligned} \dot{x} &= F(x, u, t) \\ y &= G(x, u, t) \end{aligned} \tag{49}$$

Using a numerical integration scheme, the value of x can be obtained at each timestep. The timestep has to be selected small enough such that the system dynamics are properly represented and the simulation plus integration is numerically stable. Typical numerical integration routines are the Euler and Runge–Kutta routines.

CONTROL SYSTEM DESIGN AND ANALYSIS

It is the pole and zero locations along with the gain that determine the transient response of any transfer function. Control system design basically results in the manipulation of the poles, zeros, and gain, although we are not much concerned with their location as we are with receiving the desired response and system robustness. Design procedures have been built around both the Nyquist and Bode plots, which represent the frequency response of a transfer function. Both phase and gain margin are used to check the robustness of single-input, single-output systems, but a discussion of phase and gain margin is outside the scope of this article. The topic on control system design should be examined for more details.

EXPERIMENTAL IDENTIFICATION OF DISCRETE TRANSFER FUNCTIONS

System models can be obtained using various methods, but the two primary methods are (1) to model the system physics

using differential equations, and (2) to identify the system dynamic using dynamic data measured from the system. The first method addresses the physical relationships between all the components that make up the system. The second approach takes measured data from an existing system, assumes a model structure, and optimizes the model parameters to fit that measured data. The identified model is typically a discrete time model since the data are typically sampled, but a continuous time model could also be derived from the data. In this section we are interested in the identification of discrete models. These models are sometimes called autoregressive (AR) and autoregressive moving average (ARMA) models, but there are other names depending on the model structure and parameter optimization scheme used. You will see this type of model in the articles on finite impulse response (FIR) digital and adaptive filters and linear systems. Equation (50) is an example of a general linear, constant coefficient difference equation.

$$\begin{aligned} a_0 y(k-d) + a_1 y(k-d-1) + \cdots + a_n y(k-d-n) \\ = b_0 u(k) + b_1 u(k-1) + \cdots + b_m u(k-m) \\ \sum_{i=0}^{n} a_i y(k-d-i) = \sum_{j=0}^{m} b_j u(k-j) \end{aligned} \tag{50}$$

Equation (2) is an example of a first-order difference equation with $n = 1$, $d = 0$, $a_0 = 1$, $a_1 = 0.-92$, and $m = 0$, and $b_0 = 16$. When identifying a discrete model of the type in Eq. (50), there are generally three steps. The first step is to analyze the system response to a step input to ascertain if there is any time delay between the input and when the effect of that input appears in the output. This time delay is represented by the value of d in Eq. (50). The second step is the selection of the structure for the model. The structure is defined by the values of n and m, which correspond to the order of the two polynomials in the coefficients a_i and b_j. The third step is parameter estimation or optimization. Parameter estimation solves for the parameters (coefficients in the polynomials) in order to reduce the error between the response of the real system and the discrete model. There are various optimization routines for solving for the parameters. We will briefly examine each of these three steps in the following sections.

Delay Estimation

The time delay, d, represented in Eq. (50) can be obtained from the system step response. Figure 14 repeats the automotive velocity model step response but includes a time delay of 2.5 samples. With experimental step response data you can measure d. However, d has to be an integer number, so you have two choices: either approximate d as either 2 or 3, or increase the sampling rate to 2 Hz (decrease the sampling period, $T = 0.5$ s). One of the difficulties with time delays is that when they occur, they are typically not constant. You are more likely to see a time delay in a fluid or thermal system than in a mechanical or electrical system.

Model Structure Selection

Model structure section comes down to selecting the mathematical representation that will be used in the parameter estimation scheme. Most parameter estimation schemes are based on model structures that are linear in the parameters or coefficients. The model variables do not have to be linear, but in this section on transfer functions, we only consider linear models of the type shown in Eq. (50). Given this restriction, model structure selection comes down to selecting the order of the polynomials, the values for n and m in Eq. (50). Typically this is done based on experience and trial and error for simple systems. A model order is selected and then the parameter estimation scheme is executed. The model order can be increased until there is no improvement in the average error between the model and the data used to fit the system. There have been theories developed on model order selection, but these theories involve the concepts of probability and random processes, which are beyond the scope of this article. The Akaike's Baysian information criterion is one such example. In the following section on parameter estimation we will assume that the model order has already been selected.

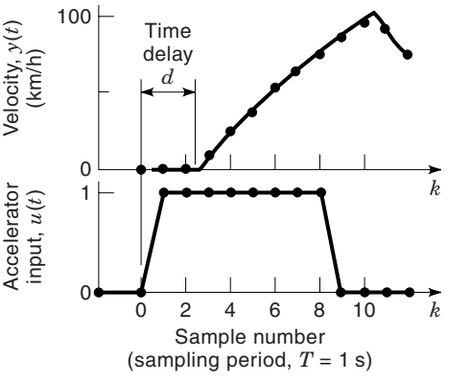


Figure 14. Automotive velocity response showing an example of how the discrete time delay would be estimated.

Parameter Estimation

There are articles in this encyclopedia on self-tuning regulators, adaptive control, recursive filters, parameter estimation, least squares approximation, and recursive estimation, so we will just refer the reader to those articles. Basically parameter estimation comes down to an optimization scheme to solve for the coefficients in the model shown in Eq. (50) based on some criterion. There are various criteria that account for the various parameter optimization schemes. Recursive methods were developed for on-line parameter estimation. Recursive schemes require less computation and less memory compared to their nonrecursive counterparts, but they require computations at each timestep. The recursive scheme came about with the advent of adaptive control. Adaptive control attempts to address slow variations in a plant's dynamic by identifying a model on-line. *Slow* in this instance is relative to the system characteristics. For example, in the automotive acceleration model described earlier, a slow variation might be the difference caused by adding passengers or a load to the vehicle. This load variation does not occur at the same rate as the change in velocity.

Research in the Area of Transfer Functions. Transfer functions are just a modeling technique for single-input, single-output systems, and they are based on the linear, constant coefficient differential and difference equations and the Laplace and Z transforms. These mathematical concepts are well defined and mature, so there are no new developments in the area of transfer functions themselves. There is research in many areas of linear systems that use transfer functions concepts. We will just mention a few areas of research.

We have mentioned the use of least squares as an optimization method for estimating the parameters that make up a discrete time transfer function identified model of a system. Least squares is one optimization method. There have been many developments in different optimization approaches to parameter estimation that offer performance improvements for a particular application. Some of these approaches are stochastic in nature and they consider the measured variables as random variable and random processes (see PROBABILITY). The research areas of linear system and linear control system design have been active with robust and μ-synthesis control design techniques and the linear matrix inequalities (LMI) approach to solving optimization problems in linear systems. Much of the systems and control research has moved beyond the restriction of linear, constant coefficient systems. For example, one of the newer approaches to system identification uses genetic programming to solve for the system structure and a nonlinear ordinary differential system (see GENETIC ALGORITHMS). Also there have been developments in the area of system modeling based on chaos and wavelets (see WAVELETS).

BIBLIOGRAPHY

K. Astrom, Theory and applications of adaptive control—A survey, *Automatica,* **19**: 471–486, 1983.

O. H. Bosgra and H. Kwakernaak, Design Methods for Control Systems, course notes of the Dutch Institute of Systems and Control, Winter term, 1996–1997, Portable Document File (PDF) Internet Resource: http://www.math.utwente.nl/disc/dmcs/, April 26, 1998.

E. O. Doebelin, *System Dynamics Modeling and Response,* Columbus, OH: Charles E. Merrill, 1972.

G. Franklin, J. Powel, and A. Emami-Naeini, *Feedback Control of Dynamic Systems,* 3rd ed., Reading, MA: Addison-Wesley, 1994.

G. C. Goodwin and K. S. Sin, *Adaptive Filtering, Prediction and Control,* Englewood Cliffs, NJ: Prentice Hall, 1984.

R. Isermann, Parameter adaptive control algorithms—A tutorial, *Automatica,* 513–528, 1982.

R. Isermann, Practical aspects of process identification, *Automatica,* **16**: 575–587, 1982.

K. Ogata, *Modern Control Engineering,* Englewood Cliffs, NJ: Prentice Hall, 1970.

A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing,* Englewood Cliffs, NJ: Prentice Hall, 1975.

D. Ridgely and S. Banda, Introduction to Robust Multivariable Control, US Air Force Wright Aeronautical Lab, Wright-Patterson Air Force Base, Dayton, OH 45433, AFWAL-TR-85-3102, 1986.

R. Rosenberg and D. Karnopp, *Introduction to Physical System Dynamics,* New York: McGraw-Hill, 1983.

R. E. Skelton, *Dynamic Systems Control: Linear Systems Analysis and Synthesis,* New York: Wiley, 1988.

R. E. Skelton, *A Unified Algebraic Approach to Linear Control Design,* New York: Wiley, 1988.

B. Wittenmark and K. Astrom, Practical issues in the implementation of self-tuning control, *Automatica,* **20** (8): 595–605, 1984.

DUANE MATTERN
Mattern Engineering, Controls and Hardware

TRANSFERRED-ELECTRON DEVICES. See GUNN OR TRANSFERRED-ELECTRON DEVICES.

TRANSFORMABLE COMPUTING. See CONFIGURABLE COMPUTING.

TRANSFORMATIONS, GRAPHICS 2-D. See GRAPHICS TRANSFORMATIONS IN 2-D.

TRANSFORMER, DC. See DC TRANSFORMER.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Traveling Salesperson Problems
Standard Article

Wei Lin[1], José G. Delgado-Frias[2], Donald C. Gause[2]
[1]Coopers & Lybrand LLP, Westport, CT
[2]State University of New York at Binghamton, Binghamton, NY

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2463
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (207K)

Abstract

The sections in this article are

Evolutionary Computation

Holland's Fundamental Theorem and the TSP Schemata

Hybrid Newton–Raphson Genetic Algorithm

| | |

Copyright © 1999-2008 All Rights Reserved.

TRAVELING SALESPERSON PROBLEMS

Many problems of both practical and theoretical importance deal with the search for an optimal solution as defined by Papadimitriou and Steiflitz (1). Wilde and Beightler (2) have identified three major requirements for an optimization method: (1) determine precisely the problem variables and their interaction, (2) develop a measure of problem effectiveness expressible in terms of the problem variables, and (3) choose those values of the problem variables that yield better solutions.

There exist several classes of problems; Pierre (3) provides mathematical definitions of some of them. Optimization problems are usually divided into two main categories: those with continuous variables and those with discrete (combinatorial) variables. In problems with continuous variables, a set of real numbers or a function that provides a solution is generally looked for. In combinatorial problems, a solution that contains a finite integer set, a permutation set, or a graph is searched for. In optimization designs, only certain feature combinations are possible. This means that the possible solutions are restricted to a subregion. The function may be designed to focus on the subregion. The goal of optimization is to find the solution in this subregion for which the function obtains its smallest value, which is referred to by Torn and Zilinxkas (4) and other researchers as the *global minimum.*

Among optimization problems, the traveling salesperson problem (TSP), classified as an NP-complete problem, has been widely studied. For this problem, no algorithm has been

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

demonstrated to find the optimum solution in polynomial time. In a 20-city TSP, there are roughly 6×10^{16} possible tours [20!/(2 × 20)]. Many other problems have similar complexity, such as computer wiring, wallpaper cutting, and job sequencing. Consequently, the solution of this NP-complete problem has been the subject of a large amount of work. The TSP is one of the most prominent of the unsolved combinatorial optimization problems and the most common basis for comparisons. According to Lawler, et al. (5), the TSP continues to influence the development of optimization concepts and algorithms.

The TSP consists of two sets: a set of cities $V = \{1, \ldots, n\}$ and a set of links A. The links are represented by pairs $(i, j) \in A$, meaning that there is a link between city i and city j. The travel distance between city i and city j is expressed as c_{ij}. The problem is to find a tour starting at any city, visit every city exactly once, and return to the starting city. This tour should take the least total traveled distance.

To formulate this problem, a variable x_{ij} is introduced; where $x_{ij} = 1$ if j immediately follows i on the tour and $x_{ij} = 0$ otherwise. The requirement that each city be entered and left exactly once is stated as

$$\sum_{i:(i,j)\in A} x_{ij} = 1 \qquad \text{for} \quad j \in V \tag{1}$$

$$\sum_{j:(i,j)\in A} x_{ij} = 1 \qquad \text{for} \quad j \in V \tag{2}$$

The above constraints are not sufficient to define a tour, since a subtour can satisfy them as well. One way to eliminate subtours is to introduce another constraint. For each subset $U \subset V$, $2 \le U \le V - 2$, the constraint is that

$$\sum_{(i,j)\in A:\, i\in U,\ \ j\in V\setminus U} x_{ij} \ge 1 \qquad \text{for} \quad j \in V \tag{3}$$

The TSP can be formulated as

$$\min\left\{\sum_{(i,j)\in A} c_{ij}x_{ij} : x \text{ satisfies Eqs. } (1)-(3)\right\} \tag{4}$$

The number of constraints is nearly $2^{(V)}$.

There are three major types of methods that deal with the TSP, namely, neural networks, heuristic searches, and genetic algorithms. Neural networks potentially offer a powerful tool for solving combinatorial problems. This approach has an embedded parallelism that potentially can be implemented in hardware. The original work can be traced back to Hopfield and Tank (6). The proposed method is to encode the TSP into a two-dimensional neuron array. The approach relies on a fully connected artificial neural network. This type of approach has been presented and evaluated by a number of researchers, among them Aiyer (7), Sanchez-Sinencio and Lau (8), Pretzel et al. (9), and Lin (10).

The other algorithms generally rely on some sort of heuristic or an intelligent guess to find good solutions in reasonable time. The most common techniques range from the simple greedy algorithm to the more complex branch-and-bound search and Lin–Kernighan (11) algorithms. A comprehensive account of such techniques for the TSP can be found in Lawler et al. (5). In general, heuristic algorithms provide a fast approach to a lower bound.

Genetic algorithms mimic natural evolution as a population-based optimization process. They are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured and randomized information exchange to form a search algorithm. In every generation, a new set of artificial chromosomes is created using pieces of the fittest chromosome of the previous generation. Genetic algorithms efficiently exploit historical information to speculate on new search points with expected improved performance (3). These algorithms generally provide good solutions at low speed. Hybrids of heuristic and genetic algorithms for the TSP are presented here, with mathematical models as well as performance results.

EVOLUTIONARY COMPUTATION

Evolutionary computation is the most commonly used term to describe the new computing paradigm that mimics natural evolution. Evolutionary computation comprises three major fields: genetic algorithms, evolutionary strategies, and evolutionary programming. Each evolutionary computation algorithm uses similar operations. Each begins with a population of contending trial solutions for the task at hand. New solutions are created by altering the existing solutions using a set of evolutionary computation operators. An objective measure called *fitness* of the trial solution is used to evaluate the new solution. A selection mechanism determines which solutions to maintain as *parents* (or *seeds*) for the subsequent generation. The differences between the procedures are characterized by the types of admissible alterations on parents to create offspring and the methods employed for selecting new parents; readers are referred to Fogel (12,13) and Srinivas and Patnaik (14). In natural evolution, genes on chromosomes carry the environment-fitting information of the species across generations. In genetic algorithms, such information is represented by a fitness function (competitive selection), and chromosomes provide the link between generations. If a chromosome does not fit well within the current environment, it will become extinct. The adaptive procedure is implemented by means of mutation, inversion, or crossover operators. Below the basic genetic algorithm and its operators are presented.

Genetic Algorithm Approaches

The simplest form of a genetic algorithm includes reproduction and selection according to Goldberg (15). Each offspring's fitness value is used as a criterion to perform the selection of the potential new parents (seeds). The basic procedure of the genetic algorithms is shown below in a programlike format. In this procedure t indicates the current generation, and the number $P(t)$ represents the population at generation t:

```
begin
    t ← 0;
    Initialize P(t0);
    Evaluate P(t0);
    while no_termination
        begin
            t ← t + 1;
            Select P(t) from P(t − 1);
```

```
        Recombine from P(t)
        Evaluate P(t);
    end
end.
```

Goldberg (15) has identified three basic recombination operators in genetics algorithms, namely crossover, inversion, and mutation. These three operators are described in the following sections.

Crossover Operator. The crossover operator requires two chromosomes. These chromosomes exchange a number of genes. The exchanged genes maintain the same relative position with respect to each other. The steps that are required for a crossover operation are listed below. In this description, the chromosomes are referred as strings:

1. Select two mating strings according to a selection policy.
2. Select two points for each string (this selection is usually random).
3. Swap the piece of string within these two points between the two strings.

Figure 1 illustrates the crossover operation. In this case there are two chromosomes called A and B. Each chromosome has 10 genes (which are numbered from 1 to 10). Although some applications allow a duplication of some genes in the same chromosome, for the TSP in general it is considered that each chromosome has a unique set of genes. Figure 1(a) shows the original two chromosomes A and B; the points 1 and 2 are randomly selected. The resulting new chromosomes are shown in Fig. 1(b).

Inversion Operator. The inversion operator works on a chromosome by reversing the sequence of a substring of genes. The steps to follow are:

1. Select a string.
2. Select two points in the string.
3. Invert the piece of the string between the two points.

Figure 2 shows this operation.

Mutation Operator. The mutation operator requires a chromosome (or string) to operate on. The objective of this operator is to slightly disrupt the current chromosome by inserting a new gene. The steps that are followed in the implementation of this operator are listed below:

1. Select a string.
2. Select a site in the current chromosome.
3. Obtain a new gene from a gene pool.
4. Replace this site with the new gene.

Figure 3 shows the mutation operation.

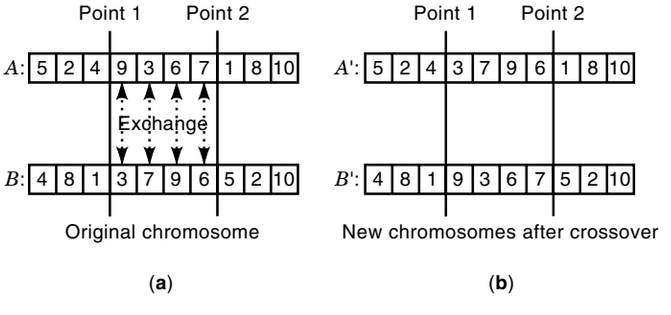


Figure 1. Standard crossover operator.

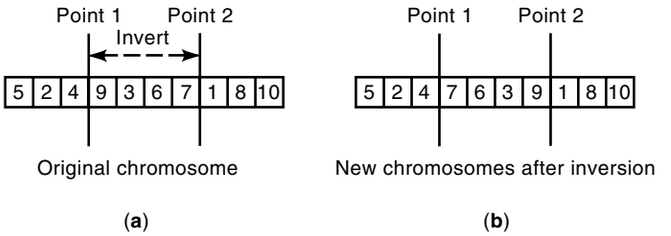


Figure 2. Standard inversion operator.

Genetic Approaches for the Traveling Salesperson Problem

Several genetic approaches to the TSP have been introduced. These approaches include *partially matched crossover* (PMX) by Goldberg (15), *cycle crossover* (CX) by Oliver et al. (16), *order crossover* (OX) by Goldberg (15), *edge recombination* by Whitley et al. (17), *matrix crossover* (MX) by Homaifar et al. (18), and *evolutionary programming* by Fogel (19). These approaches are described in the following sections.

Goldberg Partially Matched Crossover Approach. PMX, introduced by Goldberg (15), has been considered as a way to tackle a blind TSP. The blind TSP approach is not aware of the distance between the cities. The total traversed distance is obtained only at the end of the tour. The application of PMX to the TSP follows the procedure provided below; an example is shown in Fig. 4:

1. Code a tour as a chromosome.
2. Select two chromosomes as parents. PMX proceeds by positionwise exchange. Two crossing sites are picked from a uniform distribution along the string.
3. Map chromosome 2 to chromosome 1. In the example (Fig. 4), *ACHB* (of chromosome 2) exchange places with *DEFG* (of chromosome 1).
4. Use a correction procedure to legitimize the TSP path.

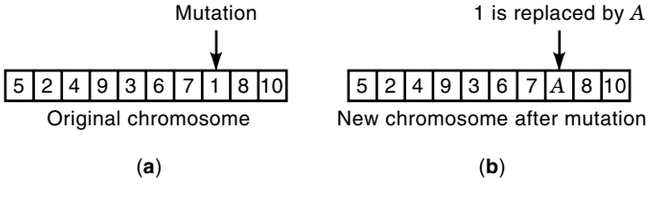


Figure 3. Standard mutation operator.

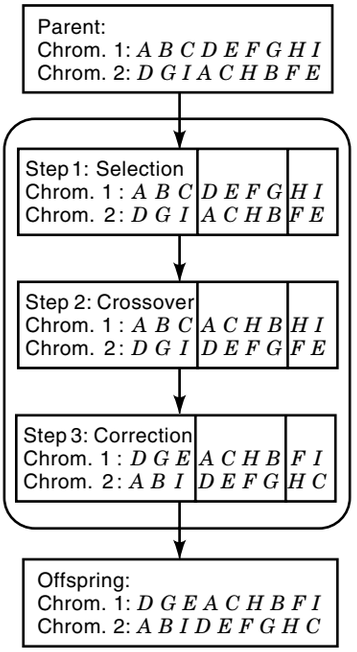


Figure 4. PMX operation.

The benefit of this crossover approach is that each string contains ordering information determined by both its parents. However, the algorithm does not use the distance information.

Oliver Cycle Crossover Approach. The CX approach, introduced by Oliver et al. (16), creates offspring from a pair of parents. Every element of an offspring comes from one of the two parents; the position of each element is identical to that in one of the parents. However, the offspring will be different from both parents. Two rules can be imposed. One is that every position of the offspring must retain a value found in the corresponding position of a parent, and the other is that the offspring must contain a permutation. The operation procedure is explained by means of an example shown in Fig. 5. The steps are as follows:

1. Select two strings as parents as shown in Fig. 5.
2. Randomly select a starting point. Assuming C is selected, the relative position of C in the other chromosome corresponds to I. I is selected in chromosome 1. The relative position of I in chromosome 2 corresponds to E. E is selected from chromosome 1. The relative position of E in the chromosome 2 corresponds to C. Since C has been already selected, a complete cycle is formed.
3. Preserve the selected genes in chromosomes 1 and 2.
4. Switch the selected genes between the two chromosomes.

In a standard crossover operator, a random crossover point, or cut section, is chosen. In the cycle crossover a random parent is chosen for each cycle. The absolute positions of both parents are preserved.

Order Crossover Approach. The OX approach requires two strings. The order crossover starts in a similar way to PMX by selecting a range of genes. Figure 6 shows an example of the use of OX. Below a description of this approach is provided:

1. Randomly select a section of genes as crossover segment.
2. Exchange string 1's segment and string 2's segment.
3. Replace the illegal city (visited twice) with a empty entry.
4. Shift those empty entries together at the opposite ends of each string.
5. Replace the empty entries with the corresponding segment in the other string.

The absolute positions are preserved.

Whitley Edge Recombination. Another operator, introduced by Whitley et al. (17), constructs an offspring tour by exclusively using links present in the two parent structures. On average, these edges should reflect the goodness of the parent structures. There is no random information that might drive the search toward any arbitrary links in the search space. Operators that break links introduce unwanted mutation, which can change the search processing. The edge recombination operator uses an edge map to construct an offspring that inherits as much information as possible from the parent structures. The procedure for this approach is listed below:

1. Assume that there are two strings of chromosomes in the parent, string 1 and string 2, selected to recombine:

   String 1: $A\ B\ C\ D\ E\ F\ G\ H\ I$
   String 2: $D\ G\ I\ A\ C\ H\ B\ F\ E$

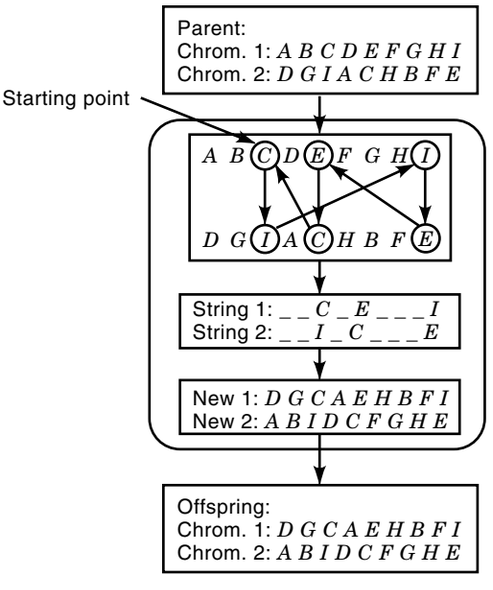


Figure 5. CX operation.

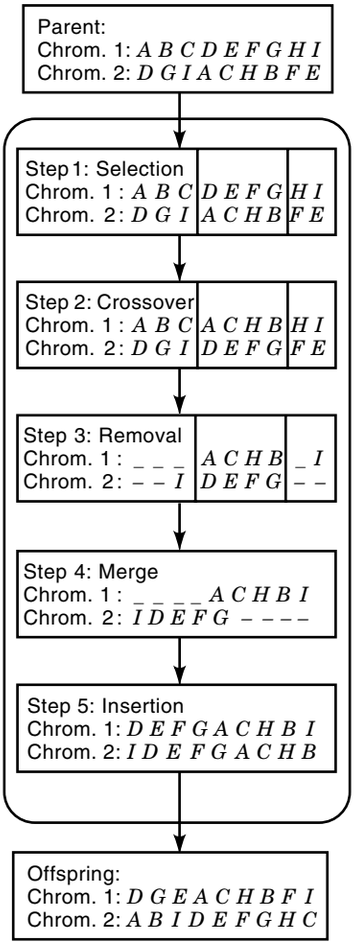


Figure 6. OX operation.

2. Select the city with the largest number of connections, or randomly select one in the case of a tie.
3. Select the city with the fewest connections first, to prevent its isolation.
4. This processing continues until the tour is constructed.

A time step table for the new offspring generation is shown in Table 1. Initially, *B*, *C*, *G*, and *H* contain four links. Assume that *B* is randomly selected as starting city. Then *B* will be eliminated from all the edge maps. At step 1, since *A*, *E*, and *F* contain only two links, assuming *A* is randomly selected as the second city to visit, then *A* is eliminated from all edge maps. At step 2, *C*, *E*, *F*, *I* contain two edges, assuming *F* is randomly selected, then *F* is eliminated from all edge maps. At step 3, *E* contains one link. Then *E* is selected and eliminated from all edge maps. At step 4, since *C*, *D*, *G*, *I* contain two edges, the randomly selected city is *D*; thus *D* is eliminated from all edge maps. At step 5, since *C* contains one edge, *C* is selected and eliminated from all edge maps. At step 6, *G*, *H*, and *I* contain two edges, and the randomly selected city is *I*; then *I* is eliminated from all edge maps. At step 7, since *G* and *H* contain only one edge each, the randomly selected city is *G*, which is eliminated from all edge maps. *H* becomes the last city to visit, and the tour goes back to city *B*. The tour construction becomes *B*, *A*, *F*, *E*, *D*, *C*, *I*, *G*, *H*. Comparing with the parent strings (*A B C D E F G H I* and *D G I A C H B F E*), we see that the entire edges of the offspring are taken from both parents except *A F* and *C I*. For *A F* and *C I*, two new edges are introduced in the tour. This is called *edge failure*. For a large TSP, the edge failures as reported by Whitley amounts to less than 2%, which is similar to a conventional mutation rate. Based on the experiments, Whitley proposed that rather than optimizing for the positions in order, the algorithm should probably allocate more reproductive trials to high-performance edges and find the critical edge combination.

Homaifar Matrix Crossover. Homaifar et al. (18) proposed a different TSP representation method. Instead of a string representation, a binary matrix is used to represent edges directly. The applied crossover is a conventional one. For this representation, the tour and procedure are now presented, using a nine-city problem as example:

1. Code strings into binary matrices for the parents. String 1 (*A B C D E F G H I*) and string 2 (*D G I A C H B F E*) are thus coded into 2-D matrices. Figure 7 shows the chromosomes.
2. Assuming the crossover sites are indicated by the arrows for string 1 and string 2, exchange the genes using the crossover operation. (This step is shown in Fig. 8.)
3. Any invalid tour is corrected by moving the duplicated 1's from the row to another row that does not have any 1. The correction is done to preserve existing edge as

Table 1. Time Step Table for the Edge Recombination

City	Initial	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
A	*B, I, C*	*I, C*							
B	*A, C, H, F*								
C	*A, B, D, H*	*A, D, H*	*D, H*	*D, H*	*D, H*	*H*			
D	*C, E, G*	*C, E, G*	*C, E, G*	*C, E, G*	*C, G*				
E	*D, F*	*D, F*	*D, F*	*D*					
F	*B, E, G*	*E, G*	*E, G*						
G	*D, F, H, I*	*D, F, H, I*	*D, F, H, I*	*D, H, I*	*H, I*	*H, I*	*H, I*	*H*	*H*
H	*B, C, G, I*	*C, G, I*	*C, G, I*	*C, G, I*	*C, G, I*	*C, G, I*	*G, I*	*G*	
I	*A, G, H*	*A, G, H*	*G, H*	*G, H*	*G, H*	*G, H*	*G, H*		

String 1

	A	B	C	D	E	F	G	H	I
A	0	1	0	0	0	0	0	0	0
B	0	0	1	0	0	0	0	0	0
C	0	0	0	1	0	0	0	0	0
D	0	0	0	0	1	0	0	0	0
E	0	0	0	0	0	1	0	0	0
F	0	0	0	0	0	0	1	0	0
G	0	0	0	0	0	0	0	1	0
H	0	0	0	0	0	0	0	0	1
I	1	0	0	0	0	0	0	0	0

Coding parent 1 into matrix 1

(**a**)

String 2

	A	B	C	D	E	F	G	H	I
A	0	0	1	0	0	0	0	0	0
B	0	0	0	0	0	1	0	0	0
C	0	0	0	0	0	0	0	1	0
D	0	0	0	0	0	0	1	0	0
E	0	0	0	1	0	0	0	0	0
F	0	0	0	0	1	0	0	0	0
G	0	0	0	0	0	0	0	0	1
H	0	1	0	0	0	0	0	0	0
I	1	0	0	0	0	0	0	0	0

Coding parent 2 into matrix 2

(**b**)

Figure 7. Coding of parents 1 and 2 into matrices.

much as possible. The resulting matrices are shown in Fig. 9. The offspring from this procedure are as follows:

String 1b: *A B F E D C G H I*

String 2b: *A C D E F G I* and *B H*

4. Since the binary matrix representation may produce an illegal tour, additional correction in the binary matrix may be needed. The correction procedure is performed with the objective of preserving as many existing edges as possible. For string 2b (*A C D E F G I* and *B H*), the correction can be made by comparing the parent tour edges and modifying the string to *A B H C D E F G I*. Figure 10 shows the changes to the matrix for string 2b.

Homaifar et al. (18) suggested that the specifics of a genetic algorithm's implementation, including its representation, play an important role in its ability to satisfactorily solve the TSP or any other such problem.

Fogel Evolutionary Programming. As described earlier, evolutionary programming is a subset of evolutionary computation. Evolutionary programming emphasizes the level of the species rather than that of the chromosome operation. Genetic algorithms, in general, may be viewed as bottom-up procedures that combine building blocks of code to arrive at superior solutions, whereas evolutionary programming is a top-down procedure that tries to optimize all parameters of a cohesive interactive code simultaneously.

Fogel (19) has proposed an inversion operation for the TSP. The procedure uses *simulated annealing*. From observing natural systems, Fogel suggests that the behavior difference across generations decreases over time as species become better predictors of their environment. The approach is described below:

1. Code the TSP in an ordered list.
2. Create a population that consists of N parent solutions. Each parent produces a single offspring through mutation.
3. Linearly decrease the maximum inversion length from one-half of the string down to neighbor inversion at the maximum number of evaluated offspring.
4. The best N solutions are retained at each iteration to become parents for the next generation.

Fogel (19) suggests that it is crucial to maintain a behavioral link between parent and offspring as behavior evolves.

String 1b

	A	B	C	D	E	F	G	H	I
A	0	1	0	0	0	0	0	0	0
B	0	0	0	0	0	1	0	0	0
C	0	0	0	0	0	0	1	0	0
D	0	0	1	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0
F	0	0	0	0	1	0	0	0	0
G	0	0	0	0	0	0	0	1	0
H	0	0	0	0	0	0	0	0	1
I	1	0	0	0	0	0	0	0	0

String 2b

	A	B	C	D	E	F	G	H	I
A	0	0	1	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	0	0	1	0	0	0	1	0
D	0	0	0	0	1	0	1	0	0
E	0	0	0	0	0	1	0	0	0
F	0	0	0	0	0	0	1	0	0
G	0	0	0	0	0	0	0	0	1
H	0	1	0	0	0	0	0	0	0
I	1	0	0	0	0	0	0	0	0

Figure 9. First-level correcting matrices 1 (from string 1a to string 1b) and 2 (from string 2a to string 2b).

String 1a

	A	B	C	D	E	F	G	H	I
A	0	1	0	0	0	0	0	0	0
B	0	0	1	0	0	1	0	0	0
C	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0
E	0	0	0	1	0	0	0	0	0
F	0	0	0	0	1	0	1	0	0
G	0	0	0	0	0	0	0	1	0
H	0	0	0	0	0	0	0	0	1
I	1	0	0	0	0	0	0	0	0

String 2a

	A	B	C	D	E	F	G	H	I
A	0	0	1	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0
C	0	0	0	1	0	0	0	1	0
D	0	0	0	0	1	0	1	0	0
E	0	0	0	0	0	1	0	0	0
F	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	1
H	0	1	0	0	0	0	0	0	0
I	1	0	0	0	0	0	0	0	0

Figure 8. Exchange of segments of matrix 1 and matrix 2 to form string 1a and string 2a.

String 2c

	A	B	C	D	E	F	G	H	I
A	0	0	1	0	0	0	0	0	0
B	0	0	0	0	0	0	0	1	0
C	0	0	0	1	0	0	0	0	0
D	0	0	0	0	1	0	0	0	0
E	0	0	0	0	0	1	0	0	0
F	0	0	0	0	0	0	1	0	0
G	0	0	0	0	0	0	0	0	1
H	0	1	0	0	0	0	0	0	0
I	1	0	0	0	0	0	0	0	0

Figure 10. Second-level correcting matrix for string 2b.

Remarks

PMX, CX, OX, edge recombination, and MX methods use a mating procedure and a crossover operator that are complex. The evolutionary improvement between parents and offspring across generations relies on the number of parents' good chromosome segments (namely, schemata) that are inherited by the offspring. Parents' search information is merged into their offspring; thus, both parents' information on the environment appears in the offspring. In the TSP, most of the offspring tours generated by those genetic permutation operators are illegitimate. A correcting procedure is usually needed to purge the offspring of inadmissible conditions. The correcting procedure produces mutationlike effects. Most of the information from parents is scrambled. The link information between cities is not preserved when an offspring is generated.

For instance, in the MX approach there are two levels of correction. The coding method applied in binary MX and inversion needs a matrix of N^2 elements to represent one tour (chromosome), where N is the number of cities. Thus, the memory requirements are proportional to $N^2 \times$ (population-size).

The operation of MX is complex; it includes expensive correcting procedures. For the first level of correction, each row and column can have only one element set to 1. This correction procedure is based on the parent connection edges; thus, a comparison of parents' edges and possible offspring correction are required. At the second level, all the connections need to be checked to prevent a subtour condition from occurring. If a subtour exists, its removal is based on the parent tour. This type of computation grows extremely fast as the problem size increases. Edge recombination constructs a chromosome according to the linkage of the parents. However, the offspring tour closely reflects the number of links of each city in the parents' tours, and new edges are created from time to time. This approach may be considered as clustering two cities as one and randomly arranging those two-city clusters. The inheritance is kept fairly well. However, the operator is complex.

The crossover operator is the most commonly used operator. Qi and Palmieri (20,21) have shown that this operator has good characteristics for searching on the solution surface. Nevertheless, this operator gives a poor performance on the TSP type of permutation problem, as a result of the inadmissible tours that occur occasionally.

In addition to the crossover operator, inversion and mutation operators are also used in genetic algorithms. They not only produce legitimate offspring tours, but also are simple operators. They seem to have potential for TSP problems.

HOLLAND'S FUNDAMENTAL THEOREM AND THE TSP SCHEMATA

The most widely used theorem to explain the behavior of genetic algorithms is Holland's fundamental theorem, which is completely explained by Goldberg (15). This theorem is based on the concept of building blocks in a chromosome structure. A good chromosome should contain many well-organized sequences of gene structures. These building blocks are called schemata. The fundamental theorem is a mathematical tool used to interpret the formation of schemata and the accumulation of good schemata in the population of chromosomes over generations. Holland's fundamental theorem and two requirements for the design of novel genetic algorithms are presented here.

Holland's fundamental theorem facilitates the investigation of the behavior of genetic algorithms. A good solution is formed in the genetic algorithms over the generations by gradually aggregating many small well-fitted gene structures, called *schemata.* The number of schemata, $M(H)$, where H is the schemata, at a given generation t is denoted as $M(H, t)$. Holland's fundamental theorem shows that in a good genetic algorithm the number of schemata in the next generation, $M(H, t + 1)$, is larger. This is due to the genetic operations such as selection, genetic operation, and mutation. The schemata can be expressed as

$$\begin{aligned} M(H, t+1) \geq M(H, t) \times [&\text{fitness improvement} \\ &\text{(selection method)}] \\ &\times [\text{genetic operator survival rate}] \\ &\times [\text{mutation survival rate}] \end{aligned} \tag{5}$$

A fitness function f represents a search surface. Let H be a schema, and $f(H)$ be its fitness value. Let f_{avg} be the population-average fitness value. If the selection method is to choose those chromosomes with higher fitness value than f_{avg}, the ratio of the fitness values ensures that the schema number increases if the operator survival rates are not very low:

$$M(H, t+1) = M(H, t) \times \frac{f(H)}{f_{\text{avg}}} \times (\text{operator survival rates}) \tag{6}$$

To ensure that M increases and leads to a globally optimal solution, genetic operators should be able to search the solution surface properly. From the above, two basic requirements for designing applications based on genetic algorithms (or hybrid genetic algorithms) are obtained:

1. The selection method must have a large effect on the population.
2. The genetic operators affect the searching behavior and should provide a low disruption rate.

Qi and Palmieri (20,21) provide measurements for these requirements. For requirement 1, the selection method should show that

$$\lim_{k \to \infty} \frac{\sum f_{\text{poor}}\, g_k(x_1, \ldots, x_n)}{\sum f_{\text{well}}\, g_k(x_1, \ldots, x_n)} = 0 \tag{7}$$

where k is the generation number. Equation (7) means that as the number of generations increases, the chance of getting a poor fitness chromosome approaches zero. Thus the population should attain a high concentration of well-fitted chromosomes.

As for requirement 2, an operator with searching behavior can be evaluated as a statistically independent interaction of two chromosomes (or gene segments). Assuming chromosome A with $[x_1^a, x_2^a, \ldots, x_n^a]$ and chromosome B with $[x_1^b, x_2^b, \ldots, x_n^b]$, the joint probability density function of A and B is

$$f_{x_1, x_2, \ldots, x_n}(x_1, x_2, \ldots x_n) = f_{x_1}(x_1) f_{x_2}(x_2) \cdots f_{x_n}(x_n) \tag{8}$$

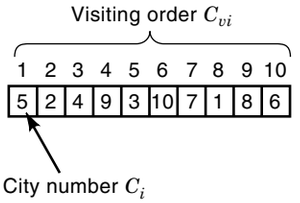


Figure 11. Ten-city TSP chromosome.

Under this condition, two chromosomes will expand their search range when a genetic operator is used. Those two chromosomes will gradually merge. Thus, their covariance will approach zero.

The disruption rate of the genetic operators is studied below along with the proposed genetic algorithms, since these operators affect the performance of the algorithms.

HYBRID NEWTON–RAPHSON GENETIC ALGORITHM

In this section an *inversion with embedded Newton–Raphson search* (IENS) is introduced. This algorithm is used as an example of genetic algorithms for the TSP. IENS uses an inversion genetic operator incorporating the Newton–Raphson method. By detecting the neighborhood tendency, the Newton–Raphson method is used to update the system variables. The algorithm combines the behavior of Newton–Raphson method, genetic algorithm neighborhood inversion, and standard inversion operators to perform a global and local search in the solution space. The fittest chromosome will become the parent of the next generation so as to seek the minimum by a Newton–Raphson update. Genetic algorithms provide powerful multiple-point search within a solution space. Through competition and reproduction, the genetic algorithm enables the Newton–Raphson operation to move around the solution surface.

Using a genetic algorithm (GA) for the solution of a problem often requires encoding the problem into a chromosome structure. An appropriate genetic-like manipulation of the chromosomes leads toward a solution of the problem. It is therefore important to choose a chromosome code that facilitates genetic-like manipulation. In this approach, a city is coded as a gene and the gene position in the chromosome is interpreted as the order of travel (adjacency representation) as introduced by Michalewicz (22).

Figure 11 shows an example of the chromosome coding for a 10-city TSP chromosome. In this example the first visited city is 5, which is followed by 2, 4, 9, 3, 10, 7, 1, 8, 6, and then 5 again. The gene pool needs be carefully selected. The gene represents the system's feature, character, or detector. The gene pool must contain sufficient information about the system. Since the TSP restriction is to visit each city exactly once and each gene represents a city, the size of the gene pool is the same as the city number. Termination is usually ensured by a ceiling on the number of generations.

For the TSP, given the chromosome representation, city ordering is the major concern. This in turn makes this problem a permutation one. The schemata are formed according to the relation between genes (cities) rather than the position within the chromosome. A permutation search genetic operator can be implemented by using an inversion. Whitely et al. (17) have used an inversion operator with reasonably good results. An inversion operator used on a 50-city TSP has outperformed a cross-and-correct operator. A drawback of using the inversion operator is that it takes information from only one parent. Thus, this approach does not provide alternative information that is often found in recombination. The search power that results from using recombination is not exploited. In order to compensate for this drawback, a numerical method has been embedded in the genetic search.

The IENS consists of five major components: the fitness function, GA operators, the operation mode, selection, and reproduction. The *fitness function,* used in this approach, is the distance of the tour. The chromosome structure provides information about the distance. A well-fitted chromosome should have a small distance (compared with its predecessors). In this approach there are three *GA operators:* inversion, neighborhood inversion, and mutation. The inversion operator randomly selects two nonneighbor genes (cities) in the chromosome and inverts the gene string. Neighborhood inversion selects two consecutive genes in the chromosome to perform a pairwise inversion. In this approach, mutation is done by randomly selecting two genes from the gene pool; these two genes swap their positions in the chromosome. The IENS *operation mode* can be summarized as having two operations: inversion search and neighbor inversion search. It is shown in Fig. 12.

The same search operation is used as long as it produces a better solution. After a threshold number of generations with no better solution, the other inversion operator is used on the best result in the history. The threshold can be determined by means of experimental runs. In the experiments and simulations the number of generations required was found to be less than 40; thus, the threshold was set to 40.

These two inversion search operations have a similar structure. The operations have five basic steps which are described below. These steps, also shown in Fig. 13, are:

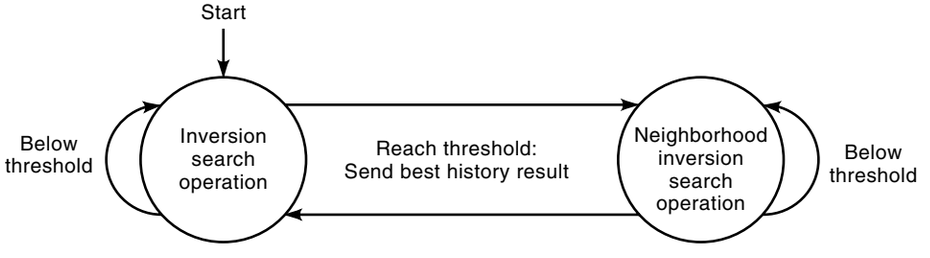


Figure 12. IENS operation mode.

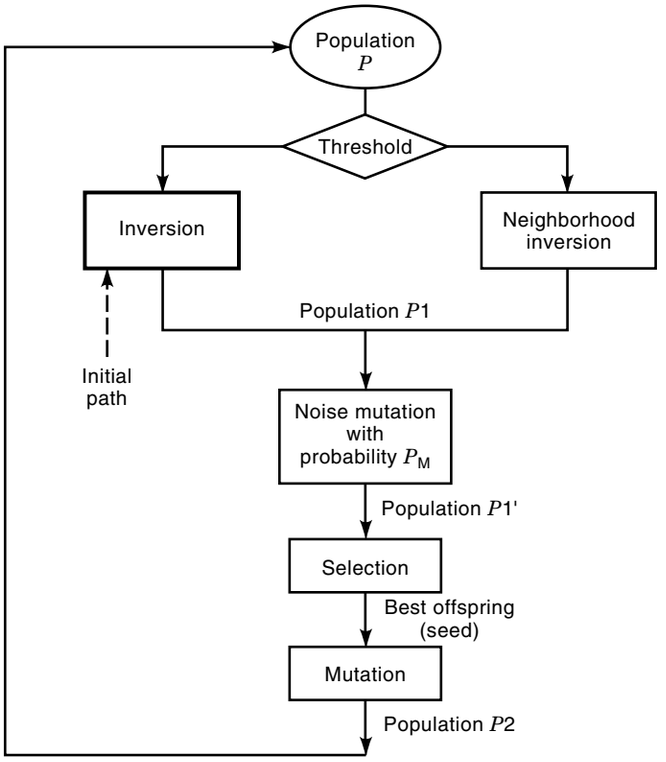


Figure 13. Inversion or neighborhood inversion operation steps.

1. Perform an inversion operation on the population P. This operation generates an offspring population P1.
2. Perform a noise mutation on offspring population P1, randomly with probability P_M. In general P_M is small, e.g., $P_M = 0.01$.
3. Evaluate P1′ and select the best offspring.
4. Generate population P2 using mutation as the reproduction operator with the best offspring as seed.
5. Replace P with P2.

The neighborhood inversion search operation is accomplished in a similar fashion to inversion search, except that the first step is replaced by the neighborhood inversion operation as shown in Fig. 13. Each parent can generate only one offspring. In both search operations an approach has been embedded to find the solution.

This approach mimics the Newton–Raphson method (23), numerical method used to solve the problem of making n functional relations equal to zero: $F_i(x_1, x_2, \ldots, x_m) = 0$, where $i = 1, 2, \ldots, n$. If X denotes the entire vector of values of x_i, then, in the neighborhood of X, each of the functions F_i can be expanded into a Taylor series as follows:

$$F_i(X + \delta X) = F_i(X) + \sum_{j=1}^{m} \frac{\partial F_i}{\partial x_j} \delta x_j + O(\delta X^2) \qquad (9)$$

By neglecting terms of order δX^2 and higher, a set of linear equations can be obtained. A change δX moves each function closer to zero when the system converges. That is,

$$\sum_{j=1}^{m} \alpha_{ij}\, \delta x_j = \beta_i, \qquad \text{where} \quad \alpha_{ij} = \frac{\partial F_i}{\partial x_j} \text{ and } \beta_i = -F_i \qquad (10)$$

If δX has a valid (finite) value, the correction becomes $x_i^{\text{new}} = x_i^{\text{old}} + \delta x_i$.

In the proposed genetic algorithm, each city is coded as a gene. Thus, F becomes a chromosome. The distance of a tour is given by $F_i(x_1, x_2, \ldots, x_m)$, where each x represents a city. If we denote the minimum distance by $D_{\min}$, then $F_i(x_1, x_2, \ldots, x_m) = D_{\min}$ represents the solution. The coefficient α_{ij} in the genetic algorithm represents a neighborhood inversion and is given by a derivative term. The neighborhood inversion acts as a local hill-climbing operator. For a given generation, i and j represent the ith chromosome and the jth gene. δx is a continuous function and provides information for the next solution. Since δx has discrete values in the genetic algorithm environment, a steepest-descent approach for δx can be used. Thus, x_i^{new} is equal to the maximum improvement in the β_i, that is, F_i. Since a steepest-descent approach is applied, the selection method is simplified to selecting only the best chromosome. To achieve reproduction, mutation is applied to generate the entire population from the best offspring. Using mutation prevents the population from settling far away from the current point. The population size has been set to be equal to the city problem size, i.e., $n = m$; the Newton–Raphson method needs the number of independent equations to be at least equal to the number of variables to have a solvable system.

To use the Newton–Raphson method, two major issues need be addressed: (1) the starting point must be near the solution and (2) the operation can easily be trapped in a local minimum. To avoid these problems, standard inversion is applied in the search process. Standard inversion provides large steps that can help not only to reach a near-optimal point but also to jump out of a trap (local minimum).

Each operation needs at least several generations to reach the solution or a stable state; thus, a generation threshold for each operation and generation counter must be set. A record of the best history is always kept. If the search result is better than the record, the record is updated and the generation counter is reset. If the search result is not better than the previous result, the generation counter is increased by one. When the generation counter reaches the generation threshold, the record is transferred to the other operation as the starting seed. When the other operation receives the seed, mutation is applied to generate the new population. Since there is no information about where the optimal solution resides, the initial chromosome is randomly generated.

Dynamics

Sirag and Weisser (24) have shown that when the inversion operates on a string chromosome, the behavior of this operator can be obtained by evaluating the survival rate of the O schemata. The O-schema number, as explained in Goldberg (15), provides information about the goodness of the solution. This approach is used to show the dynamics of IENS. It is assumed that the chromosome length is L and two distinct points A and B are selected to represent the ends of the inversion.

Standard Inversion. For standard inversion, A could be either larger or smaller than B. The expected survival rate E_A for a given A is expressed as

$$E_A = P(A > B|A)E_{A>B} + P(A < B|A)E_{A<B} \tag{11}$$

where $E_{A>B}$ and $E_{A<B}$ are the expected O-schema survival rates under the condition $A > B$ and $A < B$, respectively. For simplicity, the probabilities $P(A > B|A)$ and $P(A < B|A)$ will be abbreviated to $P(A > B)$ and $P(A < B)$ respectively. These probabilities are

$$P(A > B) = \frac{A-1}{L-1} \quad \text{and} \quad P(A < B) = \frac{L-A}{L-1} \tag{12}$$

The total number of O schemata can be considered as the total number of subsets; thus, for chromosomes of size L, the total number of O schemata is 2^L. Then $E_{A>B}$ can be calculated by using the total number of O schemata minus the number of O schemata with genes in the inversion range. The result is multiplied by the probability for a gene to be selected as $B[P = 1/(A - 1)]$. Thus,

$$E_{A>B} = \frac{1}{A-1}[2^L - (2^{1+A-B} - 1)] \tag{13}$$

$E_{A<B}$ can be calculated in a similar fashion:

$$E_{A<B} = \frac{1}{L-A}[2^L - (2^{1+B-A} - 1)] \tag{14}$$

By replacing B with x, we obtain the expected value of $E_{A>B}$ as

$$\begin{aligned} E_{A>B} &= \sum_{x=1}^{A}\left(\frac{1}{A-1}[2^L - (2^{1+A-x} - 1)]\right) \\ &= \frac{1}{A-1}\left(2^L(A-1) + (A-1) - \sum_{x=2}^{A} 2^x\right) \end{aligned}$$

Likewise, the expected value of $E_{A<B}$ is

$$\begin{aligned} E_{A<B} &= \sum_{x=1}^{1+L-A}\left(\frac{1}{L-A}[2^L - (2^{1+x-A} - 1)]\right) \\ &= \frac{1}{L-A}\left(2^L(L-A) + (L-A) - \sum_{x=2}^{1+L-A} 2^x\right) \end{aligned}$$

The products of the probabilities and expected values are expressed as

$$\begin{aligned} P(A > B)E_{A>B} &= \frac{A-1}{L-1} \\ &\quad \times \frac{1}{A-1}\left(2^L(A-1) + (A-1) - \sum_{x=2}^{A} 2^x\right) \\ P(A < B)E_{A<B} &= \frac{L-A}{L-1} \\ &\quad \times \frac{1}{L-A}\left(2^L(L-A) + (L-A) - \sum_{x=2}^{1+L-A} 2^x\right) \end{aligned}$$

Since $E_A = P(A > B)\, E_{A>B} + P(A < B)\, E_{A<B}$,, then

$$E = \sum_{A=1}^{L} E_A \tag{15}$$

Thus

$$\begin{aligned} E &= \left[\sum_{A=1}^{L}\frac{L}{L-1}\left(2^L(A-1) + (A-1) - \sum_{x=2}^{A} 2^x\right)\right. \\ &\quad \left. + \left(2^L(L-A) + (L-A) - \sum_{x=2}^{1+L-A} 2^x\right)\right] \\ &= \sum_{A=1}^{L}\left(2^L + 1 - \frac{1}{L-1}(2^{A+1} - 4 + 2^{2+L-A} - 4)\right) \\ &= L(2^L+1)\frac{1}{L-1}\left(\sum_{A=1}^{L} 2^{A+1} + \sum_{A=1}^{L} 2^{2+L-A} - 8L\right) \\ &= L(2^L+1)\frac{1}{L-1}(8 \times 2^L - 8 - 8L) \end{aligned}$$

Thus,

$$E = L \times (2^L + 1) - \frac{1}{L-1}(8 \times 2^L - 8 - 8L)$$

and $E_{\text{avg}} = E/L$, so that

$$E_{\text{avg}} = (2^L + 1) + \frac{8}{L-1} + \frac{8}{L(L-1)} - \frac{8 \times 2^L}{L(L-1)}$$

Thus, the survival rate is

$$\text{SR}_{\text{inv}} = \frac{E_{\text{avg}}}{2^L} = 1 + \frac{1}{2^L} + \frac{8}{2^L(L-1)} + \frac{8}{2^L L(L-1)} - \frac{8}{L(L-1)}$$

For a large chromosome, L becomes large. Thus, the survival rate can be simplified to

$$\text{SR}_{\text{inv}} = 1 - \frac{8}{L(L-1)} \tag{16}$$

Neighborhood Inversion. Neighborhood inversion can be considered as a special case of standard inversion. Since only one component will be selected, the selection probabilities become

$$P(A > B) = \tfrac{1}{2} \quad \text{and} \quad P(A < B) = \tfrac{1}{2}$$

The equations (13) and (14) become

$$\begin{aligned} E_{A>B} &= 2^L - (2^2 - 1) \\ E_{A<B} &= 2^L - (2^2 - 1) \end{aligned}$$

Using Eq. (11) results in $E_A = 2^L - (2^2 - 1)$. Thus, the survival rate becomes

$$\text{SR}_{\text{n inv}} = 1 - \frac{3}{2^L} \tag{17}$$

Mutation. Mutation behaves similarly to standard inversion from the expected-value point of view. Thus, the expected value is given by Eq. (11), and the selection probabilities are given by Eq. (12). $E_{A>B}$ can be calculated by using the total number of O schemata minus the number of O schemata that use genes in the inversion range. The result is multiplied by the probability $P = 1/(A - 1)$ for a gene to be selected as B. Thus, $E_{A>B}$ is

$$E_{A>B} = \frac{1}{A-1}[2^L - (2^2 - 1)]$$

In a similar fashion $E_{A<B}$ is calculated using $P = 1/(L - A)$:

$$E_{A<B} = \frac{1}{L-A}[2^L - (2^2 - 1)]$$

The PE products become

$$P(A > B)E_{A>B} = \frac{A-1}{L-1} \times \frac{1}{A-1}[2^L \times (A-1) - 3 \times (A-1)]$$

and

$$P(A < B)E_{A<B} = \frac{L-A}{L-1} \times \frac{1}{L-A}[2^L \times (L-A) - 3 \times (L-A)]$$

Thus, using Eq. (17), E becomes

$$\begin{aligned} E &= \sum_{A=1}^{L} \left(\frac{1}{L-1}[2^L(A-1) - 3(A-1)] \right. \\ &\quad \left. + \frac{1}{L-1}[2^L(L-A) - 3(L-A)] \right) \\ &= \sum_{A=1}^{L} (2^L - 6) \\ &= L \times 2^L - 6L \end{aligned}$$

The average E is

$$E_{\text{avg}} = \frac{E}{L} = 2^L - 6$$

and the survival rate becomes

$$\text{SR}_{\text{mut}} = 1 - \frac{6}{2^L} \tag{18}$$

Analysis Using Holland's Fundamental Theorem. Holland's fundamental theorem is used to investigate the behavior of IENS. The selection method is to choose the fittest chromosome as seed; therefore, the next-generation O-schema number $M(H, t + 1)$ is

$$M(H, t+1) = M(H, t)\frac{f(H)}{f_{\text{avg}}} \tag{19}$$

where H is the O schema, $f(H)$ is the fitness value, and f_{avg} is the population-average fitness value. The ratio of the fitness values ensures that the O-schema number increases. The reproduction is done by means of the mutation operator; thus Eq. (19) becomes

$$M(H, t+1) = M(H, t)\frac{f(H)}{f_{\text{avg}}}\left(1 - \frac{6}{2^L}\right) \tag{20}$$

For the inversion search operation, $M(H, t + 1)$ satisfies the following inequality:

$$\begin{aligned} M(H, t+1) \geq M(H, t)\frac{f(H)}{f_{\text{avg}}}\left(1 - \frac{6}{2^L}\right) \\ \times \left(1 + \frac{1}{2^L} + \frac{8}{2^L(L-1)} + \frac{8}{2^L L(L-1)} \right. \\ \left. - \frac{8}{L(L-1)} - \frac{6P}{2^L}\right) \end{aligned}$$

When L is large, this can be simplified to

$$M(H, t+1) \geq M(H, t)\frac{f(H)}{f_{\text{avg}}}\left(1 - \frac{8}{L(L-1)}\right) \tag{21}$$

For the neighborhood inversion search operation, $M(H, t + 1)$ satisfies the following inequality:

$$M(H, t+1) \geq M(H, t)\frac{f(H)}{f_{\text{avg}}}\left(1 - \frac{6}{2^L}\right)\left(1 - \frac{3}{2^L} - \frac{6P}{2^L}\right)$$

For large L, this is simplified to

$$M(H, t+1) \geq M(H, t)\frac{f(H)}{f_{\text{avg}}} \tag{22}$$

To ensure that M increases as the equilibrium point gets closer, a comparison between $f(H, t)$ at time t and $f(H, t + 1)$ at time $t + 1$ is necessary. The best history of each operation is transferred to the other as seed; this in turn provides a monotonic change of M.

Effect of Operators on Distance

Changes in M depend largely on the selection, as shown earlier. The inversion and mutation operators provide a slow and steady improvement. The effect of these operators on the traveled distance is evaluated in this subsection.

Inversion. For the inversion operator two points must be selected. The probability of randomly selecting two points from a string of length m is $\binom{m}{2}^{-1}$. A uniform distribution is used, and the chromosome is considered to be a ring string rather than a line string. Assuming that A and B are the two inversion points, then $|A - B| + 1$ is the number of genes to be inverted in the string. Having $A = i$ and $B = j$, the original distance and the new distance after inversion are

$$\begin{aligned} \text{Dist}_{\text{orig}} &= d_{1,2} + d_{2,3} + \cdots + d_{i-1,i} + d_{i,i+1} + \cdots + d_{j-1,j} \\ &\quad + d_{j,j+1} + \cdots + d_{m-1,m} + d_{m,1} \\ \text{Dist}_{\text{inv}} &= d_{1,2} + d_{2,3} + \cdots + d_{i-1,j} + d_{j,j-1} + \cdots + d_{i+1,i} \\ &\quad + d_{i,j+1} + \cdots + d_{m-1,m} + d_{m,1} \end{aligned}$$

The new distance Dist_{inv} has been modified in $|A - B| + 1$ genes. However, the overall distance change is equal to $|d_{i-1,i} + d_{jj+1} - d_{i-1,j} - d_{i,j+1}|$. This is because in the considered TSP

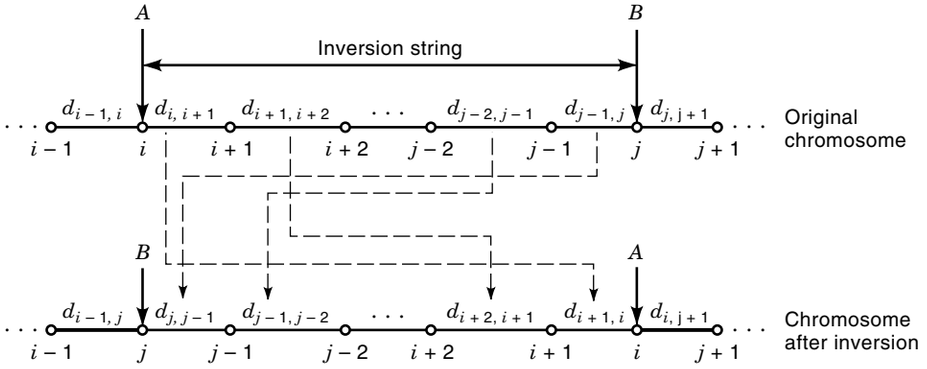


Figure 14. Changes in the chromosome edges after using inversion.

the distance calculation is direction-insensitive, i.e., $d_{i,i+1} = d_{i+1,i}$. It should be pointed out that there are other TSPs where distance is direction-sensitive ($d_{i,i+1} \neq d_{i+1,i}$).

Figure 14 shows the changes in the chromosome edges when the inversion operation is applied. The new links in this chromosome are drawn with thicker lines. The new distance is rewritten as

$$\begin{aligned}\text{Dist}_{\text{inv}} = {} & d_{1,2} + d_{2,3} + \cdots + d_{i-1,j} + d_{i,i+1} + \cdots + d_{j-1,j} \\ & + d_{j,j+1} + \cdots + d_{m-1,m} + d_{m,1}\end{aligned}$$

Thus, Dist_{inv} can be expressed in terms of the original distance as follows:

$$\text{Dist}_{\text{inv}} = \text{Dist}_{\text{orig}} - d_{i-1,i} - d_{j,j+1} + d_{i-1,j} + d_{i,j+1}$$

on replacing i by A and$j$ by B, the distance expression becomes

$$\text{Dist}_{\text{inv}} = \text{Dist}_{\text{orig}} - d_{A-1,A} - d_{B,B+1} + d_{A-1,B} + d_{A,B+1}$$

In order to obtain the upper and lower bounds, distances d_{orig} and d_{inv} need be introduced:

$$\begin{aligned}d_{\text{orig}} &= \max(d_{A-1,A}, d_{B,B+1}) \\ d_{\text{inv}} &= \max(d_{A-1,B}, d_{A,B+1})\end{aligned}$$

Thus Dist_{inv} becomes

$$\text{Dist}_{\text{inv}} = \text{Dist}_{\text{orig}} - 2d_{\text{orig}} + 2d_{\text{inv}}$$

Thus the upper and lower bounds of the distance after inversion become

$$\text{Dist}_{\text{orig}} - 2d_{\text{orig}} \leq \text{Dist}_{\text{inv}} \leq \text{Dist}_{\text{orig}} + 2d_{\text{inv}}$$

Considering a 2-D Euler surface, the maximum distance between two points in a unit square is

$$d_{\max} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = \sqrt{2} \tag{23}$$

Therefore, the bounds for Dist_{inv} are $\text{Dist}_{\text{orig}} - 2\sqrt{2} \leq \text{Dist}_{\text{inv}} \leq \text{Dist}_{\text{orig}} + \sqrt{2}$.

For an r-square surface, the following changes are introduced: x_1 to rx_1, x_2 to rx_2, y_1 to ry_1, and y_2 to ry_2. Consequently, the distance bounds become

$$\text{Dist}_{\text{orig}} - 2r\sqrt{2} \leq \text{Dist}_{\text{inv}} \leq \text{Dist}_{\text{orig}} + 2r\sqrt{2} \tag{24}$$

Mutation. For mutation distance analysis a similar approach and assumptions to those of the inversion analysis can be used. Having $A = i$ and $B = j$, the original distance and the new distance after mutation are

$$\begin{aligned}\text{Dist}_{\text{orig}} = {} & d_{1,2} + d_{2,3} + \cdots + d_{i-1,i} + d_{i,i+1} + \cdots + d_{j-1,j} \\ & + d_{j,j+1} + \cdots + d_{m-1,m} + d_{m,1} \\ \text{Dist}_{\text{mut}} = {} & d_{1,2} + d_{2,3} + \cdots + d_{i-1,j} + d_{j,i+1} + \cdots + d_{j-1,i} \\ & + d_{i,j+1} + \cdots + d_{m-1,m} + d_{m,1}\end{aligned}$$

The new distance has been modified only on two genes. However, the overall distance change is equal to $|d_{i-1,i} + d_{i,i+1} +$

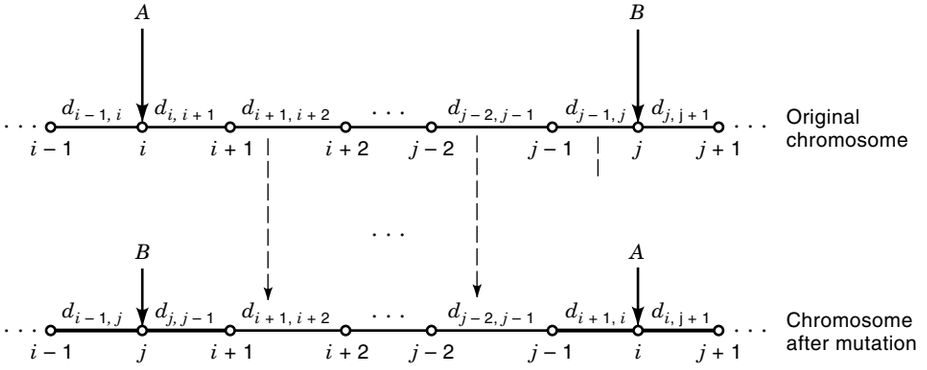


Figure 15. Changes in the chromosome edges after using mutation.

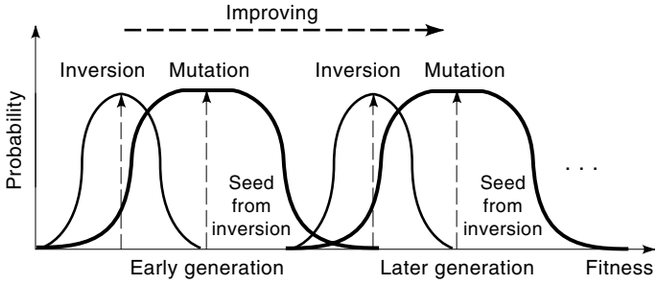


Figure 16. Monotonic behavior of the proposed IENS genetic algorithm.

$d_{j-1,j} + d_{j,j+1} - d_{i-1,j} - d_{j,i+1} - d_{j-1,i} - d_{i,j+1}|$. This is because a mutation operation changes four edges.

Figure 15 shows changes in the chromosome edges when the mutation is applied. This figure highlights the affected links. On replacing i by A and j by B, the distance expression becomes

$$\begin{aligned}\text{Dist}_{\text{mut}} = \text{Dist}_{\text{orig}} &- d_{A-1,A} - d_{A,A+1} - d_{B-1,B} - d_{B,B+1} \\ &+ d_{A-1,B} + d_{B,A+1} + d_{B-1,A} + d_{A,B+1}\end{aligned}$$

In order to obtain the upper and lower bounds, two distances (d_{orig} and d_{mut}) are introduced:

$$\begin{aligned}d_{\text{orig}} &= \max(d_{A-1,A}, d_{A,A+1}, d_{B-1,B}, d_{B,B+1}) \\ d_{\text{mut}} &= \max(d_{A-1,B}, d_{B,A+1}, d_{B-1,A}, d_{A,B+1})\end{aligned}$$

Thus Dist_{mut} becomes

$$\text{Dist}_{\text{mut}} = \text{Dist}_{\text{orig}} - 4d_{\text{orig}} + 4d_{\text{mut}}$$

Thus the upper and lower bounds of the distance after mutation become

$$\text{Dist}_{\text{orig}} - 4d_{\text{orig}} \le \text{Dist}_{\text{mut}} \le \text{Dist}_{\text{orig}} + 4d_{\text{mut}}$$

Considering a 2-D Euler unit square and the distance $d_{\max}$ in Eq. (23), from Eq. (24) the bounds for Dist_{mut} are $\text{Dist}_{\text{orig}} - 4\sqrt{2} \le \text{Dist}_{\text{mut}} \le \text{Dist}_{\text{orig}} + 4\sqrt{2}$. For an r-square surface, this becomes

$$\text{Dist}_{\text{orig}} - 4r\sqrt{2} \le \text{Dist}_{\text{mut}} \le \text{Dist}_{\text{orig}} + 4r\sqrt{2} \qquad (25)$$

Behavior

It is interesting to observe how the IENS approach reaches the solution. Using Eqs. (20) to (25), it is possible to illustrate the behavior of IENS as function of fitness.

The monotonic behavior of IENS is shown in Fig. 16. It can be observed that the inversion and mutation operators drive the chromosome population towards better fitness values.

Since the best chromosome after inversion is preserved as the seed for reproduction (using mutation), the distance is driven toward better fitness. Each of the reproduced chromosome can obtain a fitness value that is as much as twice its parent's fitness value. The reproduced chromosomes are then used by the inversion operator.

The inversion and neighborhood inversion searches have similar effects on the distance. Thus, these two operators have comparable distributions. The best-history chromosome is kept as the starting seed for each search operator. From the fundamental theory analysis for IENS and the distance analysis, both the ordering survival rate and number of O schemata can be estimated as well as how much the distance is improved. IENS preserves the order and at the same time explores the search solution space in a monotonic fashion.

Performance

The IENS approach has been applied to a number of TSP benchmarks. These benchmarks include 10-city problems (25), 30-city problems (16), 50- and 75-city problems (26), and 105- and 318-city problems (11).

For the 10-city TSP, a set of five benchmarks, fully described in Lin et al. (10,25), have been used. The hybrid genetic algorithm has been used to find solutions for these benchmarks. A summary of the results is shown in Table 2. The optimal solutions are included. The ceiling number of generations is set at 200. The threshold number of generations is 5. For each benchmark 100 runs were performed. From Table 2, it can be observed that:

- The IENS algorithm obtains the optimal solution for all the 10-city TSPs in every run. The distance is that of the best solution, i.e., at least one chromosome per run represents the optimal solution.
- The number of generations to find the optimal solutions in the best case is extremely small. It ranges from 3 to 7.
- The average number of generations to find the optimal solution is fairly small. With exception of benchmark 10.4, it is below 30.

Other benchmarks for 30- to 318-city TSPs were used to evaluate the proposed IENS approach. The results of the simulations are shown in Table 3. For comparison, the results are reported using both the integer and real-number solutions. The integer solutions are obtained by adding the roundoff distance between any two cities in the traveled path. The percentage divergence rate gives a measurement of the difference

Table 2. Simulation Results for 10-City TSP Benchmarks

Benchmark:	10.1	10.2	10.3	10.4	10.5
Optimal distance:	2.986918	3.52247	2.82804	2.88463	2.9262
IENS best solution:	2.986918	3.52247	2.82804	2.88463	2.9262
IENS best-solution generation:	5	3	6	7	4
IENS average distance:	2.986918	3.52247	2.82804	2.88463	2.9262
IENS average generations:	21.58	20.38	23.94	69.47	26.04

Table 3. Results for 30- to 318-City TSP Problems

Benchmark:	30	50	75	105	318
Best known solution:	420 (integer)	425 (integer)	535 (integer)	14382.9	41345 (integer)
IENS best solution (real):	423.740	427.855	542.309	14382.995	43105.6048
IENS best solution (int.):	420	425	535	—	43020
IENS best-soln. gen.:	262	8938	48353	25794	942052
IENS average distance:	425.75	433.821	550.849	14742.736	43710.278
IENS avg. generations:	2027.78	5899.68	50807.88	45287.16	647305.01
IENS ceiling generations:	5000	10,000	100,000	100,000	1000,000
IENS runs:	50	50	50	30	18
IENS divergence rate (%):	0.47	1.39	1.57	2.5	5.4

between the average distance and the best known solution. The percentage divergence rate is computed as follows:

$$\text{divergence rate} = \frac{\text{average distance} - \text{best known distance}}{\text{best known distance}} \times 100\% \tag{26}$$

It can be observed that:

- IENS obtains the best known solution for all the benchmarks with the exception of the 318-city benchmark.
- All the results are always extremely close to the best known solutions. Average divergence rates vary from 0.47% to 5.4% as for 30- to 318-city benchmarks.
- The average number of generations increases fast from the 30- to the 318-city benchmark. IENS requires more search time for large problems. However, due to its simplicity, IENS operates fast in each generation.

Remarks

The hybrid genetic algorithm called *inversion with embedded Newton–Raphson search* (IENS), which combines the Newton–Raphson numerical method and a genetic algorithm, has been introduced. This algorithm is an example of hybrid genetic algorithms for the TSP. The operations and their analysis can be used for other TSP genetic approaches. The IENS approach has the following characteristics:

- Simple and inexpensive operation
- Monotonic improvement of the tour
- Optimal solution for all the 10-city TSPs at every run. The distance is identical to that of the optimal solution.
- Best known solution for all the TSP benchmarks except the 318-city benchmark. It can be observed (in Table 3) that the solutions are always extremely close to the optimal solutions. The average distance increases from 0.47% to 5.4% as one goes from 30- to 318-city benchmarks.

This example shows that genetic algorithms have a potential for obtaining extremely good solutions to optimization problems such as the TSP.

BIBLIOGRAPHY

1. C. H. Papadimitriou and K. Steiflitz, *Combinatorial Optimization: Algorithms and Complexity,* Englewood Cliffs, NJ: Prentice-Hall, 1982.
2. D. J. Wilde and C. S. Beightler, *Foundations of Optimization,* Englewood Cliffs, NJ: Prentice-Hall, 1967.
3. D. A. Pierre, *Optimization Theory with Applications,* New York: Wiley, 1969.
4. A. Torn and A. Zilinxkas, *Global Optimization,* New York: Springer-Verlag, 1987.
5. E. L. Lawler et al., *The Traveling Salesman Problem,* New York: Wiley, 1985.
6. J. J. Hopfield and D. W. Tank, Neural computation of decisions in optimization problems, *Biol. Cybern.,* **52**: 141–152, 1985.
7. S. V. B. Aiyer, Solving combinatorial optimization problems using neural networks with applications in speech recognition, Ph.D. dissertation, Cambridge University, England, 1991.
8. E. Sanchez-Sinencio and C. Lau, *Artificial Neural Networks Paradigms, Applications, and Hardware Implementations,* New York: IEEE Press, 1992.
9. P. W. Pretzel, D. L. Palumbo, and M. K. Arras, Performance and fault-tolerance of neural networks for optimization, *IEEE Trans. Neural Netw.,* **4**: 600–614, 1993.
10. W. Lin, High quality tour hybrid genetic schemes for TSP optimization problems, Ph.D. dissertation, State University of New York at Binghamton, 1995.
11. S. Lin and B. W. Kernighan, An effective heuristic algorithm for the traveling salesman problem, *Oper. Res.,* **21**: 498–516, 1976.
12. D. B. Fogel and L. J. Fogel, Evolutionary computation, *IEEE Trans. Neural Netw.,* **5**: 1, 1994.
13. D. B. Fogel, An introduction to simulated evolutionary optimization, *IEEE Trans. Neural Netw.,* **5**: 3–14, 1994.
14. M. Srinivas and L. M. Patnaik, Genetic algorithms: A survey, *IEEE Comput.,* **27**: 17–26, 1994.
15. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning,* Reading, MA: Addison-Wesley, 1989.
16. I. M. Oliver, D. J. Smith, and J. R. C. Holland, A study of permutation crossover operators on the traveling salesman problem, *Proc. 2nd Int. Conf. Genet. Algorithms: Genet. Algorithms Appl.,* Cambridge, MA, 1987, pp. 224–230.
17. D. Whitley, T. Starkweather, and Q. Fuquay, Scheduling problems and traveling salesman: The genetic edge recombination operator, *Proc. 3rd Int. Conf. Genet. Algorithms: Genet. Algorithms Appl.,* Arlington, VA, 1989, pp. 133–140.
18. A. Homaifar, S. Guan, and G. E. Liepins, A new approach on the traveling salesman problem by genetic algorithms, *Genet. Algorithms Appl.: Proc. 5th Int. Conf. Genet. Algorithms,* 1993, pp. 460–466.
19. D. B. Fogel, Applying evolutionary programming to selected traveling salesman problems, *Cybern. Syst.: Int. J.,* **24**: 27–36, 1994.
20. X. Qi and F. Palmieri, Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. Part

I: Basic properties of selection and mutation, *IEEE Trans. Neural Netw.,* **5**: 102–119, 1994.

21. X. Qi and F. Palmieri, Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. Part II: Analysis of diversification role of crossover, *IEEE Trans. Neural Netw.,* **5**: 120–129, 1994.
22. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs,* Berlin: Springer-Verlag, 1992.
23. W. H. Press et al., *Numerical Recipes,* Cambridge, England: Cambridge University Press, 1986.
24. D. J. Sirag and P. T. Weisser, Toward a unified thermodynamic genetic operator, *Genet. Algorithms Appl.: Proc. 2nd Int. Conf. Genet. Algorithms,* 1987, pp. 116–122.
25. W. Lin et al., An evaluation of energy function for a neural network model for optimization problems, *IEEE Int. Jt. Conf. Neural Netw.,* Orlando, FL, 1994, pp. 4518–4522.
26. S. Eilon and N. Christofides, Distribution management: Mathematical modeling and practical analysis, *Oper. Res. Q.,* **20**: 309, 1969.

WEI LIN
Coopers & Lybrand LLP
JOSÉ G. DELGADO-FRIAS
DONALD C. GAUSE
State University of New York at Binghamton

TRAVELING WAVE AMPLIFIERS. See DISTRIBUTED AMPLIFIERS; TRAVELING WAVE TUBES.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Vectors
Standard Article
C. D. Cantrell[1]
[1]University of Texas at Dallas, Richardson, TX
Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.
: 10.1002/047134608X.W2464
Article Online Posting Date: December 27, 1999
Abstract | Full Text: HTML PDF (174K)

Abstract

The sections in this article are

Vector Components in Orthogonal Curvilinear Coordinate Systems

Vector Integration

Vector Operators

| | |

Copyright © 1999-2008 All Rights Reserved.

VECTORS

A *vector* is a physical quantity that has the attributes of magnitude and direction, such as electric field intensity ($\boldsymbol{E}$), magnetic field intensity ($\boldsymbol{H}$), and current density ($\boldsymbol{J}$). The magnitude of a vector is always a nonnegative real number (such as 0, 2 or π). The *null vector* (or zero vector) $\boldsymbol{0}$ has a magnitude of zero, and has no direction. An ordinary number is called a *scalar*. A *unit vector* is a vector with a magnitude of 1 and any direction. Unit vectors are used to specify directions. Figure 1 shows several vectors of different magnitudes and directions.

The concept of a vector is a generalization of the concept of a displacement in space (such as "go 300 m east" or "move the capacitor 5 mm toward the connector end of the printed-circuit board"). Vectors that lie entirely in a single plane are called *two-dimensional* vectors, in contrast to *three-dimensional* vectors, which can point in any direction in three-dimensional space. A two- or three-dimensional vector is represented with an arrow, the length of which is proportional to the length of the vector, and the direction of which is the same as the direction of the vector. Abstract vectors with more than three dimensions are useful for computational purposes; see below.

The standard symbolic notations for vectors are boldface letters ($\boldsymbol{a}$) and lightface letters with an arrow overhead ($\vec{a}$).

Physically meaningful vectors have units. For example, the units of the electric field intensity vector $\boldsymbol{E}$ are volts per meter. The units of the vector are the same as the units of its magnitude. The set of all vectors with the same units is a *vector space*. Units are also called dimensions, but the term "dimension" has a different meaning in vector spaces.

Physical vectors such as position and momentum are properties of an object. The position vector of a point in space, $\boldsymbol{r}$, is defined by the object's x, y, and z coordinates: $\boldsymbol{r} = (x, y, z)$. However, electrical engineering also makes use of vector fields, such as the electric field, $\boldsymbol{E}$, the magnetic field, $\boldsymbol{H}$, and the current density field, $\boldsymbol{J}$. In general, a *vector field* is a function, for which we use the generic symbol $\boldsymbol{F}$, that assigns a vector $\boldsymbol{F}(\boldsymbol{r})$ to each point in space.

If a vector $\boldsymbol{a}$ with a magnitude a is multiplied by a number α, the result is a vector

$$\boldsymbol{b} = \alpha \boldsymbol{a}$$

in the same direction as $\boldsymbol{a}$ with a magnitude equal to αa. In mathematical language, a vector space is closed under *scalar multiplication.*

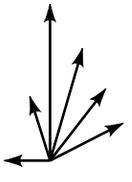


Figure 1. Several vectors of different magnitudes and directions.

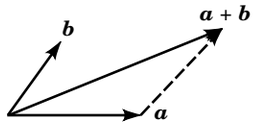


Figure 2. Head-to-tail method for combining successive displacements.

The rule for adding two vectors follows directly from the common-sense head-to-tail method for combining successive displacements, as shown in Fig. 2. The sum of two vectors is another vector; in mathematical terms, a vector space is closed under vector addition. Vector addition is *commutative,* that is, the sum of two vectors does not depend on the order in which the vectors are added:

$$\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{b} + \boldsymbol{a}$$

Vector addition is also *associative,* meaning that the sum of three or more vectors does not depend on the manner in which the vectors are grouped:

$$\boldsymbol{a} + (\boldsymbol{b} + \boldsymbol{c}) = (\boldsymbol{a} + \boldsymbol{b}) + \boldsymbol{c}$$

To compute the sum of two vectors, it often is convenient to express the vectors in terms of *components* along coordinate axes. The *Cartesian components* of a vector $\boldsymbol{a}$ are its components along the Cartesian x and y coordinate axes (and the z axis, in three dimensions). The component of a vector $\boldsymbol{a}$ along the x axis (for example) is usually written a_x. The x and y components of a two-dimensional vector are shown in Fig. 3, which also illustrates the important theorem that a vector is equal to the vector sum of its projections on the coordinate axes:

$$\boldsymbol{a} = a_x \hat{\boldsymbol{x}} + a_y \hat{\boldsymbol{y}}$$

From the Pythagorean theorem and from the fact that a_x and a_y can be thought of as distances along the x and y axes, which are mutually perpendicular, one sees that $a^2 = a_x^2 + a_y^2$. This gives the important formula

$$a = \sqrt{a_x^2 + a_y^2}$$

for the magnitude of a two-dimensional vector in terms of the magnitudes of the components along mutually perpendicular axes.

A two-dimensional vector $\boldsymbol{a}$ is the displacement vector from the origin to the point whose Cartesian coordinates are (a_x, a_y). For this reason a complex number (such as an impedance, or a phasor representing an alternating voltage or current) can be represented by a two-dimensional vector. For ex-

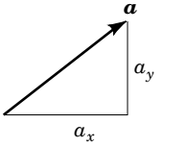


Figure 3. The x and y components of a 2-D vector.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

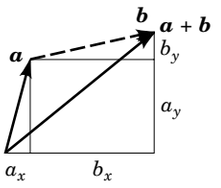


Figure 4. Illustration of components where sum of two or more vectors are equal to sums of components of the vectors.

ample, an impedance $\boldsymbol{Z} = 21 + j28\ \Omega$ can be represented by a two-dimensional vector with the Cartesian components (21, 28). The magnitude of the impedance in this example is $|\boldsymbol{Z}| = \sqrt{21^2 + 28^2} = 35\ \Omega$.

In three dimensions, the component resolution of a vector $\boldsymbol{a}$ is

$$\boldsymbol{a} = a_x\hat{\boldsymbol{x}} + a_y\hat{\boldsymbol{y}} + a_x\hat{\boldsymbol{z}}$$

and the formula for the vector's magnitude is

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

in terms of the Cartesian components a_x, a_y, and a_z.

Figure 4 illustrates the fact that the components of the sum of two or more vectors are equal to the sums of the components of the vectors:

$$\boldsymbol{a} + \boldsymbol{b} = (a_x\hat{\boldsymbol{x}} + a_y\hat{\boldsymbol{y}}) + (b_x\hat{\boldsymbol{x}} + b_y\hat{\boldsymbol{y}}) = (a_x + b_x)\hat{\boldsymbol{x}} + (a_y + b_y)\hat{\boldsymbol{y}}$$

In other words, the x component of $\boldsymbol{a} + \boldsymbol{b}$ is $a_x + b_x$.

The effective impedance of a combination of impedances in series can be found by vector addition. For example, if an impedance $\boldsymbol{Z}_1 = 1 + j3\ \Omega$ is in series with an impedance $\boldsymbol{Z}_2 = 5 - j4\ \Omega$, the effective impedance can be found by calculating the vector sum

$$\boldsymbol{Z} = \boldsymbol{Z}_1 + \boldsymbol{Z}_2 = (1, 3) + (5, -4) = (6, -1)$$

from which one sees that the effective impedance is $\boldsymbol{Z} = 6 - j1\ \Omega$. Figure 5 shows a graphical solution to this example.

To specify the direction of a two-dimensional vector numerically one uses the angle θ between the direction of the vector and a reference line, which usually is chosen as the positive x axis. For example, an impedance $\boldsymbol{Z} = 6 - j1\ \Omega$ (expressed in terms of its resistive and reactive components) can also be represented in terms of its magnitude and phase angle as $\boldsymbol{Z} = 6.08\angle -9.46°$ (Fig. 5).

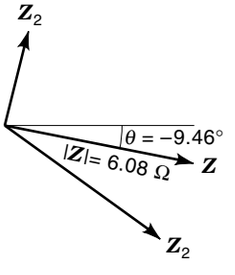


Figure 5. Graphical solution to $Z = 6 - j1\ \Omega$.

Given the x and y components of a two-dimensional vector, one can determine the angle θ that the vector makes with the x axis from the formula

$$\tan\theta = \frac{a_y}{a_x}$$

When using this formula numerically, one should remember that the arctangent function returns angles modulo 180°, that is, $\tan^{-1}(a_y/a_x)$ lies in the range (−90°, 90°), not in the range (0°, 360°) or (−180°, 180°). Therefore, one should determine the quadrant in which the angle lies by examining a_x and a_y, and adjust the value of the angle returned by the arctangent function by 180° or 360° if necessary. For example, if $a_x < 0$ and $a_y > 0$, then θ lies in the second quadrant, and therefore

$$\theta = \tan^{-1}\left(\frac{a_y}{a_x}\right) + 180°$$

in this case.

There are two meaningful products of two- or three-dimensional vectors. The *scalar product* (also called the dot product or inner product), which is written $\boldsymbol{a} \cdot \boldsymbol{b}$, is the number

$$\boldsymbol{a} \cdot \boldsymbol{b} = ab\cos\theta_{ab}$$

where θ_{ab} is the angle between $\boldsymbol{a}$ and $\boldsymbol{b}$ (measured so that θ_{ab} is not less than 0° and is not greater than 180°). If $\hat{\boldsymbol{n}}$ is a unit vector, then $\hat{\boldsymbol{n}} \cdot \boldsymbol{a} = a\ \cos\ \theta_{na}$ is the component of $\boldsymbol{a}$ along the direction specified by $\hat{\boldsymbol{n}}$ (see Fig. 6).

Geometrically, $\boldsymbol{a} \cdot \boldsymbol{b}$ is equal to the magnitude of $\boldsymbol{a}$ times the magnitude of the component of $\boldsymbol{b}$ along the direction of $\boldsymbol{a}$. Trigonometric manipulation shows that the scalar product is equal to the sum of the products of the Cartesian components:

$$\boldsymbol{a} \cdot \boldsymbol{b} = a_x b_x + a_y b_y + a_z b_z$$

If $\boldsymbol{a}$ and $\boldsymbol{b}$ are perpendicular to each other, then their scalar product vanishes:

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{0}$$

and one says that $\boldsymbol{a}$ and $\boldsymbol{b}$ are *orthogonal.* One of the most important applications of the dot product is the computation of vector components. For example, to compute the rectangular components of a vector $\boldsymbol{a}$, one takes the dot product of $\boldsymbol{a}$ with the Cartesian unit vectors:

$$a_x = \hat{\boldsymbol{x}} \cdot \boldsymbol{a} \text{ and } a_y = \hat{\boldsymbol{y}} \cdot \boldsymbol{a}$$

The *vector product* (also called the cross product) of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is the vector $\boldsymbol{a} \times \boldsymbol{b}$ with the magnitude $ab|\sin\ \theta_{ab}|$,

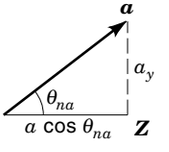


Figure 6. $\hat{\boldsymbol{n}} \cdot \boldsymbol{a} = a\ \cos\ \theta_{na}$ is the component of $\boldsymbol{a}$ along the direction specified by $\hat{\boldsymbol{n}}$.

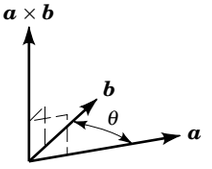


Figure 7. Right-hand rule.

orthogonal to both $\boldsymbol{a}$ and $\boldsymbol{b}$, and with the sense that one obtains by rotating $\boldsymbol{a}$ into $\boldsymbol{b}$ and using the right-hand rule (Fig. 7). To apply the right-hand rule, let the fingers of the right hand point from $\boldsymbol{a}$ to $\boldsymbol{b}$. The thumb then points in the direction of $\boldsymbol{a} \times \boldsymbol{b}$.

The magnitude of the vector product $\boldsymbol{a} \times \boldsymbol{b}$ is equal to the area of the parallelogram defined by $\boldsymbol{a}$ and $\boldsymbol{b}$. Referring to Fig. 8, one sees that the base of the parallelogram is the length of the vector $\boldsymbol{a}$, while the height is the length of $\boldsymbol{b}$ times the sine of the angle between $\boldsymbol{a}$ and $\boldsymbol{b}$. Hence the area of the parallelogram is equal to $ab\,|\sin\theta_{ab}|$, which is the magnitude of $\boldsymbol{a} \times \boldsymbol{b}$. Often one refers to $\boldsymbol{a} \times \boldsymbol{b}$ as the *directed area* of the parallelogram, because $\boldsymbol{a} \times \boldsymbol{b}$ has both magnitude and direction.

More generally, a directed area is any polygon, together with one of the two possible directions in space perpendicular to the polygon.

The *triple scalar product* $\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c})$ is unchanged under cyclic permutations of $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, and changes sign under permutations that change the cyclic order:

$$\begin{aligned}\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) &= \boldsymbol{b} \cdot (\boldsymbol{c} \times \boldsymbol{a}) = \boldsymbol{c} \cdot (\boldsymbol{a} \times \boldsymbol{b}) \\ &= -\boldsymbol{a} \cdot (\boldsymbol{c} \times \boldsymbol{b}) - \boldsymbol{b} \cdot (\boldsymbol{a} \times \boldsymbol{c}) = -\boldsymbol{c} \cdot (\boldsymbol{b} \times \boldsymbol{a})\end{aligned}$$

The triple scalar product is equal to the volume of the parallelepiped whose edges are defined by the vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, because the directed area of the base of the parallelepiped is $\boldsymbol{a} \times \boldsymbol{b}$, and the height of the parallelepiped is the component of $\boldsymbol{c}$ that is perpendicular to $\boldsymbol{a} \times \boldsymbol{b}$.

The value of the *triple vector product* $\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c})$ is given by the *BAC–CAB* rule

$$\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{b}(\boldsymbol{a} \cdot \boldsymbol{c}) - \boldsymbol{c}(\boldsymbol{a} \cdot \boldsymbol{b})$$

This formula is useful in electromagnetics and other areas in which one must express a vector $\boldsymbol{b}$ in terms of components that are parallel or perpendicular to a given unit vector $\hat{\boldsymbol{n}}$. For example, if one sets $\boldsymbol{a} = \boldsymbol{c} = \hat{\boldsymbol{n}}$ in the BAC–CAB rule, then one sees that

$$\boldsymbol{b} = (\hat{\boldsymbol{n}} \cdot \boldsymbol{b})\hat{\boldsymbol{n}} + \hat{\boldsymbol{n}} \times (\boldsymbol{b} \times \hat{\boldsymbol{n}})$$

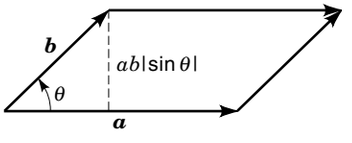


Figure 8. Base of parallelogram is length of vector $\boldsymbol{a}$; height is length of $\boldsymbol{b}$ times the sine of the angle between $\boldsymbol{a}$ and $\boldsymbol{b}$.

where $\boldsymbol{b}$ is any vector. In this formula, the vector $(\hat{\boldsymbol{n}} \cdot \boldsymbol{b})\hat{\boldsymbol{n}}$ is parallel to $\hat{\boldsymbol{n}}$, and the vector $\hat{\boldsymbol{n}} \times (\boldsymbol{b} \times \hat{\boldsymbol{n}})$ is perpendicular to $\hat{\boldsymbol{n}}$.

For computational purposes, an *n-dimensional vector* is any array $v[\ ]$ with n elements $v[k] = v_k$, where $k = 0, \ldots, n - 1$. For example, a numerical solution of a differential equation such as

$$LC\frac{d^2v}{dt^2} + RC\frac{dv}{dt} + v = 0$$

is a vector, the elements $v[k] = v_k = v(t_k)$ of which are the approximate numerical values $v(t_k)$ of the function v at n sampling times $t_0, \ldots, t_{n-1}$. Vectors of time-sampled values are examples of abstract vectors, that is, objects that do belong to three-dimensional space, but that have many of the properties of physical vectors.

An *abstract vector space* V is a set, the elements $\bar{x}, \bar{y}, \bar{z}, \ldots$ of which obey the following axioms, where $\alpha, \beta, \ldots$ are scalars:

1. An operation of addition is defined, such that, for every pair of vectors $\bar{x}, \bar{y}$ in V, the vector sum $\bar{x} + \bar{y}$ belongs to V, and $\bar{y} + \bar{x} = \bar{x} + \bar{y}$.
2. Vector addition is associative: $\bar{x} + (\bar{y} + \bar{z}) = (\bar{x} + \bar{y}) + \bar{z}$.
3. A null vector $\bar{0}$ exists, with the property that for every vector $\bar{x}$ in V, $\bar{x} + \bar{0} = \bar{x}$..
4. For every vector $\bar{x}$ in V, there exists a vector $-\bar{x}$ such that $\bar{x} + (-\bar{x}) = \bar{x} - \bar{x} = \bar{0}$.
5. For every vector $\bar{x}$ in V, and for every scalar α, there is a scalar multiple $\alpha\bar{x}$ that belongs to V.
6. Scalar multiplication obeys the distributive laws $\alpha(\bar{x} + \bar{y}) = \alpha\bar{x} + \alpha\bar{y}$ and $(\alpha + \beta)\bar{x} = \alpha\bar{x} + \beta\bar{x}$.
7. Scalar multiplication is associative, meaning that $\alpha(\beta\bar{x}) = (\alpha\beta)\bar{x}$.
8. Multiplying any vector $\bar{x}$ by the zero scalar 0 gives the null vector $\bar{0}$.
9. Multiplying any vector $\bar{x}$ by the unit scalar 1 gives back $\bar{x}$: $1\bar{x} = \bar{x}$.

The concept of an abstract vector space is very widely applicable. The "vectors" $\bar{x}$ can be functions, column vectors, row vectors, or even matrices. Any property that is true for abstract vectors is true for functions, column vectors, and so on.

A set of abstract vectors $\bar{x}_0, \bar{x}_1, \ldots, \bar{x}_{m-1}$ is called *linearly independent* if the only scalars $\alpha_0, \alpha_1, \ldots, \alpha_{m-1}$ such that

$$\sum_{k=0}^{m-1} \alpha_k \bar{x}_k = \bar{0}$$

are $\alpha_0 = \alpha_1 = \cdots = \alpha_{m-1} = 0$. On the other hand, if this equation holds with some nonzero values of the scalars $\alpha_0, \alpha_1, \ldots, \alpha_{m-1}$, then the vectors $\bar{x}_0, \bar{x}_1, \ldots, \bar{x}_{m-1}$ are called *linearly dependent.* In particular, if two abstract vectors $\bar{x}, \bar{y}$ are linearly dependent, then there is a number α such that $\bar{y} = \alpha\bar{x}$. For example, the functions $\sin \omega t$ and $\cos \omega t$ are linearly independent, for there is no number α such that $\cos \omega t = \alpha \sin \omega t$.

It can be proved that, for any given abstract vector space V, the number of linearly independent vectors in a set $\bar{x}_0, \bar{x}_1,$

. . ., x_{m-1} is less than or equal to a certain maximum number n, which is called the *dimension* of the vector space. In effect, the dimension of a vector space is equal to the number of degrees of freedom in the space. For example, there are at most three linearly independent vectors in any set of three-dimensional vectors, because every three-dimensional vector is linearly dependent on the Cartesian unit vectors $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$. A set of linearly independent vectors $\overline{e}_0$, $\overline{e}_1$, . . ., e_{n-1} that has the property that every other vector in a vector space V can be expressed as a linear combination

$$\overline{v} = \sum_{k=0}^{m-1} v_k \overline{e}_k$$

is called a basis of V. If the basis vectors are normalized so that each one has unit length and they are mutually orthogonal, then the basis is called *orthonormal.* The Cartesian unit vectors $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$ are an orthonormal basis of the space of three-dimensional vectors. Orthonormal bases of vector spaces whose elements are functions are used in many applications, ranging from electromagnetic boundary-value problems to the theory of random processes.

VECTOR COMPONENTS IN ORTHOGONAL CURVILINEAR COORDINATE SYSTEMS

Some electromagnetic engineering problems, such as scattering, radiation by an antenna, and wave propagation in cylindrical waveguides, take their simplest form when expressed in coordinates that are adapted to the geometrical symmetry of the problem. For example, an optical fiber can be idealized as a cylinder. The natural coordinate system in this case is the cylindrical system. The *circular cylindrical coordinates* of a point $\boldsymbol{r} = (x, y, z)$ in three-dimensional space are (ρ, ϕ, z), where

$$\rho = \sqrt{x^2 + y^2}$$

is the perpendicular distance from the z axis to the point $\mathbf{r}$,

$$\phi = \tan^{-1}\left(\frac{y}{x}\right)$$

is the azimuthal angle, measured from the X axis to the line in the XY plane that goes from the origin to the point $(x, y, 0)$, and z is equal to the Z Cartesian coordinate of $\boldsymbol{r}$, as is shown in Fig. 9. Evidently cylindrical coordinates are simply

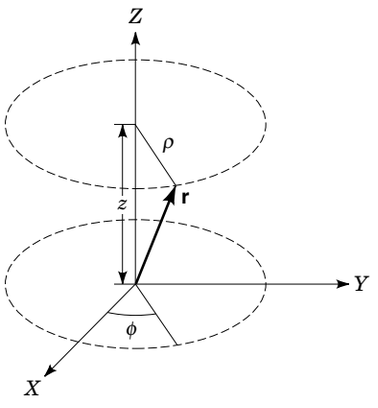


Figure 9. Circular cylindrical coordinates of point $\mathbf{r}$.

plane polar coordinates (ρ, ϕ), with a Z coordinate added to describe three-dimensional objects. The other important orthogonal curvilinear coordinates for engineering applications are spherical polar coordinates, which we discuss below.

One of the natural unit vectors in circular cylindrical coordinates is $\hat{\boldsymbol{z}}$, which points in the direction that one takes if one increases z while holding x and y (or ρ and ϕ) constant. Another is the unit vector $\hat{\boldsymbol{\rho}}$ that points in the direction in which one moves if one increases ρ while holding z and ϕ constant. Still another is the unit vector $\hat{\boldsymbol{\phi}}$ that points in the direction of increasing ϕ (with z and ρ held constant). In terms of the Cartesian unit vectors $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$, the circular cylindrical unit vectors in the XY plane (which are the same as the plane polar unit vectors) are

$$\hat{\boldsymbol{\rho}} = \hat{\boldsymbol{x}} \cos\phi + \hat{\boldsymbol{y}} \sin\phi$$

and

$$\hat{\boldsymbol{\phi}} = -\hat{\boldsymbol{x}} \sin\phi + \hat{\boldsymbol{y}} \cos\phi$$

Because $\hat{\boldsymbol{z}}$ is orthogonal to $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$,

$$\hat{\boldsymbol{z}} \cdot \hat{\boldsymbol{\rho}} = 0 \text{ and } \hat{\boldsymbol{z}} \cdot \hat{\boldsymbol{\phi}} = 0$$

Substituting for $\hat{\boldsymbol{\rho}}$ and $\hat{\boldsymbol{\phi}}$ from the formulas just above, one finds that

$$\hat{\boldsymbol{\phi}} \cdot \hat{\boldsymbol{\rho}} = 0$$

These equations demonstrate that $\hat{\boldsymbol{\rho}}$, $\hat{\boldsymbol{\phi}}$, and $\hat{\boldsymbol{z}}$ form an orthonormal system of unit vectors, which one can use (instead of the fixed rectangular unit vectors $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$) to express any vector in terms of its ρ, ϕ, and z components. The unit vectors $\hat{\boldsymbol{\rho}}$, $\hat{\boldsymbol{\phi}}$, and $\hat{\boldsymbol{z}}$ define a local Cartesian coordinate system, the axes of which point in different directions, depending on the location of the point $\boldsymbol{r}$. Because the unit vectors are mutually orthogonal, circular cylindrical coordinates are an example of orthogonal curvilinear coordinates.

Any vector field $\boldsymbol{F}$ can be expressed in terms of its components along $\hat{\boldsymbol{\rho}}$, $\hat{\boldsymbol{\phi}}$, and $\hat{\boldsymbol{z}}$ as follows:

$$\mathbf{F} = F_\rho \hat{\boldsymbol{\rho}} + F_\phi \hat{\boldsymbol{\phi}} + F_z \hat{\boldsymbol{z}}$$

To compute the circular cylindrical components of $\boldsymbol{F}$, one takes the dot product with the circular cylindrical unit vectors, obtaining

$$F_\rho = \hat{\boldsymbol{\rho}} \cdot \boldsymbol{F} = F_x \cos\phi + F_y \sin\phi$$

and

$$F_\phi = \hat{\boldsymbol{\phi}} \cdot \boldsymbol{F} = -F_x \sin\phi + F_y \cos\phi$$

These formulas are used to compute the circular cylindrical components, given the rectangular components.

The formulas that give the rectangular (XYZ) unit vectors in terms of the circular cylindrical (or plane polar) unit vectors are

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{\rho}} \cos\phi - \hat{\boldsymbol{\phi}} \sin\phi$$

and

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{\rho}} \sin\phi + \hat{\boldsymbol{\phi}} \cos\phi$$

The unit vector $\hat{\boldsymbol{z}}$ is the same in both coordinate systems. These formulas can be used to compute the rectangular components, given the circular cylindrical components.

In general, if a global coordinate system is not Cartesian, it would be convenient to be able to "attach" a local Cartesian coordinate system at each point $\boldsymbol{r}$. Curvilinear coordinate systems in which this is possible are called *orthogonal curvilinear coordinates*. Examples besides circular cylindrical and spherical polar coordinates include elliptic and parabolic coordinates in two dimensions and, in three dimensions, elliptic cylinder coordinates, parabolic cylinder coordinates, conical coordinates, parabolic coordinates, prolate spheroidal coordinates, oblate spheroidal coordinates, ellipsoidal coordinates and paraboloidal coordinates.

The *spherical polar coordinates* of a point $\boldsymbol{r} = (x, y, z)$ are (r, θ, ϕ), where

$$r = \sqrt{x^2 + y^2 + z^2}$$

is the distance from the origin to the point $\boldsymbol{r}$,

$$\theta = \cos^{-1}\left(\frac{z}{r}\right)$$

is the polar angle, measured from the Z axis to the line that goes from the origin to $\boldsymbol{r}$, and

$$\phi = \tan^{-1}\left(\frac{y}{x}\right)$$

is the azimuthal angle, measured from the X axis to the line in the XY plane that goes from the origin to the point $(x, y, 0)$, as is shown in Fig. 10. To locate points on the surface of a sphere (defined by the equation r = constant), one needs only the polar angle θ and the azimuthal angle ϕ.

The formulas for the spherical polar unit vectors, $\hat{\boldsymbol{r}}$, $\hat{\theta}$, and $\hat{\boldsymbol{\phi}}$ in terms of the rectangular unit vectors $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$ are

$$\hat{\boldsymbol{r}} = \hat{\boldsymbol{x}} \sin\theta \cos\phi + \hat{\boldsymbol{y}} \sin\theta \sin\phi + \hat{\boldsymbol{z}} \cos\theta$$
$$\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{x}} \cos\theta \cos\phi + \hat{\boldsymbol{y}} \cos\theta \sin\phi - \hat{\boldsymbol{z}} \sin\theta$$
$$\hat{\boldsymbol{\phi}} = -\hat{\boldsymbol{x}} \sin\phi + \hat{\boldsymbol{y}} \cos\phi$$

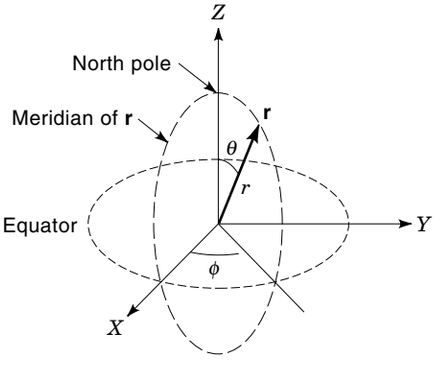


Figure 10. Spherical polar coordinates of point **r**.

These formulas imply that $\hat{\boldsymbol{r}}$, $\hat{\theta}$, and $\hat{\boldsymbol{\phi}}$ are mutually orthogonal,

$$\hat{\boldsymbol{r}} \cdot \hat{\theta} = 0$$
$$\hat{\boldsymbol{r}} \cdot \hat{\phi} = 0$$
$$\hat{\boldsymbol{\theta}} \cdot \hat{\phi} = 0$$

and therefore define a set of local Cartesian coordinate axes that are attached at the point $\boldsymbol{r}$. The inverse relations

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{r}} \sin\theta \cos\phi + \hat{\theta} \cos\theta \cos\phi - \hat{\phi} \sin\theta$$
$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{r}} \sin\theta \sin\phi + \hat{\theta} \cos\theta \sin\phi + \hat{\phi} \sin\theta$$
$$\hat{\boldsymbol{z}} = \hat{\boldsymbol{r}} \cos\theta - \hat{\theta} \sin\theta$$

give the spherical polar unit vectors in terms of the rectangular unit vectors.

To compute the spherical polar coordinates of a vector field $\boldsymbol{F}$ whose rectangular components are known, take the dot products of the spherical polar unit vectors with $\boldsymbol{F}$:

$$F_r = F_x \sin\theta \cos\phi + F_y \sin\theta \sin\phi + F_z \cos\theta$$
$$F_\theta = F_x \cos\theta \cos\phi + F_y \cos\theta \sin\phi - F_z \sin\theta$$
$$F_\phi = -F_x \sin\phi + F_y \cos\phi$$

To compute the rectangular components given the spherical polar coordinates, use the formulas given above for $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$ in terms of the spherical polar unit vectors $\hat{\boldsymbol{r}}$, $\hat{\theta}$, and $\hat{\boldsymbol{\phi}}$.

VECTOR INTEGRATION

Integrals that involve vectors are important in electrical engineering because important quantities such as the electrical current through a surface or the work done in moving a charge along a three-dimensional path depend upon the magnitude and direction of a vector, as well as the orientation of a surface or line.

The *line integral* of a vector field $\boldsymbol{F}$ along a curve C is defined as the limit of the sum of contributions from line segments that approximate the curve:

$$\int_C \boldsymbol{F}(\boldsymbol{r}) \cdot d\boldsymbol{l} = \lim_{N\to\infty} \sum_{i=0}^{N-1} \boldsymbol{F}(\boldsymbol{r}_i) \cdot (\boldsymbol{r}_{i+1} - \boldsymbol{r}_i)$$

For example, this formula gives the work that is done by a force field F acting on a particle as it moves along the curve C. The potential difference V_{AB} between two points A and B is equal to minus the line integral of the electric field intensity $\boldsymbol{E}$ from A to B,

$$V_{AB} = -\int_A^B \boldsymbol{E} \cdot d\boldsymbol{l}$$

In this example, the value of the potential difference is independent of the path taken from A to B. In general, however, the value of a line integral depends upon the path. For exam-

ple, the electromotive force $\mathscr{E}$ around a wire loop L that is immersed in a time-varying magnetic field,

$$\mathscr{E} = \oint_L \boldsymbol{E} \cdot d\boldsymbol{l}$$

depends upon the loop's size, shape, and number of turns.

The *surface integral* of a vector field $\boldsymbol{F}$ over a surface S is the limit of the sum, over plane polygons (with directed areas $\Delta \boldsymbol{S}_i$) that approximate S, of products of the form (area of $\Delta \boldsymbol{S}_i$ times component of $\boldsymbol{F}$ perpendicular to $\Delta \boldsymbol{S}_i$):

$$\int_S \boldsymbol{F}(\boldsymbol{r}) \cdot d\boldsymbol{S} = \lim_{N\to\infty} \sum_{i=0}^{N-1} \boldsymbol{F}(\boldsymbol{r}_i) \cdot \Delta \boldsymbol{S}_i$$

For example, the electric current I_S that flows through a surface S is equal to the surface integral of the current density $\boldsymbol{J}$ over S,

$$I_S = \int_S \boldsymbol{J}(\boldsymbol{r}) \cdot d\boldsymbol{S}$$

Gauss' law states that the electric flux through a closed surface, $\int_S \boldsymbol{D} \cdot d\boldsymbol{S}$, is equal to the charge enclosed in S.

Several important formulas involve the integral of a scalar field ψ, which is defined as the limit of a sum, over small volumes ΔV_i that approximate V, of products of the form (value of ψ times volume ΔV_i):

$$\int_V \psi(\boldsymbol{r}) dV = \lim_{N\to\infty} \sum_{i=0}^{N-1} \psi(\boldsymbol{r}_i) \Delta V_i$$

For example, the total electric charge distributed over a volume V is equal to the volume integral of the electric charge density ρ.

VECTOR OPERATORS

Electromagnetic engineering makes intensive use of the gradient, divergence, curl, and Laplacian, which are called *vector differential operators* because they either operate on vectors, produce a vector result, or both.

The *gradient* of a scalar field ψ (written as $\nabla\psi$, or sometimes as gradψ) is a vector field that, at each location $\boldsymbol{r}$ in space, points in the direction in which ψ increases most rapidly. The magnitude of $\nabla\psi$ is equal to the maximum rate at which ψ changes. These properties are evident from the definition of the gradient in terms of its component along an arbitrary unit vector $\hat{\boldsymbol{n}}$:

$$\hat{\boldsymbol{n}} \cdot (\nabla\psi(\boldsymbol{r})) = \lim_{\Delta l\to 0} \frac{\psi(\boldsymbol{r} + \hat{\boldsymbol{n}}\Delta l) - \psi(\boldsymbol{r})}{\Delta l}$$

The change in ψ under a small displacement from $\boldsymbol{r}$ to $\boldsymbol{r} + \hat{\boldsymbol{n}}\Delta l$ is approximately equal to the dot product of $\hat{\boldsymbol{n}}\Delta l$ with $\nabla\psi$:

$$\psi(\boldsymbol{r} + \hat{\boldsymbol{n}}\Delta l) - \psi(\boldsymbol{r}) \approx (\nabla\psi(\boldsymbol{r})) \cdot (\hat{\boldsymbol{n}}\Delta l)$$

For example, if ψ is the electric potential, then $\Delta\psi$ is the work done by an external force in moving a unit charge from $\boldsymbol{r}$ to $\boldsymbol{r} + \hat{\boldsymbol{n}}\Delta l$. The equation $\boldsymbol{E} = -\nabla\psi$ implies that the electric field intensity is equal to minus the gradient of the electric potential. The *gradient theorem* states that the line integral from a point $\boldsymbol{r}$ to a point $\boldsymbol{s}$ of the gradient of a scalar field ψ is equal to the difference of the values of ψ at $\boldsymbol{s}$ and $\boldsymbol{r}$:

$$\int_r^s (\nabla\psi) \cdot d\boldsymbol{l} = \psi(\boldsymbol{s}) - \psi(\boldsymbol{r})$$

A special case is the familiar formula for the integral of an ordinary derivative,

$$\int_a^b \frac{d\psi}{dx} dx = \psi(b) - \psi(a)$$

An extremely important consequence of the gradient theorem is the fact that the integral of a gradient is independent of the path of integration.

Formulas for the basic vector operators in the three most important coordinate systems (rectangular, circular cylindrical and spherical polar coordinates) are needed constantly in electromagnetics. The gradient of a scalar field ψ is

$$\nabla\psi = \frac{\partial\psi}{\partial x}\hat{\boldsymbol{x}} + \frac{\partial\psi}{\partial y}\hat{\boldsymbol{y}} + \frac{\partial\psi}{\partial z}\hat{\boldsymbol{z}}$$

in Cartesian coordinates,

$$\nabla\psi = \frac{\partial\psi}{\partial\rho}\hat{\rho} + \frac{1}{\rho}\frac{\partial\psi}{\partial\phi}\hat{\phi} + \frac{\partial\psi}{\partial z}\hat{\boldsymbol{z}}$$

in circular cylindrical coordinates, and

$$\nabla\psi = \frac{\partial\psi}{\partial r}\hat{\boldsymbol{r}} + \frac{1}{r}\frac{\partial\psi}{\partial\theta}\hat{\theta} + \frac{1}{r\sin\theta}\frac{\partial\psi}{\partial\phi}\hat{\phi}$$

in spherical polar coordinates.

The *divergence* of a vector field $\boldsymbol{F}$ (written as $\nabla \cdot \boldsymbol{F}$, or sometimes as div$\boldsymbol{F}$) is defined as the limit, as the volume ΔV approaches zero, of the integral of $\boldsymbol{F}$ over the surface ΔS that bounds ΔV, divided by ΔV:

$$\nabla \cdot \boldsymbol{F} = \lim_{\Delta V\to 0} \frac{\int_{\Delta S} \boldsymbol{F} \cdot d\boldsymbol{S}}{\Delta V}$$

If $\boldsymbol{F}$ is a flux vector (a rate of transfer of some physical quantity Q per unit area, per unit time), then the divergence $\nabla \cdot \boldsymbol{F}$ is equal to the rate (per unit volume) at which Q is created or destroyed. In other words, the divergence of $\boldsymbol{F}$ is nonzero wherever there are sources or sinks of the physical quantity Q of which $\boldsymbol{F}$ is the flux. *Gauss' theorem,*

$$\int_V (\nabla \cdot \boldsymbol{F})\, dV = \int_S \boldsymbol{F} \cdot d\boldsymbol{S}$$

states that the integral of the divergence of $\boldsymbol{F}$ over any volume V is equal to the integral of $\boldsymbol{F}$ over the surface S that bounds V. For example, according to Gauss' theorem and one of Maxwell's equations,

$$\int_S \boldsymbol{D} \cdot d\boldsymbol{S} = \int_V \nabla \cdot \boldsymbol{D} dV = \int_V \rho dV = Q_V$$

where $\boldsymbol{D}$ is the electric flux density, ρ is the electric charge density, and Q_V is the electric charge enclosed in V. This is simply a restatement of Gauss' law. The conservation of electric charge can be expressed in the equation

$$\nabla \cdot \boldsymbol{J} + \frac{\partial \rho}{\partial t} = 0$$

in terms of the divergence of the electric current density $\boldsymbol{J}$.

Explicit formulas for the divergence $\nabla \cdot \boldsymbol{F}$ of a vector field $\boldsymbol{F}$ are

$$\nabla \cdot \boldsymbol{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}$$

in Cartesian coordinates,

$$\nabla \cdot \boldsymbol{F} = \frac{\partial (\rho F_\rho)}{\partial \rho} + \frac{1}{\rho}\frac{\partial F_\phi}{\partial \phi} + \frac{\partial F_z}{\partial z}$$

in circular cylindrical coordinates, and

$$\nabla \cdot \boldsymbol{F} = \frac{1}{r^2}\frac{\partial (r^2 F_r)}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial (\sin\theta F_\theta)}{\partial \theta} + \frac{1}{r\sin\theta}\frac{\partial F_\phi}{\partial \phi}$$

in spherical polar coordinates.

The *curl* of a vector field $\boldsymbol{F}$ (written as $\nabla \times \boldsymbol{F}$, or sometimes as curl$\boldsymbol{F}$) is defined as the limit, as the area of a small, flat surface $\Delta \boldsymbol{S}$ ($\perp \hat{\boldsymbol{n}}$) approaches zero, of the line integral of $\boldsymbol{F}$ around the closed curve ΔC that bounds $\Delta \boldsymbol{S}$:

$$\hat{\boldsymbol{n}} \cdot (\nabla \times \boldsymbol{F}) = \lim_{\Delta S \to 0} \frac{\oint_{\Delta C} \boldsymbol{F} \cdot d\boldsymbol{l}}{\Delta S}$$

Stokes' theorem,

$$\int_S (\nabla \times \boldsymbol{F}) \cdot d\boldsymbol{S} = \oint_C \boldsymbol{F} \cdot d\boldsymbol{l}$$

states that the line integral of a vector field $\boldsymbol{F}$ around a closed curve C is equal to the integral of the curl of $\boldsymbol{F}$ over any surface enclosed by C. Stokes' theorem and Faraday's law,

$$\mathscr{E} = \oint_C \boldsymbol{E} \cdot d\boldsymbol{l} = -\frac{\partial \Phi_C}{\partial t}$$

(where $\Phi_C = \int_S \boldsymbol{B} \cdot d\boldsymbol{S}$ is the magnetic flux enclosed by C, and $\boldsymbol{B}$ is the magnetic induction), imply one of Maxwell's equations,

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$$

In a fluid, the curl of the velocity field $\boldsymbol{v}$ is nonzero only where there is a net circulation of fluid, as there is in a whirlpool or eddy. In a region of space, or in a material, in which there are no conduction currents, the electric field intensity $\boldsymbol{E}$ obeys the wave equation

$$\nabla \times (\nabla \times \boldsymbol{E}) + \mu\epsilon \frac{\partial^2 \boldsymbol{E}}{\partial t^2} = \boldsymbol{0}$$

The curl, gradient and divergence obey the operator identities

$$\nabla \times (\nabla \psi) = \boldsymbol{0}$$

(the curl of a gradient is zero) and

$$\nabla \cdot (\nabla \times \boldsymbol{F}) = 0$$

(the divergence of a curl is zero). The relation $\nabla \times (\nabla \psi) = \boldsymbol{0}$ implies that the integral of a gradient around a closed path vanishes, and the relation $\nabla \cdot (\nabla \times \boldsymbol{F}) = 0$ implies that the curl of a vector field cannot describe sources or sinks.

Formulas for the curl $\nabla \times \boldsymbol{F}$ of a vector field $\boldsymbol{F}$ in the major coordinate systems are

$$\nabla \times \boldsymbol{F} = \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}\right)\hat{\boldsymbol{x}} + \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}\right)\hat{\boldsymbol{y}} + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right)\hat{\boldsymbol{z}}$$

in Cartesian coordinates,

$$\nabla \times \boldsymbol{F} = \left(\frac{1}{\rho}\frac{\partial F_z}{\partial \phi} - \frac{\partial F_\phi}{\partial z}\right)\hat{\boldsymbol{\rho}} + \left(\frac{\partial F_\rho}{\partial z} - \frac{\partial F_z}{\partial \rho}\right)\hat{\boldsymbol{\phi}} + \frac{1}{\rho}\left(\frac{\partial (\rho F_\phi)}{\partial \rho} - \frac{\partial F_\rho}{\partial \phi}\right)\hat{\boldsymbol{z}}$$

in circular cylindrical coordinates, and

$$\nabla \times \boldsymbol{F} = \frac{1}{r\sin\theta}\left(\frac{\partial (\sin\theta F_\phi)}{\partial \theta} - \frac{\partial F_\theta}{\partial \phi}\right)\hat{\boldsymbol{r}} + \frac{1}{r}\left(\frac{1}{\sin\theta}\frac{\partial F_r}{\partial \phi} - \frac{\partial (r F_\phi)}{\partial r}\right)\hat{\theta} + \frac{1}{r}\left(\frac{\partial (r F_\theta)}{\partial r} - \frac{\partial F_r}{\partial \theta}\right)\hat{\phi}$$

in spherical polar coordinates.

The *Laplacian* operator on a scalar field ψ (usually written as $\nabla^2 \psi$) is defined as the divergence of the gradient:

$$\nabla^2 \psi = \nabla \cdot (\nabla \psi)$$

The Laplacian of a scalar field ψ is nonzero only at points at which there are sources or sinks of the vector field $\nabla \psi$. For example, the right-hand side of Poisson's equation,

$$\nabla^2 \psi = -\frac{\rho}{\epsilon_0}$$

is nonzero only where the electric charge density ρ is nonzero.

In Cartesian coordinates, the formula for the Laplacian is

$$\nabla^2 \psi = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \frac{\partial^2 \psi}{\partial z^2}$$

The formulas for the Laplacian in the most important orthogonal curvilinear coordinate systems are

$$\nabla^2 \psi = \frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho \frac{\partial \psi}{\partial \rho}\right) + \frac{1}{\rho^2}\frac{\partial^2 \psi}{\partial \phi^2} + \frac{\partial^2 \psi}{\partial z^2}$$

in circular cylindrical coordinates, and

$$\nabla^2 \psi = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial \psi}{\partial r}\right) + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta \frac{\partial \psi}{\partial \theta}\right) + \frac{1}{r^2 \sin^2\theta}\frac{\partial^2 \psi}{\partial \phi^2}$$

in spherical polar coordinates.

The Laplacian of a vector field is defined in such a way that the equation

$$\nabla \times (\nabla \times \boldsymbol{F}) = \nabla(\nabla \cdot \boldsymbol{F}) - \nabla^2 \boldsymbol{F}$$

is true in all orthogonal curvilinear coordinate systems. In Cartesian coordinates, the Laplacian acts only on the components of a vector field $\boldsymbol{F}$:

$$\nabla^2 \boldsymbol{F} = (\nabla^2 F_x)\hat{\boldsymbol{x}} + (\nabla^2 F_y)\hat{\boldsymbol{y}} + (\nabla^2 F_z)\hat{\boldsymbol{z}}$$

However, in curvilinear coordinates (such as circular cylindrical or spherical polar coordinates), the gradient and divergence operators act on the unit vectors as well as on the components of $\boldsymbol{F}$, because the unit vectors depend upon the position vector, $\boldsymbol{r}$.

BIBLIOGRAPHY

M. L. Boas, *Mathematical Methods in the Physical Sciences,* 2nd ed., New York: Wiley, 1983.

C. D. Cantrell, *Modern Mathematical Methods for Physicists and Engineers,* Cambridge University Press, 1998.

D. A. Danielson, *Vectors and Tensors in Engineering and Physics,* Reading, MA: Addison-Wesley, 1997.

G. H. Golub and C. F. Van Loan, *Matrix Computations,* 3rd ed., Baltimore, MD: Johns Hopkins University Press, 1996.

G. E. Hay, *Vector and Tensor Analysis,* New York: Dover, 1979.

B. Hoffmann, *About Vectors,* New York: Dover, 1975.

E. Kreyszig, *Advanced Engineering Mathematics,* 7th ed., New York: Wiley, 1992.

P. Liebeck, *Vectors and Matrices,* New York: Pergamon, 1972.

P. A. Morse and H. Feshbach, *Methods of Theoretical Physics,* Vols. I and II, New York: McGraw-Hill, 1953.

C. D. CANTRELL
University of Texas at Dallas

VECTOR SPACES, LINEAR. See LINEAR ALGEBRA.

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Walsh Functions
Standard Article

Mohsen Razzaghi[1] and Jalal Nazarzadeh[2]
[1]Mississippi State University, Mississippi State, MS
[2]Amirkabir University, Tehran, Iran

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2465
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (172K)

Abstract

The sections in this article are

Different Types of Piecewise Constant Basis Function

Walsh Function Generator

Relationship Between Walsh and Fourier Series

Summary

| | |

Copyright © 1999-2008 All Rights Reserved.

WALSH FUNCTIONS

Orthogonal properties (1,2) of the familiar sine–cosine functions have been known for over two centuries. Use of such functions in an elegant manner to solve complex analytical problems was initiated by the work of the famous mathematician Baron Jean-Baptiste-Joseph Fourier (3). The system of sine and cosine functions plays a distinguished role in many areas of electrical engineering. There are a number of historical and practical reasons for this. From the theoretical point of view, one of the major reasons is that Fourier series and Fourier transform permit the representation of a large class of functions by a superposition of sine and cosine functions. This representation makes it possible to apply the concept of frequency, which was originally defined for sine and cosine only, to other functions.

In the fields of circuit analysis, control theory, and communications the complete and orthogonal properties of sine and cosine functions produce attractive solutions. But with the application of digital techniques and semiconductor technology in these areas, awareness for other more general complete systems of orthogonal functions has developed. This class of functions, though not possessing some of the desirable properties of sine-cosine functions in linear time-invariant networks, has other advantages rendering its use more directly applicable to all such applications in the context of digital technology. Many members of this class of orthogonal functions are piecewise constant basis functions (PCBF), thus resembling the high-low switching characteristic of semiconductor devices. Walsh functions belong to the class of PCBFs that have been developed in the twentieth century and have played an important role in scientific and engineering applications. The mathematical techniques of studying functions, signals, and systems through series expansions in orthogonal complete sets of basis functions are now a standard tool in all branches of science and engineering. Actually, the signals involved in Morse telegraphy are PCBFs, but no mathematical study of these signals was made prior to the beginning of the twentieth century.

The origin of the mathematical study of PCBFs is due to the Hungarian mathematician Alfred Haar (studies completed 1910–1912), who used a set of functions now bearing his name. These functions have not found much use in comparison to the Walsh and block-pulse functions. The development and utilization of Walsh functions has been strongly influenced by the parallel developments in digital electronics and computer science and engineering. Efforts to replace Fourier transforms by Walsh-type transforms have been made in communication, signal processing, image processing, pattern recognition, and so forth. Applications of Walsh functions in the systems and control field were begun only about two decades ago, and developments since then have occurred rapidly.

DIFFERENT TYPES OF PIECEWISE CONSTANT BASIS FUNCTION

Haar Functions

The set of Haar functions is periodic, orthogonal, and complete and was proposed in 1910 by Alfred Haar (4). Figure 1 shows the set of first eight Haar functions. A recurrence relation that enables one to generate the Haar functions {har(j, n, t)} in the semi-open interval $t \in [0, 1)$ is given by (5). The first member of the set is

$$\text{har}(0, 0, t) = 1 \qquad t \in [0, 1)$$

while the general term for other members is given by

$$\text{har}(j, n, t) = \begin{cases} 2^{j/2} & \dfrac{n-1}{2^j} \le t < \dfrac{n-0.5}{2^j} \\ -2^{j/2} & \dfrac{n-0.5}{2^j} \le t < \dfrac{n}{2^j} \\ 0 & \text{elsewhere} \end{cases}$$

where j, n, and m are integers governed by the relation

$$0 \le j \le \log_2 m \qquad 1 \le n \le 2^j$$

The number of members in the set is of the form $m = 2^k$, k being a positive integer.

Haar's set is such that the formal expansion of a given continuous function in terms of Haar functions converges uniformly to the given function (6).

Rademacher Functions

Rademacher functions are an incomplete set of orthonormal functions which were developed by the German mathemati-

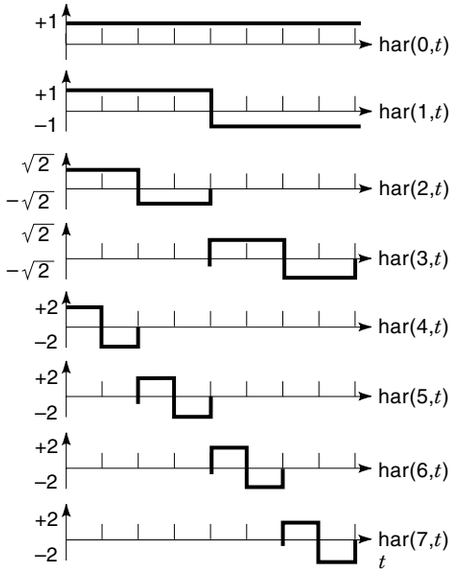


Figure 1. A set of the first eight Haar functions.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

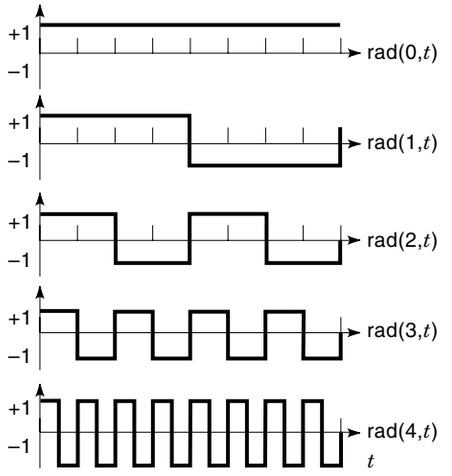


Figure 2. A set of the first five Rademacher functions.

cian H. Rademacher in 1922 (7). Figure 2 shows the set of the first five Rademacher functions. The Rademacher function of index m, denoted by rad(m, t) is given by a square wave of unit amplitude and 2^{m-1} cycle in the semi-open interval [0, 1), with the exception of rad(0, t) which has a constant value of unity throughout the interval. Rademacher functions can be generated using the recurrence relation (8)

$$\text{rad}(m,t) = \text{rad}(1, 2^{m-1}t), \qquad m \neq 0$$

with

$$\text{rad}(1,t) = \begin{cases} 1 & t \in [0, 0.5) \\ -1 & t \in [0.5, 1) \end{cases}$$

Walsh Functions

The incomplete set of Rademacher functions was completed by J. L. Walsh in 1923, to form the complete orthogonal set of rectangular functions we now call the Walsh functions (9).

As indicated by Walsh, there are many possible orthogonal function sets of this kind. Since Walsh's work several researchers have suggested orthogonal series formed with the help of combinations of the well-known piecewise constant orthogonal functions (10–12).

In his original paper Walsh pointed out that, ". . . Haar's set is, however, merely one of an infinity of sets which can be constructed of functions of this same character." While proposing his new set of orthogonal functions, Walsh wrote, ". . . each function takes only the values +1 and −1 except at a finite number of points of discontinuity, where it takes the value zero."

It is interesting to note that some of the square wave patterns of individual Walsh functions appear in several ancient designs (13). Chess board or checker board designs are two-dimensional Walsh functions, whereas the Rubik Cube is a three-dimensional Walsh function.

The set of Walsh functions is generally classified into three groups. These groups differ from one another in that the order in which individual functions appear is different. The three types of orderings are: (1) Sequency or Walsh ordering, (2) Dyadic or Paley ordering, and, (3) Natural or Hadamard ordering. In what follows, we discuss some aspects of each of these orderings.

Sequency or Walsh Ordering

This is the ordering which was originally employed by Walsh (9). Sequency ordered Walsh functions are arranged in ascending order of zero crossings. Sequency is defined as one-half the average number of zero crossings over the unit interval [0, 1), and is used as a measure of generalized frequency of wave forms. Figure 3 shows a set of the first eight sequency order Walsh functions wal(m, t), where m is the sequency order number and $0 \leq t < 1$.

If each waveform is divided into eight intervals, the magnitude of the waveform can be expressed as a matrix

$$W(m,l) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \tag{1}$$

where m denotes the order of Walsh function (the row of the matrix), l the corresponding bit of this order (the column of the same matrix), and $W(m, l)$ is called the Walsh matrix.

Walsh functions are either symmetrical or asymmetrical with respect to their middle point. They are called *cal* and *sal* functions respectively. These functions are expressed as

$$\text{wal}(2m,t) = \text{cal}(m,t) \qquad m = 1, 2, \ldots, \frac{N}{2} \tag{2}$$

$$\text{wal}(2m-1,t) = \text{sal}(m,t) \qquad m = 1, 2, \ldots, \frac{N}{2} \tag{3}$$

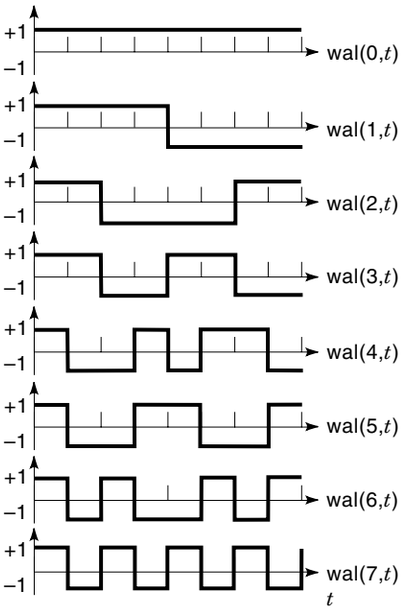


Figure 3. A set of the first eight Walsh functions arranged in sequency order.

Because of their symmetrical characteristic, sal and cal terms can be thought of as being analogous to the sine and cosine terms of the Fourier series.

Similarly to the Fourier series representation, the Walsh series representation of a time function that is absolutely integrable in [0, 1) is defined as

$$f(t) = \sum_{m=0}^{\infty} F_m \text{wal}(m, t) \tag{4}$$

where F_m is the coefficient of the Walsh function of $f(t)$. It is desirable to determine the coefficient such that the integral square error is minimized

$$\epsilon = \int_0^1 \left[f(t) - \sum_{m=0}^{\infty} F_m \text{wal}(m, t) \right]^2 dt$$

Taking the partial derivative of ϵ with respect to F_m and setting it equal to zero yields

$$F_m = \int_0^1 f(t) \text{wal}(m, t)\, dt \qquad m = 0, 1, 2, \ldots \tag{5}$$

This simple result is due to the orthonormal property of Walsh functions. Let us illustrate the Walsh series expansion for the ramp function

$$f(t) = t$$

Substituting $f(t)$ into Eq. (5) and taking four terms yields

$$F_0 = \int_0^1 t \text{wal}(0, t)\, dt = \frac{1}{2}$$
$$F_1 = \int_0^1 t \text{wal}(1, t)\, dt = -\frac{1}{4}$$
$$F_2 = \int_0^1 t \text{wal}(2, t)\, dt = 0$$
$$F_3 = \int_0^1 t \text{wal}(3, t)\, dt = -\frac{1}{8}$$

After substituting these obtained values of coefficients into Eq. (4) we have

$$t = \frac{1}{2}\text{wal}(0, t) - \frac{1}{4}\text{wal}(1, t) - \frac{1}{8}\text{wal}(3, t)$$

which is the four term sequency ordered Walsh function series expansion of the ramp function.

Dyadic or Paley Ordering

The dyadic type of ordering was introduced by Paley (14). The dyadic order is obtained by generating Walsh functions from successive Rademacher functions. The set of Walsh and Rademacher functions that are referred to here as pal(n, t) and rad(q, t) respectively have the following relation:

$$\text{pal}(0, t) = \text{rad}(0, t)$$
$$\text{pal}(1, t) = \text{rad}(1, t)$$
$$\text{pal}(2, t) = [\text{rad}(2, t)]^1 [\text{rad}(1, t)]^0$$
$$\text{rad}(3, t) = [\text{rad}(2, t)]^1 [\text{rad}(1, t)]^1$$
$$\text{pal}(4, t) = [\text{rad}(3, t)]^1 [\text{rad}(2, t)]^0 [\text{rad}(1, t)]^0$$
$$\text{pal}(5, t) = [\text{rad}(3, t)]^1 [\text{rad}(2, t)]^0 [\text{rad}(1, t)]^1$$
$$\text{pal}(6, t) = [\text{rad}(3, t)]^1 [\text{rad}(2, t)]^1 [\text{rad}(1, t)]^0$$
$$\text{pal}(7, t) = [\text{rad}(3, t)]^1 [\text{rad}(2, t)]^1 [\text{rad}(1, t)]^1$$
$$\vdots$$
$$\text{pal}(n, t) = [\text{rad}(q, t)]^{b_q} [\text{rad}(q-1, t)]^{b_{q-1}} \cdots [\text{rad}(1, t)]^{b_1}$$

where

$$q = [\log_2(n)] + 1 \tag{6}$$

in which $[\cdot]$ means taking the greatest integer. Therefore,

$$n = b_q 2^{q-1} + b_{q-1} 2^{q-2} + \cdots + b_1 2^0$$

where $b_q b_{q-1} \cdots b_1$ is the binary expression of n.

Hence, if a particular Walsh function pal(n, t) is given and its Rademacher function components are required, we simply change n into binary form and then substitute into Eq. (6). For example, the Rademacher function components of Walsh function pal(10, t) is

$$\text{pal}(10, t) = [\text{rad}(4, t)]^1 [\text{rad}(3, t)]^0 [\text{rad}(2, t)]^1 [\text{rad}(1, t)]^0$$

where

$$q = [\log_2 10] + 1 = 4$$

because Rademacher functions are easy to draw, as are Walsh functions. Figure 4 shows the Walsh functions in Paley ordering from pal(0, t) to pal(7, t).

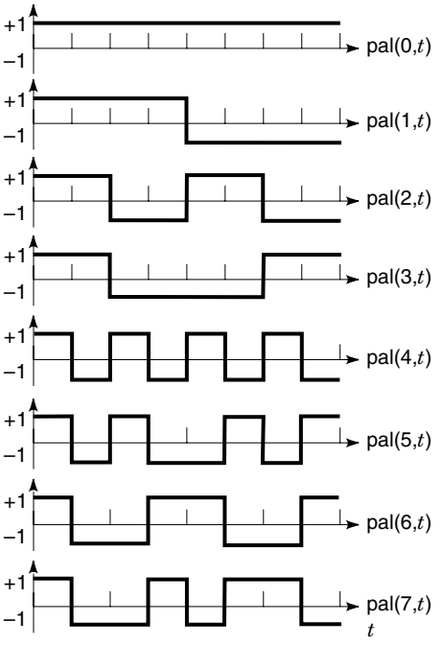


Figure 4. A set of the first eight Walsh functions arranged in dyadic order.

Since all Radmacher functions except rad(0, t) are odd functions about $t = 0.5$, they do not form a complete set. On the contrary, one can see that the Walsh functions constitute a complete orthonormal set of functions. The Walsh series representation of a function $f(t)$, which is absolutely integrable in [0, 1) in a dyadic ordering is

$$f(t) = \sum_{m=0}^{\infty} c_m \text{pal}(m, t) \tag{7}$$

where

$$c_m = \int_0^1 f(t)\text{pal}(m, t)\, dt \qquad m = 0, 1, \ldots \tag{8}$$

Let us now return to the Walsh coefficient evaluation in dyadic ordering for ramp function. Substituting $f(t) = t$ into Eqs. (7) and (8), we have

$$f(t) = t = \frac{1}{2}\text{pal}(0, t) - \frac{1}{4}\text{pal}(1, t) - \frac{1}{8}\text{pal}(2, t) - \frac{1}{16}\text{pal}(4, t) - \frac{1}{32}\text{pal}(8, t) - \frac{1}{64}\text{pal}(16, t) + \cdots \tag{9}$$

The original curve $f(t) = t$ and its Walsh series approximations are shown in Figure 5. They are stairways waves. The first representation is obtained by taking one term of the Walsh series, or $\frac{1}{2}$ pal(0, t); the second one is $\frac{1}{2}$ pal(0, t) $-$ $\frac{1}{4}$ pal(1, t). Figure 5 shows up to a four term approximation. From the coefficient evaluation process, we can easily see the similarities between the Fourier series and Walsh series.

Natural or Hadamard Ordering

This ordering was originally proposed by Henderson (15) and follows the Hadamard matrix derived from successive Kronecker products. A Hadamard matrix is a square array whose coefficients comprise only $+1$ and -1 and in which the rows (and columns) are orthogonal to one another. In a symmetrical Hadamard matrix it is possible to interchange rows and columns or to change the sign of every element in a row without affecting these orthogonal properties. This makes it possible to obtain a symmetrical Hadamard matrix whose first row and first column contain only $+1$'s. The matrix obtained in this way is known as the normal form for the Hadamard matrix. The lowest-order Hadamard matrix is of order two,

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Higher-order matrices, restricted to having powers of 2, can be obtained from the recursive relationship

$$H_N = H_{N/2} \otimes H_2$$

where $\otimes$ denotes the direct or Kronecker product (16) and N is a power of 2. In the Kronecker product each element in the matrix (in this case $H_{N/2}$) is replaced by the matrix H_2. Thus, for $N = 4$ we have

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Furthermore, if we now replace each element in the H_4 matrix by an H_2 matrix we obtain an H_8 matrix. By replacing each row of this matrix by its equivalent naturally ordered Walsh functions we can form a series of functions which will indicate the ordering obtained through this derivation. Therefore, for a series consisting of eight terms we get

$$H_8 = H_4 \otimes H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} \text{had}(0, t) \\ \text{had}(1, t) \\ \text{had}(2, t) \\ \text{had}(3, t) \\ \text{had}(4, t) \\ \text{had}(5, t) \\ \text{had}(6, t) \\ \text{had}(7, t) \end{bmatrix}$$

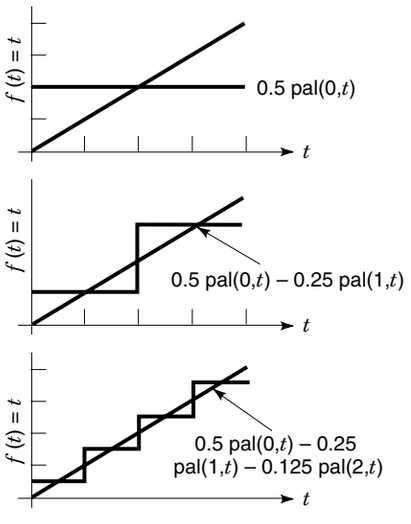


Figure 5. Expanding a ramp function into a Walsh series.

Relationship Between Ordered Series

The wal(i, t), pal(i, t), and had(i, t), $i = 0, 1, 2, \ldots$ ordered Walsh functions are related (1) through a bit reversal for the position of each component in a series, (2) through a conversion using a Gray code or (3) by a combination of both of these. For example, given a function numbered in dyadic ordering, the corresponding sequency order is given by

$$\text{pal}(n, t) = \text{wal}(b(n), t)$$

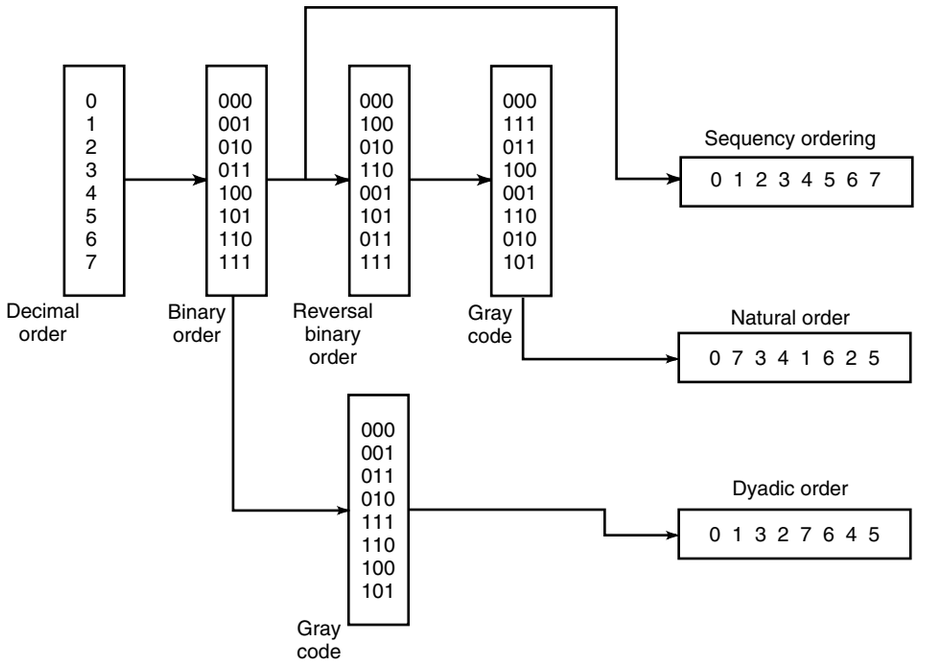


Figure 6. Relationships between three methods of ordering the Walsh functions series.

where $b(n)$ is a Gray-code-to-binary conversion of n. A procedure for carrying out this conversion is described by Yuen (17). These relationships for $N = 8$ are shown in Fig. 6 in which both dyadic and natural ordering are related to a sequency ordered.

Block-Pulse Functions

Block-pulse functions constitute another complete set of orthogonal basis functions. The type of approximation is the same as with Walsh functions, the only difference being in the simplicity of computations. The block-pulse function $b(i, t)$, $i = 1, 2, \ldots, m$ over a time interval $t \in [0, 1)$ is defined as

$$b(i,t) = \begin{cases} 1 & \dfrac{i-1}{m} \le t < \dfrac{i}{m} \quad i = 1, 2, \ldots, m \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Thus, as shown in Fig. 7, the 8-order block-pulse functions are time functions having a unit height and $\frac{1}{8}$ width. By using the orthogonal property that is

$$\int_0^1 b(i,t)b(j,t)\,dt = \begin{cases} \dfrac{1}{m} & i = j \quad i, j = 1, 2, \ldots, m \\ 0 & i \ne j \end{cases}$$

A time function $f(t)$ which is absolutely integrable in $t \in [0, 1)$ can be approximately represented by a block-pulse series as

$$f(t) = \sum_{i=1}^{m} a_i b(i,t) = \gamma^T B(t)$$

where

$$\alpha_i = m \int_{i-1/m}^{i/m} f(t)\,dt \qquad i = 1, 2, \ldots, m$$

and

$$\gamma = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_m]^T$$
$$B(t) = [b(1,t) \quad b(2,t) \quad \cdots \quad b(m,t)]^T$$

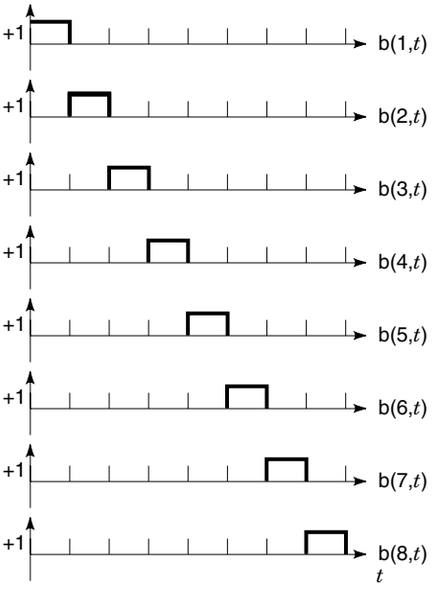


Figure 7. A set of the first eight block-pulse functions.

Relationship between Walsh and Block-Pulse Functions

A one-to-one relationship between Walsh and block-pulse functions was first offered in (18). In their work, Chen et al. used block-pulse operational matrices for simplifying their approach in Walsh domain analysis. The completeness of the block-pulse functions was first proved by Rao and Srinivasan (19) and later the convergent properties of the block-pulse series as well as its completeness were investigated by Kwong and Chen (20).

It was shown by Chen et al. (18) that the block-pulse functions $b(i, t)$ are related to the Walsh functions $\text{wal}(i, t)$ by the relation

$$\phi_b(t) = W_{(m\times m)}\phi_w(t)$$

where $W_{(m\times m)}$ is a square matrix of order m called the Walsh matrix, $\phi_b(t)$ and $\phi_w(t)$ are block-pulse and Walsh vectors respectively, defined by

$$\phi_b(t) = [b(1,t) \quad b(2,t) \quad \cdots \quad b(m,t)]^T$$
$$\phi_w(t) = [\text{wal}(0,t) \quad \text{wal}(1,t) \quad \cdots \quad \text{wal}(m-1,t)]^T$$

The Walsh matrix has the following property:

$$W^2_{(m\times m)} = mI_{(m)}$$

where $I_{(m)}$ is a unit matrix of order m. For $m = 8$ Walsh matrix is given by Eq. (1).

Construction of any function in time domain is easier with block-pulse functions rather than with Walsh functions. If a function is represented by block-pulse functions, then the amplitude of each block-pulse in any sub-interval represents the average value of that function in that particular time interval.

Some properties of the block-pulse functions are (1) they form a complete orthogonal set which could easily be normalized, (2) they are computationally much simpler, and yet produce the same numerical accuracy as that obtained by the Walsh function approach, (3) any number of block-pulse functions can be used to form a complete set, while Walsh functions require 2^p ($p = 1, 2, \ldots$) component functions to form a set suitable for analytical manipulations, and (4) they need less computation time as well as computer memory space than Walsh functions (21–23).

WALSH FUNCTION GENERATOR

The Walsh function and Walsh transform are important analytical and hardware tools for signal processing. They have found wide application in digital communications (24) as well as in system analysis (25). Walsh function generators have frequent use in many areas of electrical engineering. Implementation of such generators through hardware logic gives rise to orthogonality error. Orthogonality error is the shift of the transition points of the Walsh functions of Fig. 3 or Fig. 4 from their assigned places in the time scale.

The sequency generators having the widest applicability are those generating a set of Walsh series, although in some cases the series are obtained by first generating a series of Radmacher functions. A generator that produces a set of m Walsh functions $\text{wal}(n, t)$ where $n = 0, 1, \ldots, m - 1$, is called an array generator. Ideally the generated waves will be orthogonal to each other, and some designs are better in achieving this than are others.

Two classifications of array generators are considered. The first generates fixed sets of Walsh functions $\text{wal}(n, t)$, where only the sequency range of the entire array is controlled externally. These array generators find use in multiplexing and signal processing. The second classification includes generators for which the sequency order n and/or the time interval t are controlled externally. These are known as programmable generators. Further subclassifications of programmable generators can be defined, namely, serial programmable generators in which the time interval is fixed and the sequency order controlled and parallel programmable generators in which the sequency range is fixed and the time interval controlled.

A global Walsh generator capable of producing three different ordered outputs. These outputs are natural, sequency, and dyadic (26). This generator generates Walsh functions through logical combinations of Rademacher functions. However, these methods are all implemented using hardware digital logic and sequential circuits.

The use of microprocessors for the generation of global Walsh functions provides wider flexibility for low-cost applications which can be controlled by supporting software with better accuracy and much wider versatility. In system analysis, where the Walsh function technique provides easier mathematical manipulations, for example, power-electronic systems (27), this kind of generator can be used to study system behavior where slow speed of software based generation does not hinder the time responses.

RELATIONSHIP BETWEEN WALSH AND FOURIER SERIES

When the Walsh series representation of a time signal is required to be converted to the more familiar Fourier series representations, then the Fourier transforms of the Walsh functions are needed in the conversion equations. A nonrecursive algorithm by Siemens and Kitai (28) is used to set up the necessary conversion coefficients. A recursive formula by Blachman (29) is used to evaluate the Walsh transforms of sinusoids. This formula can also be modified to yield the Fourier transform of Walsh functions. The result in both cases is the same, although in the former case the resulting conversion matrix is more easily obtainable. A brief review of Fourier series follows.

A periodic function $f(t)$ defined over the interval 0 to 1 may be expanded into Fourier series as follows

$$f(t) = a_0 + \sum_{n=1}^{\infty}\{a_n \cos(2n\pi t) + a_n^* \sin(2n\pi t)\} \qquad (11)$$

where the Fourier coefficients a_n and a_n^* are given by

$$a_0 = \int_0^1 f(t)\,dt$$
$$a_n = 2\int_0^1 f(t)\cos(2n\pi t)\,dt \qquad n = 1, 2, \ldots$$
$$a_n^* = 2\int_0^1 f(t)\sin(2n\pi t)\,dt \qquad n = 1, 2, \ldots$$

if $f(t)$ is truncated up to its first $2r + 1$ terms, then Eq. (11) can be written as

$$f(t) = a_0 + \sum_{n=1}^{r} \{a_n \cos(2n\pi t) + a_n^* \sin(2n\pi t)\} = A^T \Psi(t) \quad (12)$$

where the Fourier series coefficient vector A and the Fourier series vector $\Psi(t)$ are defined as

$$A = [a_0 \quad a_1 \quad a_2 \quad \cdots \quad a_r \quad a_1^* \quad a_2^* \quad \cdots \quad a_r^*]^T$$
$$\Psi(t) = [\phi_0(t) \quad \phi_1(t) \quad \cdots \quad \phi_r(t) \quad \phi_1^*(t) \quad \cdots \quad \phi_r^*(t)]^T$$

with

$$\phi_n(t) = \cos(2n\pi t) \qquad n = 0, 1, 2, \ldots$$
$$\phi_n^*(t) = \sin(2n\pi t) \qquad n = 1, 2, \ldots$$

The elements of $\Psi(t)$ are orthogonal in the interval $t \in [0, 1)$.

The sal and cal terms defined in Eqs. (2) and (3) for the Walsh functions are analogous to sine and cosine terms in the Fourier series, respectively. In a similar fashion to Fourier series expansion by truncating Eq. (4), any time signal $f(t)$ can be expressed as a sum of Walsh functions as

$$f(t) = d_0 \mathrm{wal}(0, t) + \sum_{i=1}^{m-1} d_i \mathrm{wal}(i, t) = D^T \phi_w(t) \quad (13)$$

where

$$d_0 = 2 \int_0^1 f(t) \mathrm{wal}(0, t)\, dt$$
$$d_1 = \int_0^1 f(t) \mathrm{wal}(i, t)\, dt, \quad i = 1, \ldots, m-1$$

and

$$D = [d_0 \quad d_1 \quad \cdots \quad d_{m-1}]^T$$

Using Eqs. (11) and (12), the following expression holds

$$B\phi_w(t) = \Psi(t)$$

where B is the Fourier–Walsh conversion matrix.

The inverse relation is also valid

$$\phi_w(t) = B^{-1} \Psi(t)$$

The following steps can be used to create the conversion matrix B:

1. For Walsh function order number v, obtain its binary equivalent expression b.
2. Convert b to its Gray code equivalent y.
3. The total number of bits in y is h and the number of bits with the binary value 1 in y is d.
4. The Fourier coefficient of order u of the Walsh function of order v appears as the element in row u and column v of the conversion matrix B. These elements can be calculated according to the following equations (30).

$$B(u, v) = 2(-1)^{y_0} (-j)^d \left[\prod_{w=0}^{h-1} \cos\left(\frac{u\pi}{2^{u+1}} - \frac{y_w \pi}{2} \right) \right] \times \frac{\sin\left(\frac{u\pi}{2^h}\right)}{\frac{u\pi}{2^h}}$$

where y_0 is the least significant bit in the Gray code expression of y and $j = \sqrt{-1}$.

The sequency-ordered matrix B of order 8×8 can be obtained as follows (30,31):

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.27 & 0 & -0.527 & 0 & -0.105 & 0 & -0.253 \\ 0 & 0 & 1.27 & 0 & 0 & 0 & -0.527 & 0 \\ 0 & 0.424 & 0 & 1.02 & 0 & -0.685 & 0 & 0.284 \\ 0 & 0 & 0 & 0 & 1.27 & 0 & 0 & 0 \\ 0 & 0.255 & 0 & 0.615 & 0 & 0.092 & 0 & -0.381 \\ 0 & 0 & 0.424 & 0 & 0 & 0 & 1.02 & 0 \\ 0 & 0.182 & 0 & -0.007 & 0 & 0.379 & 0 & 0.914 \end{bmatrix}$$

Relationship between Block-Pulse and Fourier Series

Using Eq. (10), the Fourier series for $b(n, t)$ is given by

$$b(n, t) = b_0 + \sum_{i=1}^{k} [b_{ni} \cos(2i\pi t) + b_{ni}^* \sin(2i\pi t)] \quad (14)$$

where

$$b_0 = \int_0^1 b(n, t)\, dt = \frac{1}{m} \quad (15)$$

$$b_{ni} = 2 \int_0^1 b(n, t) \cos(2\pi i t)\, dt = \frac{2}{\pi i} \sin\left(\frac{\pi i}{m}\right) \cos\left(\frac{\pi i}{m}(2n-1)\right) \qquad \begin{matrix} i = 1, \ldots, k \\ n = 1, \ldots, m \end{matrix} \quad (16)$$

and

$$b_{ni}^* = 2 \int_0^1 b(n, t) \sin(2\pi i t)\, dt = \frac{2}{\pi i} \sin\left(\frac{\pi i}{m}\right) \sin\left(\frac{\pi i}{m}(2n-1)\right) \qquad \begin{matrix} i = 1, \ldots, k \\ n = 1, \ldots, m \end{matrix} \quad (17)$$

Using Eqs. (14–17) the following expression holds

$$B(t) = R\Psi(t)$$

where R is the $m \times (1 + 2k)$ Fourier block-pulse conversion matrix given by

$$R = \begin{bmatrix} b_0 & b_{11} & \cdots & b_{ik} & b_{11}^* & \cdots & b_{1k}^* \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ b_0 & b_{m1} & \cdots & b_{mk} & b_{m1}^* & \cdots & b_{mk}^* \end{bmatrix}$$

and

$$\Psi(t) = [1 \quad \cos(2\pi t) \quad \cdots \quad \cos(2k\pi t) \quad \sin(2\pi t) \quad \cdots \quad \sin(2k\pi t)]^T$$

For example for $k = 5$ and $m = 6$, we get

$$R = \frac{1}{6}\begin{bmatrix} 1 & 1.654 & 0.827 & 0 & -0.414 & -0.331 \\ 1 & 0 & -1.654 & 0 & 0.827 & 0 \\ 1 & -1.654 & 0.827 & 0 & -0.414 & 0.331 \\ 1 & -1.654 & 0.827 & 0 & -0.414 & 0.331 \\ 1 & 0 & -1.654 & 0 & 0.827 & 0 \\ 1 & 1.654 & 0.827 & 0 & -0.414 & -0.331 \end{bmatrix}$$
$$\begin{bmatrix} 0.955 & 1.432 & 1.273 & 0.716 & 0.191 \\ 1.910 & 0 & -1.273 & 0 & 0.382 \\ 0.955 & -1.432 & 1.273 & -0.716 & 0.191 \\ -0.955 & 1.432 & -1.273 & 0.716 & -0.191 \\ -1.910 & 0 & 1.273 & 0 & -0.382 \\ -0.955 & -1.432 & -1.273 & -0.716 & -0.191 \end{bmatrix}$$

using Eq. (12), we get

$$A = R^T C$$

Application of Walsh Functions in Dynamic Systems, Identification, and Control

The initiation of the analysis of dynamic systems in the time domain via Walsh functions was made by Corrington in 1973 (32) in his paper on the solution of differential and integral equations. The key idea was the observation that successive integrals of Walsh functions are expressed as Walsh series with well-defined, tabulated coefficients. The differential equation is solved for the highest derivative, and the result is then integrated as many times as required to give the solution. Two years later, Chen and Hsiao (33) presented the solution of dynamic systems in state space formulation by a more systematic use of the Walsh function integration property expressed by an operational equation as

$$\int_0^t P(t)\,dt = EP(t)$$

where

$$P(t) = [\text{pal}(0,t) \quad \text{pal}(1,t) \quad \cdots \quad \text{pal}(n-1,t)]^T$$

and E is a well-defined operational matrix. Using this operational equation, the state-space differential system is converted to a linear algebraic system, which has to be solved for a set of unknown Walsh series coefficients. In what follows the operational matrix for Walsh functions will be briefly discussed.

Let us take pal(0, t), pal(1, t), . . ., pal(7, t) and integrate them; we will have various triangular waves (33). If we evaluate the Walsh coefficient for these triangular waves, we get the following matrix for $E_{(8\times 8)}$:

$$E_{(8\times 8)} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{4} & -\frac{1}{8} & 0 & -\frac{1}{16} & 0 & 0 & 0 \\ \frac{1}{4} & 0 & 0 & -\frac{1}{8} & 0 & -\frac{1}{16} & 0 & 0 \\ \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{16} & 0 \\ 0 & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{16} \\ \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 \end{bmatrix}$$

It is preferable to make the dimension of the matrix equal to 2^n, where n is an integer. Making this choice will enable us to obtain simpler results. It is noted that

$$\int_0^t \text{pal}(0,t)\,dt = t$$

therefore, the first row of E is the first four terms of Eq. (9). A general formula for $E_{(n\times n)}$ can be written as

$$E_{(n\times n)} = \left[\begin{array}{c|c|c} \frac{1}{2} & -\frac{2}{n}I_{n/8} & \\ \hline \frac{2}{n}I_{n/8} & 0_{n/8} & -\frac{1}{n}I_{n/4} \\ \hline & \frac{1}{n}I_{n/4} & 0_{n/4} \end{array}\right. \left.\begin{array}{c} \\ \\ -\frac{1}{2n}I_{n/2} \\ \\ \hline \\ 0_{n/2} \end{array}\right] \qquad (18)$$
$$\left[\begin{array}{c|c} & \\ & \\ & \\ \hline & \frac{1}{2n}I_{n/2} \end{array}\right.$$

It is interesting to note that if Eq. (18) is partitioned into four parts as shown, the upper left part of $E_{(n\times n)}$ is identical to

$$E_{(\frac{n}{2}\times\frac{n}{2})}$$

and the upper left corner of

$$E_{(\frac{n}{2}\times\frac{n}{2})} \qquad \text{is} \qquad E_{(\frac{n}{4}\times\frac{n}{4})}$$

Therefore, this regularity of the structure of the E matrix enables us to write the nth enlarged matrix to any dimension, if the dimension number is restricted to 2^n where n is an integer number.

Let us illustrate the application of operational matrix of integration by solving the following state equation

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad x(0) = x_0$$

where $x(t)$ is a state vector of n components and $u(t)$ is an input vector of l components. A and B are $n \times n$ and $n \times l$ matrices, respectively. We now solve the state equations via Walsh series.

First we assume the rate variable vector $\dot{x}(t)$ can be expressed as

$$\dot{x}(t) = [c_0 \quad c_1 \quad \cdots \quad c_{m-1}]P(t) = CP(t)$$

where each c_i, $i = 1, \ldots, m-1$, is an n vector. The state variable $x(t)$ may be obtained as

$$x(t) = C\int_0^t P(t)\,dt + x_0$$

Also the input vector can be expressed by Walsh series as

$$u(t) = HP(t)$$

where H is a $l \times m$ matrix. Thus we get

$$CP(t) = A(CEP(t) + x_0) + BHP(t)$$

Ax_0 can also be written as

$$Ax_0 = Ax_0P(t) = [Ax_0 \quad 0 \quad \cdots \quad 0]P(t) = GP(t)$$

Finally we have

$$C = ACE + G + BH$$

hence

$$C = ACE + K$$

where

$$G + BH = K$$

If we arrange the $n \times m$ matrix C as an nm vector c by changing its first column into the first n components of the vector; the second column, the second n components of the vector, and so on; and rearrange K in the same manner; we obtain

$$c = (A \otimes E^T)c + k \tag{19}$$

where $\otimes$ denote Kronecker product. Using Eq. (19), the solution of c is

$$c = [I - A \otimes E^T]^{-1}k$$

once c has been decided, the Walsh series representation for the rate variable is determined. The state variable vector is then found by substitution.

In addition to being applied to system analysis Walsh function expansions have also been applied with success to the design and implementation of optimal filter and controllers, naturally providing piecewise constant approximations of the optimal feedback gains. Previously, such approximations were determined by prespecifying the structural form of the time varying gains by Kleinman, Fortmann, and Athans in 1968 (34). The idea in using Walsh function series in optimal control problems was first employed by Chen and Hsiao (33). Essentially, the method belongs to the direct variational approach and is very powerful and easily implemenable. Tzafestas and Stavroulakis (35) used finite Walsh series expansion for designing approximate (suboptimal) observers and filters incorporated in a close-loop optimal controller. Both continuous and discrete time systems, time-invariant, and time-varying are considered. The solution provides a computational algorithm that gives the Walsh expansion coefficients of the state and observer output. Further, Chen and Hsiao applied Walsh functions (1) to solve the problems of linear systems by the state space model (36), (2) for time domain synthesis (37), (3) To solve the optimal control problem (38), (4) in the variational problem (39), and (5) for fractional calculus as applied to distributed systems (40).

Additionally, Walsh functions proved to be very powerful in solving the identification (or synthesis) problem of dynamic systems from given input-output records. The paper by Chen and Hsiao (37) appears to be the first work in which the problem of identifying dynamic systems is solved with the aid of Walsh functions. The key idea is the application of repeated integration together with the Walsh operational matrix employed for determining the system response. In Ref. 41, bilinear system identification is considered and solved by using the Walsh operational matrix and the group properties of Walsh functions. The same type of systems were also researched by Chen and Shih (42).

Rao and Palanisamy (43) provides a methodology for improving the identification accuracy of continuous systems through the so-called one-shot operational matrices for repeated integration via Walsh functions. Further, a multistep parameter estimation algorithm is given in Ref. 43 for systems with large, unknown time delays. Some additional works in the area of systems identification via Walsh functions are described in Rao and Sivakumar (44), Gopalsami and Deekshatulu (45), Tzafestas and Chrysochoides (46), and Tzafestas, Papastergoius, and Anoussis (47).

Moreover, Rao used Walsh function for (1) optimal control of time delay systems (48) (2) identification of time-lag systems (49) (3) transfer function matrix identification (50) (4) parameter estimation (25) (5) solving functional differential equations and related problems (51). Rao and Tzafestas (52) indicated the potentiality of Walsh and related functions in the area of systems and control in a review paper.

W. L. Chen defined a shift Walsh matrix for solving delay-differential equations (53) and used Walsh functions for parameter estimation of bilinear systems (42) as well as in the analysis of multidelay systems (54). Paraskevopoulos determined the transfer function of a single input single output (SISO) system from its impulse response with the help of Walsh functions and a fast Walsh algorithm (55). Tzafestas applied a Walsh series approach for lumped and distributed system identification (56). Mahapatra used Walsh functions for solving matrix Riccati equation arising in optimal control studies of linear diffusion equation (57). Mouldeens work was concerned with the application of Walsh spectral analysis of ordinary differential equations in a very formal as well as mathematical manner (58). Deb and Datta was the first to define Walsh operational transfer function for analysis of linear SISO systems (27,59) and Deb, Sen, and Datta (60) gave a review paper in Walsh functions and their applications in 1992.

The mathematical basis of Walsh function methods has become strong and versatile enough to encourage their application to the analysis of power-electronic circuits, and systems (31–61). From the study of different aspects of the Walsh

functions, we find the following properties suitable for application to the analysis power-electronic systems

1. Any member of the Walsh-function set resembles, to some extent, the typical switching pattern of a power-electronic converter. Hence, the voltage output of such a converter can be well represented by Walsh functions.
2. Walsh functions are defined in time domain. Thus, we do not need any inverse transformation as we do in Laplace domain analysis.
3. The set of Walsh functions is complete and orthonormal, thereby offering the facility for easier mathematical manipulations, including the design of fast computational algorithm.

Application of Walsh Functions in Different Areas of Science and Technology

Scientists have found that the binary nature of Walsh functions and its striking similarity to the familiar sine–cosine functions could adapt it for application in many areas of science and technology.

In the early 1960s, the first significant application of a Walsh function in the field of communications was noted. The credit goes mostly to Harmuth and his associates (62–64). Consequently, the Walsh functions were found to be an efficient tool in the field of signal multiplexing. Several experimental multiplexing systems were developed which made use of this nonsinusoidal technology.

In the multiplexing scheme, several independent signals are sent via a common communication channel. Walsh functions as carriers of communication signals are used in multiplexing schemes. While any set of complete orthogonal functions can be used as carriers, practical difficulties restrict the use of orthogonal functions to Walsh, block-pulse, and sinusoidal functions only. In time division multiplexing (TDM) block-pulses are generally used as carriers. Frequency division multiplexing (FDM) generally uses sinusoidal functions as carriers. However, in sequency division multiplexing (SDM) schemes Walsh functions are used as carriers.

One of the major advantages of using Walsh functions as carriers is that the multiplication process produces only one side-band and not two, as in the case with sinusoidal products. The reason is that the Walsh functions form a group under multiplication, that is, the product of two Walsh functions results in another Walsh function. The application of sequency digital processing technique covers a very wide field and includes various uses of spectral analysis. Some of these areas are, radar and sonar application, medical signal processing, speech processing, digital image processing and optics communications (65–68).

The processing of fixed or changing visual images by using digital techniques requires the manipulation of multidimensional signals involving operations on large numbers of data values. Since this process generally involves high speed online computer and substantial processing time, serious attempts have been made to find efficient alternative techniques. The sequency techniques (especially two-dimensional Walsh functions, with their emphasis on rapid computational ease) play significant roles in these development.

Considerable development in applying Walsh functions has been carried out by Hurst (69) and others during the 1970s. The techniques of domain analysis have also led to their use in the design of higher logic functions such as threshold logic gates (70). Here a subset of Walsh series, known as the Chow parameters (71), have proved particularly useful. In addition to these, application of Walsh functions has expanded to the formulation of multiinput gate structures (72), digital system fault diagnosis (73), digital circuit synthesis (74), and other related areas.

The widespread interest in practical applications of Walsh functions has stimulated further contributions to the mathematical theory. Of special interest is the logical differential calculus of Gibbs (75–76). In contrast to the sine-cosine functions, which often represent the characteristic solution to certain linear differential equations, the Walsh functions are shown to represent the solutions to what is known as the logical differential equations. Applications of Gibbs derivative are found in mathematical logic (77), approximation theory (78), statistics (79–80), and linear system theory (81).

SUMMARY

With the efforts of many researchers during the past twenty five years, the mathematical basis of Walsh functions methods has become strong and versatile. This basis encourages the application of these methods to the analysis of problems related to circuits, systems, and communications. When the analysis is carried out in the sequency domain instead of frequency domain, the method is straightforward, elegant, and compatible with easy computer manipulations.

BIBLIOGRAPHY

1. G. Sansone, *Orthogonal Functions,* New York: Interscience Publishers Inc., 1959.
2. K. G. Beauchamp, *Walsh Functions and Their Applications,* London: Academic Press, 1975.
3. J. B. Fourier, *The Analytic Theory of Heat,* 1878, New York: Reprinted by Dover Pub. Co. 1955.
4. A. Haar, Zur Theorie der orthogonalen funktionensysteme, *Math Annalen,* **69**: 331–371, 1910.
5. B. S. Nagy, *Introduction to Real Functions and Orthogonal Expansions,* New York: Oxford Univ. Press, 338–340, 1965.
6. F. Schmidt, Zur Theorie der Linearen und Nichtlinearen Integralgleichungen, *Math Annlaen,* **63**: 433–476, 1905.
7. H. Rademacher, Einige Satze von Allegemeinen Orthogonalfunktionen, *Math Annalen,* **87**: 122–138, 1922.
8. N. Ahmed, H. H. Schreiber, and P. Lopresti, On notation and definition of terms related to a class of complete orthogonal functions, *IEEE Trans. Electromagn. Compat.,* **15**: 75–80, 1973.
9. J. L. Walsh, A closed set of normal orthogonal functions, *Am. J. Math,* **45**: 5–24, 1923.
10. K. R. Rao, M. A. Narasimhan, and K. Revuluri, Image data processing by Hadamard-Haar transform, *IEEE Trans. Computers,* **C-24**: 888–896, 1975.
11. D. M. Huang, Walsh-Hadamard hybrid transforms, *IEEE Proc. 5th Int. Conf. Pattern Recognition,* 180–182, 1980.
12. B. J. Fino and V. R. Algazi, Slant-Haar transform, *Proc. IEEE,* **62**: 653–654, 1974.
13. H. F. Harmuth, *Transmission of Information by Orthogonal Functions,* 2nd Ed., Berlin: Springer-Verlag, 1972.
14. R. E. Paley, A remarkable set of orthogonal functions, *Proc. London Math Soc.,* **34**: 241–279, 1932.

15. K. W. Henderson, Comment on "computation of the fast Walsh-Fourier transform," *IEEE Trans. Comput.,* **C-19**: 850–851, 1970.
16. P. Lancaster, *Theory of Matrices,* Academic Press: New York, 1969.
17. C. K. Yuen, Walsh functions and the Gray code, *Proc. Applic. Walsh Functions,* Washington, D.C., AD727000, 68–73, 1971.
18. C. F. Chen, Y. T. Tsay, and T. T. Wu, Walsh operational matrices for functional calculus and their applications to distributed systems, *J. Franklin Inst.,* **303**: 267–284, 1977.
19. C. P. Rao and T. Srinivasan, Remarks on "Authors reply" to "comments on design of piecewise constant gains for optimal control via Walsh functions," *IEEE Trans. Autom. Control,* **AC-23**: 762–763, 1978.
20. C. P. Kwong and C. F. Chen, The convergence properties of block-pulse series, *Int. J. Syst. Sci.,* **12**: 745–751, 1981.
21. P. Sannuti, Analysis and synthesis of dynamic systems via block-pulse functions, *Proc. IEE,* **124**: 569–571, 1977.
22. O. N. Dalton, Further comments on "Design of piecewise constant gains for optimal control via Walsh functions," *IEEE Trans. Autom. Control,* **AC-23**: 760–762, 1978.
23. G. P. Rao and T. Srinivasan, Analysis and synthesis for dynamic systems containing time delay via block-pulse functions, *Proc. IEE,* **125**: 1064–1068, 1978.
24. H. F. Harmuth, Applications of Walsh functions in communications, *IEEE Spectrum,* **6**: 82–91, 1969.
25. G. P. Rao, Piecewise constant orthogonal functions and their application to systems and control, Berlin: Springer-Verlag, 1983.
26. S. G. Tzafestas, Global Walsh functions generator, Electron Eng., **48**: 45–50, 1976.
27. A. Deb and A. K. Datta, Analysis of pulse-fed power electronic circuits using Walsh function, *Int. J. Electronics,* **62**: 449–459, 1987.
28. K. H. Siemens and R. Kitai, A nonrecursive equation for the Fourier transform of a Walsh function, *IEEE Trans. Electromagn.,* Compatibility, **EMC-15** (2): 81–82, 1973.
29. N. M. Blachman, Sinusoids versus Walsh functions, *Proc. IEEE,* **62** (3): March 1974.
30. F. Swift and A. Kamberis, A new Walsh domain technique of harmonic elimination and voltage control in pulse-width modulated inventors, *IEEE Trans. Power Electron.,* **8**: April 1993.
31. M. Razzaghi and J. Nazarzadeh, Optimum pulse-width modulated patterns in induction motors using Walsh functions, *Electric Power Systems Research,* **35**: 87–91, 1995.
32. M. S. Corrington, Solution of differential and integral equations with Walsh functions, *IEEE Trans. Circuit Theory,* **CT-20**: 470–476, 1973.
33. C. F. Chen and C. H. Hsiao, Design of piecewise constant gains for optimal control via Walsh functions, *IEEE Trans. Autom. Control,* **AC-20**: 596–603, 1975.
34. D. L. Kelinman, T. Fortmann, and M. Athans, On the design of linear systems with piece-wise constant feedback gains, *IEEE Trans. Autom. Control,* **AC-13**: 354–361, 1968.
35. P. Stavroulakis and S. Tzafestas, Walsh series approach to observer and filter design in optimal control systems, *Int. J. Control,* **26**: 721–736, 1977.
36. C. F. Chen and C. H. Hsiao, A state approach to Walsh series solution of linear systems, *Int. J. Syst. Sci.,* **6**: 833–858, 1975.
37. C. F. Chen and C. H. Hsiao, Time domain synthesis via Walsh functions, *Proc. IEE,* **122**: 565–570, 1975.
38. C. F. Chen and C. H. Hsiao, Walsh series analysis in optimal control, *Int. J. Control,* **21**: 881–897, 1975.
39. C. F. Chen and C. H. Hsiao, A Walsh series direct method for solving variational problems, *J. Franklin Inst.* **300**: 265–280, 1975.
40. C. F. Chen, Y. Y. Tasy, and T. T. Wu, Walsh operational matrices for fractional calculus and their applications to distributed systems, *J. Franklin Inst.,* **303**: 267–284, 1977.
41. V. R. Karanam, P. A. Frick, and R. R. Mohler, Bilinear system identification by Walsh functions, *IEEE Trans. Autom. Control,* **Ac-23**: 709–713, 1978.
42. W. L. Chen and Y. P. Shih, Parameter estimation of bilinear systems via Walsh functions, *J. Franklin Inst.,* **305**: 249–257, 1978.
43. G. P. Rao and K. R. Palanisamy, Improved algorithms for parameter identification in continuous systems via Walsh functions, *Proc. IEE,* **130**: 6–19, 1983.
44. G. P. Rao and L. Sivakumar, Transfer function matrix identification in MIMO systems via Walsh functions, *Proc. IEEE,* **69**: 465–466, 1981.
45. N. Golpalsami and B. L. Deekshatulu, Time-domain synthesis via Walsh functions, *Proc. IEE,* **123**: 461–462, 1976.
46. S. C. Tazfestas and N. Chrysochoides, Time-varying reactivity reconstruction via Walsh functions, *IEEE Trans. Autom. Control,* **AC-22**: 886–888, 1977.
47. S. G. Tzafestas, C. N. Papastergiou, and J. N. Anoussis, Dynamic reactivity computation in Nuclear reactors using block-pulse function expansion, *Int. J. Modeling Simulation,* **4**: 73–76, 1984.
48. G. P. Rao and K. R. Palanisamy, Optimal control of time-delay systems via Walsh functions, *9th LFIP Conf. on Optimization Techniques,* Polish Academy of Sc. System Research Inst., Poland, Sept. 1979.
49. G. P. Rao and L. Sivakumar, Identification of time-lag systems via Walsh functions, *IEEE Trans. Autom. Control,* **AC-24**: 806–808, 1979.
50. G. P. Rao and L. Sivakumar, Transfer function matrix identification in MIMO systems via Walsh functions, *Proc. IEEE,* **69**: 465–466, 1981.
51. G. P. Rao and K. R. Palanisamy, A new operational-matrix for delay via Walsh functions and some aspects of its algebra and applications, *National Systerms Conf.* **NSC-78**: Pau Ludhiana (India), Sept, 1978.
52. G. P. Rao and S. G. Tzafestas, A decade of piecewise constant orthogonal functions in systems and control, *Math Computation and Simulation,* **27**: 389–407, 1985.
53. W. L. Chen and Y. P. Shih, Shift Walsh matrix and delay differential equations, *IEEE Trans. Autom. Control,* **AC-23**: 1023–1028, 1978.
54. W. L. Chen, Walsh series analysis of multi-delay systems, *J. Franklin Inst.,* **313**: 207–217, 1982.
55. P. N. Paraskevopoulos and S. J. Varoufakis, Transfer function determination from impulse response via Walsh functions, *Int. J. Circuit Theory Appl.,* **8**: 85–89, 1980.
56. S. G. Tzafestas, Walsh Series approach to lumped and distributed system identification, *J. Franklin Inst.,* **305**: 199–220, 1978.
57. G. B. Mahapatra, Solution of optimal control problem of linear diffusion equation via Walsh functions, *IEEE Trans. Autom. Control,* **Ac-25**: 319–321, 1980.
58. T. H. Mouldon and M. A. Scott, Walsh spectral analysis for ordinary differential equations, Part 1—Initial value 6 problems, *IEEE Trans. Circuits Syst.,* **CAS-35**, 742–745, 1988.
59. A. Deb and K. Datta, Analysis of continuously variable pulse-wide modulated system via Walsh functions, *Int. J. Syst. Sci.* **23**: 151–166, 1992.
60. A. Deb, S. K. Sen, and A. Datta, Walsh functions and their applications: a review, *IETE Tech. Rev.,* **9**: 238–252, 1992.
61. J. Nazarzadeh, M. Razzaghi, and K. Y. Nikravesh, Harmonic elimination in pulse-width modulated inverters using piecewise constant orthogonal functions, *Electric Power Systems Research,* **40**: 45–49, 1997.

62. H. F. Harmuth, On the transmission of information by orthogonal time functions, *Trans. AIEE Comm. Electronic,* **79**: 248–255, 1960.

63. H. F. Harmuth, Tragersystem Fur die Nachrichtentechnik, W. German Patent 1-191-416, HS0239 (U.S. Patent 3470324), 1963.

64. H. F. Harmuth, *Nonsinusoidal Wave for Radar and Radio Communication,* New York: Academic Press, 1981.

65. M. K. Srirama et al., Waveform synthesis using Walsh functions, *Int. J. Electronics,* **74**: 6: 857–869, 1993.

66. B. J. Falkowski, Recursive relationships, fast transforms, generations and VLSI iterative architecture for Gray code ordered Walsh functions, *IEE Proc. Comput. Digital Techniques,* **142** (5): 325–331, 1995.

67. E. Macii and M. Poncino, Using symbolic Rademacher-Walsh spectral transforms to evaluate the agreement between Boolean functions, *IEE Proc. Comput. Digital Techniques,* **143** (1): 64–68, 1996.

68. V. V. Orlov and A. R. Bulygin, Volume holograms of co-orthogonal waves for optical channel switching and expansion of light waves in Walsh basis functions, *Optics Communications 133,* 415–433, 1977.

69. S. L. Hurst, *Logical Processing of Digital Signals,* New York: Crane Russak, London: Edward Arnold, 1978.

70. C. R. Edwards, The application of the Rademacher-Walsh transform to Boolean function classification and threshold logic synthesis, *IEEE Trans. Computers,* **C-24**: 48–62, 1975.

71. C. K. Chow, On the characterization of threshold functions, *Proc. IEEE Symp. Switching Theory and Logic Design,* **S-314**: 34–38, 1961.

72. C. R. Edwards, A special class of universal logic gates and their evaluation under a Walsh transform, *Int. J. Electronics,* **44**: 49–59, 1978.

73. R. G. Bennetis and S. L. Hurst, Rademacher-Walsh spectral transform: a new tool for problem in digital network fault diagnosis?, *IEE Proc. Comput. Digital Techniques,* **1**: 38–44, 1978.

74. C. R. Edwards, The application of the Rademacher-Walsh transform to digital circuit synthesis, *Proc. Theory and Application of Walsh Functions,* Hatfield Polytechnic, England, 1973.

75. J. E. Gibbs and M. J. Millard, Walsh functions as solutions of a logical differential equation, National Physical Laboratory, Teddinton, England, DES Reports No. 1 and 2, 1969.

76. J. E. Gibbs, Sine waves and Walsh waves in physics, *Proc. Symp. on Application of Walsh Functions,* Washington DC, 260–274, 1970.

77. P. Liedi, Harmonische Analysis bei Augssagenkallkuelen, Math Logik, **13**: 158–167, 1970.

78. P. L. Butzer and H. J. Wagner, Walsh-Fourier series and the concept of a derivative, in *Applicable Analysis,* Vol. 1, London: Gordon and Breach, 1972.

79. J. Pearl, Applications of Walsh transform to statistical analysis, *IEEE Trans. Syst. Man. Cybern.,* **SMC-1**: 111–119, 1971.

80. K. K. Nambiar, Approximation and representation of joint probability distribution of binary random variables by Walsh Range functions, *Proc. Symp. Appl. of Walsh Functions,* Washington DC, 70–72, 1970.

81. P. Pichler, Some aspect of a theory of correlation with respect to Walsh harmonics analysis, Maryland University, Report No. AD 714 596, 1970.

MOHSEN RAZZAGHI
Mississippi State University

JALAL NAZARZADEH
Amirkabir University

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Wavelet Methods for Solving Integral and Differential Equations

Standard Article

Jaideva C. Goswami[1], Richard E. Miller[2], Robert D. Nevels[3]

[1]Sugar Land Product Center, Sugar Land, TX

[2]Texas A&M University, College Station, TX

[3]Texas A&M University, College Station, TX

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2470

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (208K)

Abstract

The sections in this article are

Wavelet Preliminaries

Integral Equations

Matrix Equation Generation

Intervallic Wavelets

Numerical Results

Semiorthogonal Versus Orthogonal Wavelets

Differential Equations

| | |

Copyright © 1999-2008 All Rights Reserved.

WAVELET METHODS FOR SOLVING INTEGRAL AND DIFFERENTIAL EQUATIONS

Many of the phenomena studied in electrical engineering and physics can be described mathematically by second-order partial differential equations (PDEs). Some examples of PDEs are the Laplace, Poisson, Helmholtz, and Schrödinger equations. Each of these equations may be solved analytically in some cases, but not for all cases of interest. These PDEs can often be converted to integral equations. One of the attractive features of integral equations is that boundary conditions are built in and, therefore, do not have to be applied externally (1). Mathematical questions of existence and uniqueness of a solution may be handled with greater ease with the integral form.

Either approach, differential or integral equations, used to represent a physical phenomenon can be viewed in terms of an operator operating on an unknown function in order to produce a known function. In this article we will deal with the linear operator. The linear operator equation is converted to a system of linear equations with the help of a complete set of basis functions which are then solved for the unknown coefficients. The finite-element and finite-difference techniques used to solve PDEs result in sparse and banded matrices, whereas integral equations almost always lead to a dense matrix; an exception is the case when the basis functions, chosen to represent the unknown functions, happen to be the eigenfunctions of the operator.

With the advent of wavelets in the 1980s (although they were known in one form or the other since the beginning of this century), numerical analysts have been presented with a new class of "local" basis functions at their disposal which can significantly improve existing methods. Two of the main properties of wavelets vis-à-vis boundary value problems are their hierarchical nature and the vanishing moments properties. Because of their hierarchical (multiresolution) nature, wavelets at different resolutions are interrelated, a property that makes them suitable candidates for multigrid-type methods in solving PDEs. On the other hand, the vanishing moment property by virtue of which wavelets, when integrated against a function of certain order, make the integral zero, is attractive in sparsifying a dense matrix generated by an integral equation.

In the next section, we provide some definitions and properties of wavelets that are relevant to understanding the materials presented in this article. A complete exposition of the application of wavelets to integral and differential equation is beyond the scope of this article. Our objective is to provide the reader with some preliminary theory and results on the application of wavelets to boundary value problems and give references where more details may be found. Since we most often encounter integral equations in electrical engineering problems, we will emphasize their solutions using wavelets. We give a few examples of commonly occurring integral equations. The first and the most important step in solving integral equations is to transform them into a set of linear equations. Both conventional and wavelet-based methods in generating matrix equations are discussed. Some numerical results are presented which illustrate the advantages of the wavelet-based technique. We also discuss wavelets on the bounded interval. Some of the techniques applied to solving integral equations are useful for differential equations as well. At the end of this article we briefly describe the applications of wavelets in PDEs and provide references where readers can find further information.

WAVELET PRELIMINARIES

In this section we briefly describe the basics of wavelet theory to facilitate further discussion in this article. More details on multiresolution and other properties of wavelets may be found elsewhere in this encyclopedia. Readers may also refer to Refs. 2–10.

As pointed out before, multiresolution analysis (MRA) plays an important role in the application of wavelets to boundary value problems. In order to achieve MRA, we must have a finite-energy function (square integrable on the real line) $\phi(x) \in L^2(\mathbb{R})$, called a scaling function, that generates a nested sequence of subspaces

$$\{0\} \leftarrow \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots \rightarrow L^2 \tag{1}$$

and satisfies the dilation (refinement) equation, namely,

$$\phi(x) = \sum_k p_k \phi(2x - k) \tag{2}$$

with $\{p_k\}$ belonging to the set of square summable bi-infinite sequences. The number 2 in Eq. (2) signifies "octave levels." In fact this number could be any rational number, but we will discuss only octave levels or scales. From Eq. (2) we see that the function $\phi(x)$ is obtained as a linear combination of a scaled and translated version of itself, and hence the name scaling function.

The subspaces V_j are generated by $\phi_{j,k}(x) := 2^{j/2}\phi(2^j x - k)$, $j, k \in \mathbb{Z}$, where $\mathbb{Z} := \{. \ . \ ., -1, 0, 1, \ . \ . \ .\}$. For each scale j, since $V_j \subset V_{j+1}$, there exists a complementary subspace W_j of V_j in V_{j+1}. This subspace W_j is called "wavelet subspace" and is generated by $\psi_{j,k}(x) := 2^{j/2}\psi(2^j x - k)$, where $\psi \in L^2$ is called the "wavelet." From the above discussion, these results follow easily:

$$\begin{cases} V_{j_1} \cap V_{j_2} = V_{j_1}, & j_1 > j_2 \\ W_{j_1} \cap W_{j_2} = \{0\}, & j_1 \neq j_2 \\ V_{j_1} \cap W_{j_2} = \{0\}, & j_1 \leq j_2 \end{cases} \tag{3}$$

The scaling function ϕ exhibits low-pass filter characteristics in the sense that $\hat{\phi}(0) = 1$, where a hat over the function denotes its Fourier transform. On the other hand, the wavelet function ψ exhibits bandpass filter characteristic in the sense that $\hat{\psi}(0) = 0$. Some of the important properties that we will use in this article are given below:

- *Vanishing Moment.* A wavelet is said to have a vanishing moment of order m if

$$\int_{-\infty}^{\infty} x^p \psi(x)\, dx = 0, \qquad p = 0, \ldots, m-1 \tag{4}$$

All wavelets must satisfy the above condition for $p = 0$.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

- *Orthonormality.* The wavelets $\{\psi_{j,k}\}$ form an orthonomal (o.n.) basis if

$$\langle \psi_{j,k}, \psi_{l,m} \rangle = \delta_{j,l}\delta_{k,m} \qquad \text{for all } j,k,l,m \in \mathbb{Z} \tag{5}$$

where $\delta_{p,q}$ is the Kronecker δ defined in the usual way as

$$\delta_{p,q} = \begin{cases} 1, & p = q; \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

The inner product $\langle f_1, f_2 \rangle$ of two square integrable functions f_1 and f_2 is defined as

$$\langle f_1, f_2 \rangle := \int_{-\infty}^{\infty} f_1(x) f_2^*(x)\, dx$$

with $f_2^*(x)$ representing the complex conjugation of f_2.

- *Semiorthogonality.* The wavelets $\{\psi_{j,k}\}$ form a semiorthogonal (s.o.) basis if

$$\langle \psi_{j,k}, \psi_{l,m} \rangle = 0; j \neq l \qquad \text{for all } j,k,l,m \in \mathbb{Z} \tag{7}$$

Given a function $f(x) \in L^2$, the decomposition into various scales begins by mapping the function into a sufficiently high-resolution subspace V_M, that is,

$$L^2 \ni f(x) \mapsto f_M = \sum_k a_{M,k}\phi(2^M t - k) \in V_M \tag{8}$$

Now since

$$\begin{aligned} V_M &= W_{M-1} + V_{M-1} \\ &= W_{M-1} + W_{M-2} + V_{M-2} \\ &= \sum_{n=1}^{N} W_{M-n} + V_{M-N} \end{aligned} \tag{9}$$

we can write

$$f_M(x) = \sum_{n=1}^{N} g_{M-n}(x) + f_{M-N}(x) \tag{10}$$

where $f_{M-N}(x)$ is the coarsest approximation of $f_M(t)$ and

$$f_j(x) = \sum_k a_{j,k}\phi(2^j t - k) \in V_j \tag{11}$$

$$g_j(x) = \sum_k w_{j,k}\psi(2^j t - k) \in W_j \tag{12}$$

If the scaling functions and wavelets are orthonormal, it is easy to obtain the coefficients $\{a_{j,k}\}$ and $\{w_{j,k}\}$. However, for the s.o. case, we need a dual scaling function ($\tilde{\phi}$) and dual wavelet ($\tilde{\psi}$). Dual wavelets satisfy the "biorthogonality condition," namely,

$$\langle \psi_{j,k}, \tilde{\psi}_{l,m} \rangle = \delta_{j,l} \cdot \delta_{k,m}, j,k,l,m \in \mathbb{Z} \tag{13}$$

For the s.o. case, both ψ and $\tilde{\psi}$ belong to the same space W_j for an appropriate j; likewise ϕ and $\tilde{\phi}$ belong to V_j. One of the difficulties with s.o. wavelets is that their duals do not have compact support. We can achieve compact support for both $\tilde{\phi}$ and $\tilde{\psi}$ if we forgo the orthogonality requirement that $V_j \perp W_j$. In such a case we get "biorthogonal wavelets" (11) and two MRAs, $\{V_j\}$ and $\{\tilde{V}_j\}$. In this article we will discuss application of o.n. and s.o. wavelets only.

INTEGRAL EQUATIONS

Integral equations appear frequently in practice, particularly the first-kind integral equations (12) in inverse problems. These equations can be represented as

$$L_K f = \int_a^b f(x')K(x,x')\, dx' = g(x) \tag{14}$$

where $f(x)$ is an unknown function, $K(x, x')$ is the known kernel which might be the system impulse response or Green's function, and $g(x)$ is the known response functions.

For instance, the electric surface current J_{sz} on an infinitely long metallic cylinder illuminated by an electromagnetic plane wave that is transverse magnetic (TM) to the z direction, as shown in Fig. 1, is related to the incident electric field via an integral equation

$$j\omega\mu_0 \int_C J_{sz}(l')G(l,l')\, dl' = E_z^i(l) \tag{15}$$

where

$$G(l,l') = \frac{1}{4j}H_0^{(2)}(k_0|\boldsymbol{\rho}(l) - \boldsymbol{\rho}(l')|) \tag{16}$$

with the wavenumber, $k_0 = 2\pi/\lambda_0$. The electric field, E_z^i, is the z component of the incident electric field and $H_0^{(2)}$ is the second-kind Hankel function of order 0, and λ_0 is the wavelength in free space. Here, the contour of integration has been parameterized with respect to the chord length. The field component E_z^i can be expressed as

$$E_z^i(l) = E_0 \exp[jk_0(x(l)\cos\phi_i + y(l)\sin\phi_i)] \tag{17}$$

where ϕ_i is the angle of incidence.

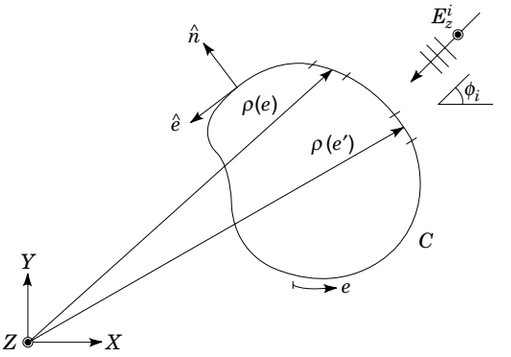


Figure 1. Cross section of an infinitely long metallic cylinder illuminated by a TM plane wave.

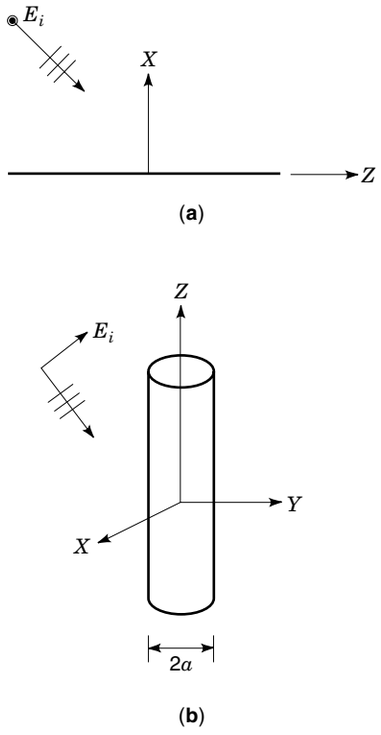


Figure 2. (a) A thin half-wavelength long metallic strip illuminated by a TM wave. (b) A thin wire of length $\lambda/2$ and thickness $\lambda/1000$ illuminated by a plane wave.

Scattering from a thin perfectly conducting strip, as shown in Fig. 2(a), gives rise to an equation similar to Eq. (15). For this case, we have

$$\int_{-h}^{n} J_{sy}(z')G(z,z')\,dz' = E_y^i(z) \tag{18}$$

where $G(z, z')$ is given by Eq. (16).

As a final example of the scattering problem, consider scattering from a thin wire as shown in Fig. 2(b). Here the current on the wire and the incident field are related to each other as

$$\int_{-l}^{l} I(z')K_w(z,z')\,dz' = -E^i(z) \tag{19}$$

where the kernel K_w is given by

$$K_w(z,z') = \frac{1}{4\pi j\omega\epsilon_0}\frac{\exp(-jk_0R)}{R^5} \times [(1+jk_0R)\times(2R^2-3a^2)+k_0^2a^2R^2] \tag{20}$$

$$E^i(z) = E_0\sin\theta\exp(jk_0z\cos\theta) \tag{21}$$

This kernel is obtained by interchanging integration and differentiation in the integrodifferential form of Pocklington's equation and by using the reduced kernel distance $R = [a^2 + (z - z')^2]^{1/2}$, where a is the radius of the wire (13).

All of the equations described thus far have the form of a first-kind integral equation, namely,

$$\int_a^b f(x')K(x,x')\,dx' = g(x) \tag{22}$$

where f is the unknown function and the kernel K and the function g are known. Here the objective is to reconstruct the function f from a set of known data (possibly measured) g. The kernel K may be thought of as the impulse response function of the system.

Although we discuss the solution technique for first-kind integral equations only, the method can be easily extended to second-kind equations (14,15) and higher-dimensional integral equations (16).

MATRIX EQUATION GENERATION

As mentioned in the previous section, the first step in solving any integral or differential equation is to convert these into a matrix equation which is then solved for the unknown coefficients which are subsequently used to construct the unknown function.

The goal is to transform Eq. (14) to a matrix equation:

$$Zi = v \tag{23}$$

where Z is a two-dimensional matrix, sometimes referred to as impedance matrix, i is the column vector of unknown coefficients, and v is another column vector related to g. Computation time depends largely upon the way we obtain and solve Eq. (23). In the following section we describe conventional and wavelet basis functions that are used to represent the unknown function.

Conventional Basis Functions

The unknown function $f(x)$ can be written as

$$f(x) = \sum_n i_n b_n(x) \tag{24}$$

where $\{b_n\}$ form a complete set of basis functions. These bases may be "global" (entire domain), extending the entire length $[a, b]$, or they may be "local" (subdomain), covering only a small segment of the interval, or a combination of both. Some of the commonly used subdomain basis functions are shown in Fig. 3.

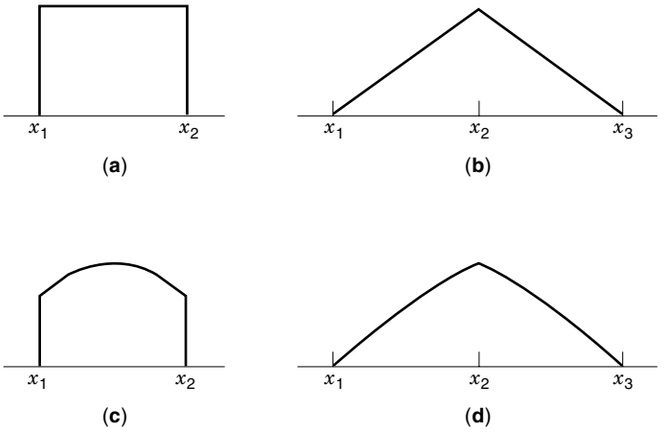


Figure 3. Typical subdomain basis functions: (a) piecewise constant, (b) piecewise linear, (c) piecewise cosine, and (d) piecewise sine functions.

For an exact representation of $f(x)$ we may need an infinite number of terms in the above series. However, in practice, a finite number of terms suffices for a given acceptable error. Substituting the series representation of $f(x)$ into the original Eq. (14), we get

$$\sum_{n=1}^{N} i_n L_K b_n \approx g \tag{25}$$

For the present discussion we will assume N to be large enough so that the above representation is exact. Now by taking the inner product of Eq. (25) with a set of *weighting functions* or *testing functions* $\{t_m: m = 1, \ldots, M\}$ we obtain a set of linear equations

$$\sum_{n=1}^{N} i_n \langle t_m, L_K b_n \rangle = \langle t_m, g \rangle, \qquad m = 1, \ldots, M \tag{26}$$

which can be written in the matrix form as

$$[Z_{mn}][i_n] = [v_m] \tag{27}$$

where

$$Z_{mn} = \langle t_m, L_K b_n \rangle, \qquad m = 1, \ldots, M, \quad n = 1, \ldots, N$$
$$v_m = \langle t_m, g \rangle, \qquad m = 1, \ldots, M$$

The solution of the matrix equation gives the coefficients $\{i_n\}$ and thereby the solution of the integral equations. Two main choices of the testing functions are (1) $t_m(x) = \delta(x - x_m)$, where x_m is a discretization point in the domain, and (2) $t_m(x) = b_m(x)$. In the former case the method is called *point matching,* whereas the latter method is known as *Galerkin method.* The method so described and those to be discussed in the following sections are generally referred to as "method of moments" (MoM) (17). We will refer to MoM with conventional bases as "conventional MoM" while the method with wavelet bases will be called "wavelet MoM." Observe that the operator L_K in the preceding paragraphs could be any linear operator—differential as well as integral.

Wavelet Bases

Conventional bases (local or global), when applied directly to the integral equations, generally lead to a dense (fully populated) matrix Z. As a result, the inversion and the final solution of such a system of linear equations are very time-consuming. In later sections it will be clear why conventional bases give a dense matrix while wavelet bases produce sparse matrices. Observe that conventional MoM is a single-level approximation of the unknown function in the sense that the domain of the function ($[a, b]$, for instance) is discretized only once, even if we use nonuniform discretization of the domain. Wavelet MoM, on the other hand, is inherently multilevel in nature, as we will discuss later.

Beylkin et al. (18) first proposed the use of wavelets in sparsifying an integral equation. Alpert et al. (14) used "wavelet-like" basis functions to solve second-kind integral equations. In electrical engineering, wavelets have been used to solve integral equations arising from electromagnetic scattering and transmission line problems (16,19–33). In what follows we briefly describe four different ways in which wavelets have been used in solving integral equations.

Use of Fast Wavelet Algorithm. In this method, the impedance matrix Z is obtained via the conventional method of moments using basis functions such as triangular functions, and then wavelets are used to transform this matrix into a sparse matrix (19,20). Consider a matrix W formed by wavelets. This matrix comprises the decomposition and reconstruction sequences and their translates. We have not discussed these sequences here, but readers may find these sequences in any standard book on wavelets (2–10), for example.

The transformation of the original MoM impedance matrix into the new wavelet basis is obtained as

$$WZW^{\mathrm{T}} \cdot (W^{\mathrm{T}})^{-1} i = Wv \tag{28}$$

which can be written as

$$Z_{\mathrm{w}} \cdot i_{\mathrm{w}} = v_{\mathrm{w}} \tag{29}$$

where W^{T} represents the transpose of the matrix W. The new set of wavelet transformed linear equations are

$$Z_{\mathrm{w}} = WZW^{\mathrm{T}} \tag{30}$$

$$i_{\mathrm{w}} = (W^{\mathrm{T}})^{-1} i \tag{31}$$

$$v_{\mathrm{w}} = Wv \tag{32}$$

The solution vector i is then given by

$$i = W^{\mathrm{T}} (WZW^{\mathrm{T}})^{-1} Wv \tag{33}$$

For orthonormal wavelets $W^{\mathrm{T}} = W^{-1}$ and the transformation (28) is "unitary similar." It has been shown in Refs. 19 and 20 that the impedance matrix Z_{w} is sparse, which reduces the inversion time significantly. Discrete wavelet transform (DWT) algorithms can be used to obtain Z_{w}. Readers may find the details of discrete wavelet transform (octave scale transform) elsewhere in this encyclopedia or in any standard book on wavelets. Sometimes it becomes necessary to compute the wavelet transform at nonoctave scales. Readers are referred to Refs. 34–36 for details of such algorithm.

Direct Application of Wavelets. In another method of applying wavelets to integral equations, wavelets are directly applied; that is, first the unknown function is represented as a superposition of wavelets at several levels (scales) along with the scaling function at the lowest level, prior to using Galerkin's method described before.

In terms of wavelets and scaling functions we can write the unknown function f in Eq. (14) as

$$f(x) = \sum_{j=j_0}^{j_u} \sum_{k=K_1}^{K(j)} w_{j,k} \psi_{j,k}(x) + \sum_{k=K_1}^{K(j_0)} a_{j_0,k} \phi_{j_0,k}(x) \tag{34}$$

where we have used the multiscale property, Eq. (10).

It should be pointed out here that the wavelets $\{\psi_{j,k}\}$ by themselves form a complete set; therefore, the unknown func-

tion could be expanded entirely in terms of the wavelets. However, to retain only a finite number of terms in the expansion, the scaling function part of Eq. (34) must be included. In other words, $\{\psi_{j,k}\}$, because of their bandpass filter characteristics, extract successively lower and lower frequency components of the unknown function with decreasing values of the scale parameter j, while $\phi_{j_0,k}$, because of its lowpass filter characteristics, retains the lowest frequency components or the coarsest approximation of the original function.

In Eq. (34), the choice of j_0 is restricted by the order of the wavelet, while the choice of j_u is governed by the physics of the problem. In applications involving electromagnetic scattering, as a "rule of thumb" the highest scale, j_u, should be chosen such that $1/2^{j_u+1}$ does not exceed $0.1\lambda_0$, with λ_0 being the operative wavelength.

When Eq. (34) is substituted in Eq. (14), and the resultant equation is tested with the same set of expansion functions, we get a set of linear equations

$$\begin{bmatrix} [Z_{\phi,\phi}] & [Z_{\phi,\psi}] \\ [Z_{\psi,\phi}] & [Z_{\psi,\psi}] \end{bmatrix} \begin{bmatrix} [a_{j_0,k}]_k \\ [w_{j,n}]_{j,n} \end{bmatrix} = \begin{bmatrix} \langle v, \phi_{j_0,k'} \rangle_{k'} \\ \langle v, \psi_{j',k'} \rangle_{j',k'} \end{bmatrix} \tag{35}$$

where the ψ term of the expansion function and the ϕ term of the testing function give rise to the $[Z_{\phi,\psi}]$ portion of the matrix Z. Similar interpretation holds for $[Z_{\phi,\phi}]$, $[Z_{\psi,\phi}]$, and $[Z_{\psi,\psi}]$.

By carefully observing the nature of the submatrices, we can explain the "denseness" of the conventional MoM and the "sparseness" of the wavelet MoM. Unlike wavelets, the scaling functions discussed in this article do not possess the vanishing moments properties. Consequently, for two pulse or triangular functions ϕ_1 and ϕ_2 (usual bases for the conventional MoM and suitable candidates for the scaling functions), even though $\langle \phi_1, \phi_2 \rangle = 0$ for nonoverlapping support, $\langle \phi_1, L_K \phi_2 \rangle$ is not very small since $L_k \phi_2 \rangle$ is not small. On the other hand, as is clear from the vanishing moment property [Eq. (4)] of a wavelet of order m, the integral vanishes if the function against which the wavelet is being integrated behaves as a polynomial of a certain order "locally." Away from the singular points the kernel has a polynomial behavior locally. Consequently, integrals such as $(L_K \psi_{j,n})$ and the inner products involving wavelets are very small for nonoverlapping support.

Because of its "total positivity" property (5, pp. 207–209), the scaling function has a "smoothing" or "variation diminishing" effect on a function against which it is integrated. The smoothing effect can be understood as follows. If we convolve two pulse functions, both of which are discontinuous but totally positive, the resultant function is a linear B-spline (triangular function) which is continuous. Likewise, if we convolve two linear B-splines, we get a cubic B-spline which is twice continuously differentiable. Analogous to these, the function $L_K \phi_{j_0,k}$ is smoother than the kernel K itself. Furthermore, because of the MRA properties that give

$$\langle \phi_{j,k}, \psi_{j',l} \rangle = 0, \qquad j \le j' \tag{36}$$

the integrals $\langle \phi_{j_0,k'}, (L_K \psi_{j,n}) \rangle$ and $\langle \psi_{j',n'}, (L_K \phi_{j_0,k}) \rangle$ are quite small.

The $[Z_{\phi,\phi}]$ portion of the matrix, although diagonally dominant, usually does not have entries which are very small compared to the diagonal entries. In the conventional MoM case, all the elements of the matrix are of the form $\langle \phi_{j,k'}, (L_K \phi_{j,k}) \rangle$. Consequently, we cannot threshold such a matrix in order to sparsify it. In the wavelet MoM case, the entries of $[Z_{\phi,\phi}]$ occupy a very small portion (5×5 for linear and 11×11 for cubic spline cases) of the matrix, while the rest contain entries whose magnitudes are very small compared to the largest entry; hence a significant number of entries can be set to zero without affecting the solution appreciably.

Wavelets in Spectral Domain. In the previous section, we have used wavelets in the space domain. The local support and vanishing moment properties of wavelet bases were used to obtain a sparse matrix representation of an integral equation. In some applications, particularly in spectral domain methods in electromagnetics, wavelets in the spectral domain may be quite useful. Whenever we have a problem in which the unknown function is expanded in terms of the basis function in the space (time) domain while the numerical computation takes place in the spectral (frequency) domain, we should look at the space-spectral window product in order to determine the efficiency of using a particular basis function. According to the "uncertainty principle," the space-spectral window product of a square integrable function cannot be less than 0.5; the lowest value is possible only for functions of Gaussian class. Because of the nearly optimal space-spectral window product of the cubic spline and the corresponding semiorthogonal wavelet, the improper integrals appearing in many spectral domain formulations of integral equations can be evaluated efficiently. This is due to the fact that higher-order wavelets generally have faster decay in the spectral domain. The spectral domain wavelets have been used to solve the transmission line discontinuity problem in Ref. 16.

Wavelet Packets. The discrete wavelet packet (DWP) similarity transformations have been used to obtain a higher degree of sparsification of the matrix than is achievable using the standard wavelets (31). It has also been shown that the DWP method gives faster matrix–vector multiplication than some of the fast multipole methods.

In the standard wavelet decomposition process, first we map the given function to a sufficiently high resolution subspace (V_M) and obtain the approximation coefficients $\{a_{M,k}\}$ (see section entitled "Wavelet Preliminaries"). The approximation coefficients $\{a_{M-1,k}\}$ and wavelet coefficients $\{w_{M-1,k}\}$ are computed from $\{a_{M,k}\}$. This process continues; that is, the coefficients for the next lower level $M - 2$ are obtained from $\{a_{M-1,k}\}$, and so on. Observe that in this scheme, only approximation coefficients $\{a_{j,k}\}$ are processed at any scale j; the wavelet coefficients are merely the outputs and remain untouched. In a wavelet packet, the wavelet coefficients are also processed which, heuristically, should result in higher degree of sparsity since in this scheme the frequency bands are further divided compared with the standard decomposition scheme.

INTERVALLIC WAVELETS

Wavelets on the real line have been used to solve integral equations arising from electromagnetic scattering and waveguiding problems. The difficulty with using wavelets on the entire real line is that the boundary conditions need to be enforced explicitly. Some of the scaling functions and wavelets must be placed outside the domain of integration. Furthermore, because of truncation at the boundary, the van-

ishing moment property is not satisfied near the boundary. Also, in signal processing, uses of these wavelets lead to undesirable jumps near the boundaries. We can avoid this difficulty by periodizing the scaling function as (4, Sec. 9.3)

$$\phi_{j,k}^{p} := \sum_{l} \phi_{j,k}(x+l) \tag{37}$$

where the superscript p implies periodic case. Periodic wavelets are obtained in a similar way. It is easy to show that if $\hat{\phi}(2\pi k) = \delta_{k,0}$, which is generally true for the scaling functions, then $\Sigma_k\ \phi(x-k) \equiv 1$. If we apply the last relation (which is also known as the "partition of unity") to Eq. (37), we can show that $\{\phi_{0,0}^{p}\} \cup \{\psi_{j,k}^{p}; j \in \mathbb{Z}^{+} := \{0, 1, 2, \ldots\}, k = 0, \ldots, 2^{j}-1\}$ generates $L^2([0, 1])$.

Periodic wavelets have been used by Refs. 28–30. However, as mentioned in Ref. 4, Sec. 10.7, unless the function which is being approximated by the periodized scaling functions and wavelets has the same values at the boundaries, we still have "edge" problems at the boundaries. To circumvent these difficulties, wavelets, constructed especially for a bounded interval, has been introduced in Ref. 33. Details on intervallic wavelets may be found in Refs. 33 and 37–39. Most of the time, we are interested in knowing the formulas for these wavelets rather than delving into the mathematical rigor of their construction. These formulas may be found in Refs. 10 and 33.

Wavelets on a bounded interval satisfy all the properties of regular wavelets that are defined on an entire real line; the only difference is that in the former case, there are a few special wavelets near the boundaries. Wavelets and scaling functions whose support lies completely inside the interval have properties that are exactly the same as those of regular wavelets. As an example, consider semiorthogonal wavelets of order m. For this case the scaling functions (B-splines of order m) have support $[0, m]$, whereas the corresponding wavelet extends the interval $[0, 2m-1]$. If we normalize the domain of the unknown function from $[a, b]$ to $[0, 1]$, then there will be 2^j segments at any scale j (discretization step $= 2^{-j}$). Consequently, in order to have at least one complete inner wavelet, the following condition must be satisfied:

$$2^{j} \geq 2m-1 \tag{38}$$

For j satisfying the above condition, there are $m-1$ boundary scaling functions and wavelets at 0 and 1, and $2^j - m + 1$ inner scaling functions and $2^j - 2m + 2$ inner wavelets. Figure 4 shows all the scaling functions and wavelets for $m = 2$ at the scale $j = 2$. All the scaling functions for $m = 4$ and $j = 3$ are shown in Fig. 5(a), while Fig. 5(b) gives only the corresponding boundary wavelets near $x = 0$ and one inner wavelet. The rest of the inner wavelets can be obtained by simply translating the first one, whereas the boundary wavelets near $x = 1$ are the mirror images of ones near $x = 0$.

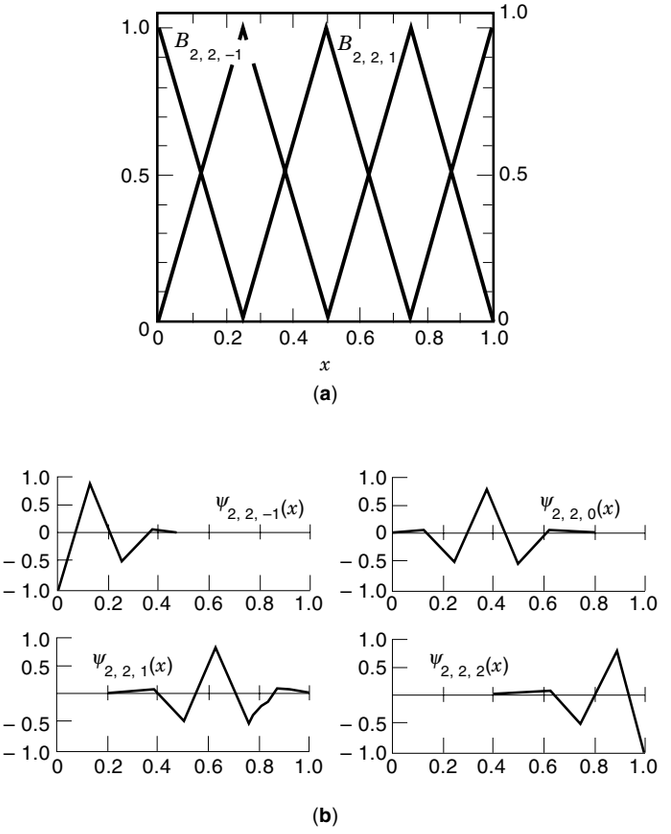


Figure 4. (a) Linear spline ($m = 2$) scaling functions on [0, 1]. (b) Linear spline wavelets on [0, 1]. The subscripts indicate the order of spline (m), scale (j), and position (k), respectively (33).

NUMERICAL RESULTS

In this section we present some numerical examples for the scattering problems described previously. Numerical results for strip and wire problems can be found in Ref. 24. Results for spectral domain applications of wavelets to transmission line discontinuity problems may be found in Ref. 16. For more applications of wavelets to electromagnetic problems, readers may refer to Ref. 32.

The matrix equation, Eq. (35), is solved for a circular cylindrical surface (33). Figure 6 shows the surface current distribution using linear splines and wavelets for different-size cylinders. The wavelet MoM results are compared with the conventional MoM results. To obtain the conventional MoM results, we have used triangular functions for both expanding the unknown current distribution and testing the resultant equation. The conventional MoM results have been verified with a series solution (40). The results of the conventional MoM and the wavelet MoM agree very well.

Next we want to show how "thresholding" affects the final solution. By "thresholding," we mean setting those elements of the matrix to zero that are smaller (in magnitude) than some positive number δ ($0 \leq \delta < 1$), called the threshold parameter, times the largest element of the matrix.

Let $z_{\max}$ and $z_{\min}$ be the largest and the smallest elements of the matrix in Eq. (35). For a fixed value of the threshold parameter δ, define % relative error (ϵ_δ) as (33)

$$\epsilon_\delta := \frac{\|f_0 - f_\delta\|_2}{\|f_0\|_2} \times 100 \tag{39}$$

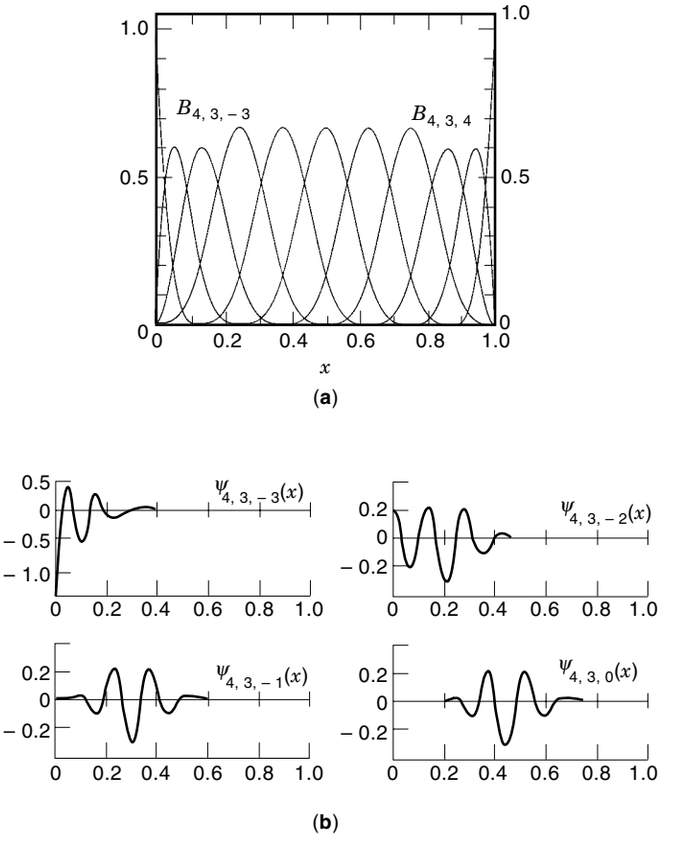


Figure 5. (a) Cubic spline ($m = 4$) scaling functions on [0, 1]. (b) Cubic spline wavelets on [0, 1]. The subscripts indicate the order of spline (m), scale (j), and position (k), respectively (33).

and % sparsity (S_δ) as

$$S_\delta := \frac{N_0 - N_\delta}{N_0} \times 100 \tag{40}$$

In the above, f_δ represents the solution obtained from Eq. (35) when the elements whose magnitudes are smaller than $\delta z_{\max}$ have been set to zero. Similarly, N_δ is the total number of elements left after thresholding. Clearly, $f_0(x) = f(x)$ and $N_0 = N^2$, where N is the number of unknowns.

Table 1 gives an idea of the relative magnitudes of the largest and the smallest elements in the matrix for conventional and wavelet MoM. As is expected, because of their higher vanishing moment property, cubic spline wavelets give the higher ratio, $z_{\max}/z_{\min}$.

Figure 7 shows a typical matrix obtained by applying the conventional MoM. A darker color on an element indicates a larger magnitude. The matrix elements with $\delta = 0.0002$ for the linear spline case are shown in Fig. 8. In Fig. 9, we present the thresholded matrix ($\delta = 0.0025$) for the cubic spline case. The $[Z_{\psi,\psi}]$ part of the matrix is almost diagonalized. Figure 10 gives an idea of the pointwise error in the solution for linear and cubic spline cases.

It is worth pointing out here that regardless of the size of the matrix, only 5×5 in the case of the linear spline and 11×11 in the case of the cubic splines (see the top-left corners of Figs. 8 and 9) will remain unaffected by thresholding;

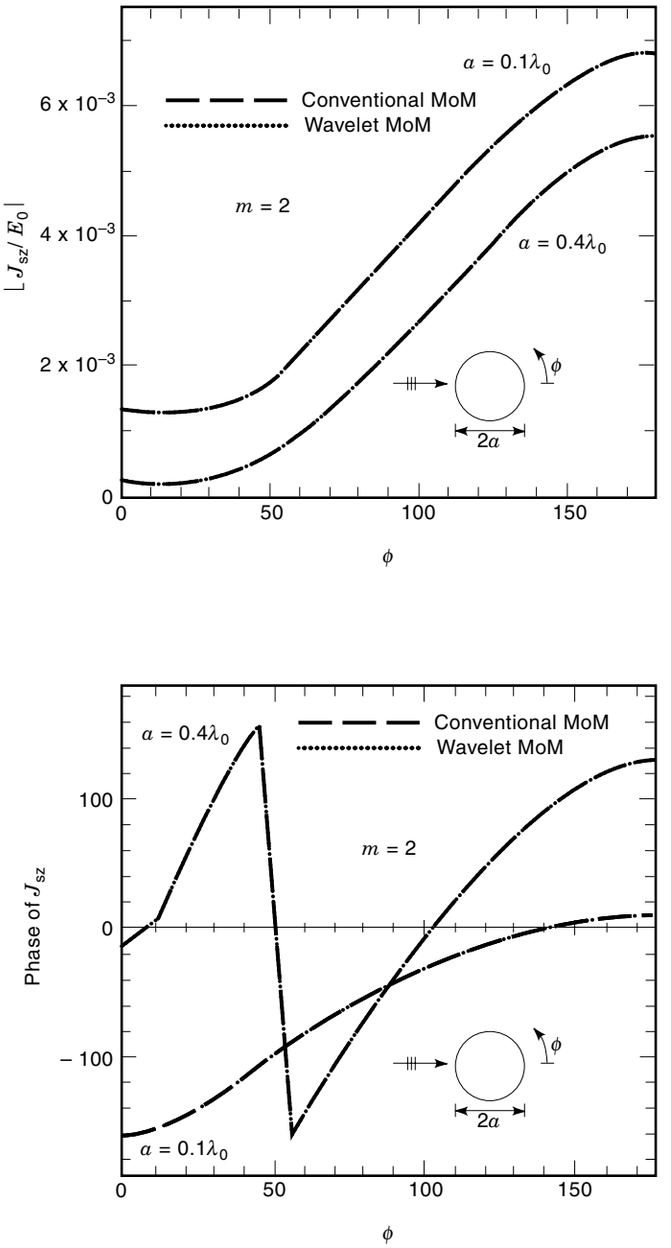


Figure 6. Magnitude and phase of the surface current distribution on a metallic cylinder using linear spline wavelet MoM and conventional MoM. Notice that the results for conventional and wavelet bases completely overlap each other (33).

Table 1. Relative Magnitudes of the Largest and the Smallest Elements of the Matrix for Conventional and Wavelet (33)

	Conventional MoM	Wavelet MoM ($m = 2$)	Wavelet MoM ($m = 4$)
$z_{\max}$	5.377	0.750	0.216
$z_{\min}$	1.682	7.684×10^{-8}	8.585×10^{-13}
Ratio	3.400	9.761×10^{6}	2.516×10^{11}

MoM. $a = 0.1\lambda_0$.

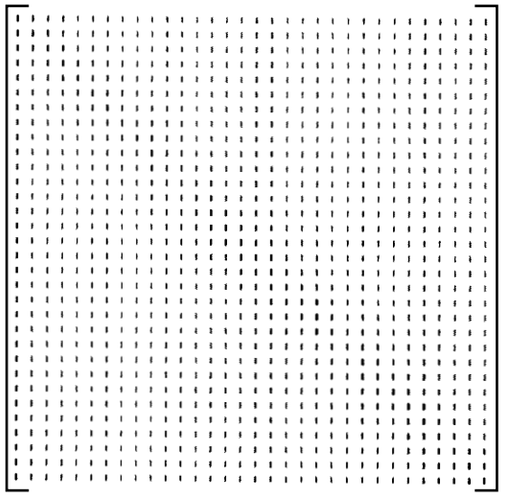


Figure 7. A typical gray-scale plot of the matrix elements obtained using conventional MoM. The darker color represents larger magnitude.

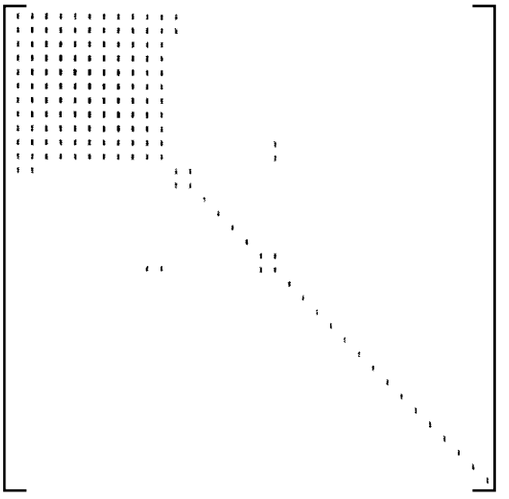


Figure 9. A typical gray-scale plot of the matrix elements obtained using cubic wavelet MoM. The darker color represents larger magnitude.

a significant number of the remaining elements can be set to zero without causing much error in the solution.

SEMIORTHOGONAL VERSUS ORTHOGONAL WAVELETS

Both semiorthogonal and orthogonal wavelets have been used for solving integral equations. A comparative study of their advantages and disadvantages has been reported in Ref. 24. The orthonormal wavelet transformation, because of its unitary similar property, preserves the condition number (κ) of the original impedance matrix Z; semiorthogonal wavelets do not. Consequently, the transformed matrix equation may require more iterations to converge to the desired solution. Some preliminary results comparing the condition number of matrices for different cases are given in Table 2.

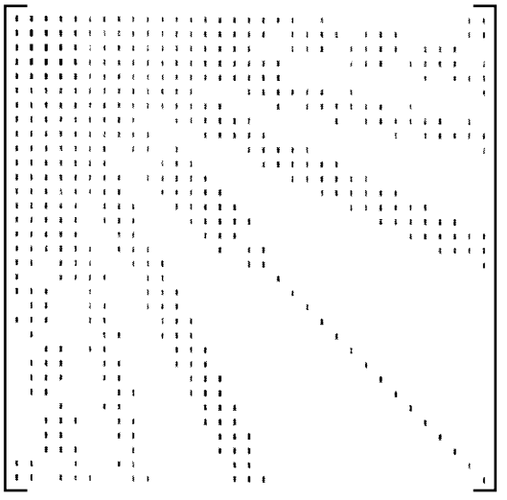


Figure 8. A typical gray-scale plot of the matrix elements obtained using linear wavelet MoM. The darker color represents larger magnitude.

In applying wavelets directly to solve integral equations, one of the most attractive features of semiorthogonal wavelets is that closed-form expressions are available for such wavelets (10,33). Most of the continuous o.n. wavelets cannot be written in closed form. One thing to be kept in mind is that, unlike signal processing applications where one usually deals with a discretized signal and decomposition and reconstruction sequences, here in the boundary value problem we often have to compute the wavelet and scaling function values at any given point. For a strip and thin wire case, a comparison of the computation time and sparsity is summarized in Tables 3 and 4 (24).

Semiorthogonal wavelets are symmetric and hence have a generalized linear phase (5, pp. 160–174), an important factor in function reconstruction. It is well known (4 Sec. 8.1) that symmetric or antisymmetric, real-valued, continuous, and compactly supported o.n. scaling functions and wavelets do not exist. Finally, in using wavelets to solve spectral domain problems, as discussed before, we need to look at the time–frequency window product of the basis. Semiorthogonal wavelets approach the optimal value of the time-frequency product, which is 0.5, very fast. For instance, this value for the cubic spline wavelet is 0.505. It has been shown (41) that this product approaches to ∞ with the increase in smoothness of o.n. wavelets.

DIFFERENTIAL EQUATIONS

An ordinary differential equation (ODE) can be represented as

$$Lf(x) = g(x); x \in [0, 1] \tag{41}$$

with

$$L = \sum_{j=0}^{m} a_j(x) \frac{d^j}{dx^j} \tag{42}$$

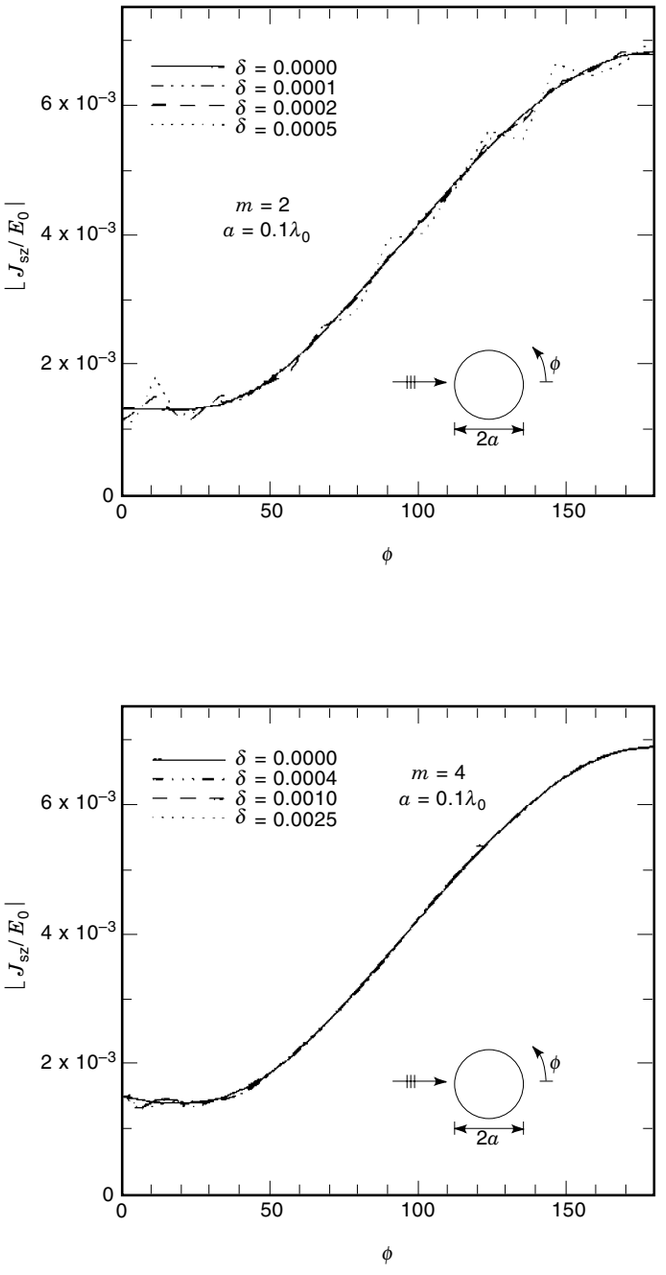


Figure 10. The magnitude of the surface current distribution computed using linear ($m = 2$) and cubic ($m = 4$) spline wavelet MoM for different values of the threshold parameter δ (33).

Table 3. Comparison of CPU Time per Matrix Element for Spline, Semiorthogonal, and Orthonormal Basis Function (24)

	Wire	Plate
Spline	0.12 s	0.25×10^{-3} s
s.o. Wavelet	0.49 s	0.19 s
o.n. Wavelet	4.79 s	4.19 s

and some appropriate boundary conditions. If the coefficients $\{a_j\}$ are independent of x, then the solution can be obtained via a Fourier method. However, in the ODE case, with nonconstant coefficients, and in PDEs, we generally use finite-element- or finite-difference-type methods.

In the traditional finite-element method (FEM), local bases are used to represent the unknown function and the solution is obtained by Galerkin's method, similar to the approach described in previous Sections. For the differential operator, we get sparse and banded stiffness matrices that are generally solved using iterative techniques, the Jacobi method for instance.

One of the disadvantages of conventional FEM is that the condition number (κ) of the stiffness matrix grows as $O(h^{-2})$, where h is the discretization step. As a result, the convergence of the iterative technique becomes slow and the solution becomes sensitive to small perturbations in the matrix elements. If we study how the error decreases with iteration in iterative techniques, such as the Jacobi method, we find that the error decreases rapidly for the first few iterations. After that, the rate at which the error decreases slows down (42, pp. 18–21). Such methods are also called "high-frequency methods" since these iterative procedures have a "smoothing" effect on the high-frequency portion of the error. Once the high-frequency portion of the error is eliminated, convergence becomes quite slow. After the first few iterations, if we could re-discretize the domain with coarser grids and thereby go to lower frequency, the convergence rate would be accelerated. This leads us to a multigrid-type method.

Multigrid or hierarchical methods have been proposed to overcome the difficulties associated with the conventional method (42–58). In this technique, one performs a few iterations of the smoothing method (Jacobi-type), and then the intermediate solution and the operator are projected to a coarse grid. The problem is then solved at the coarse grid, and by interpolation one goes back to the finer grids. By going back and forth between finer and coarse grids, the convergence can be accelerated. It has been shown for elliptic PDEs that for wavelet-based multilevel methods, the condition number is

Table 2. Effect of Wavelet Transform Using Semiorthogonal and Orthonormal Wavelets on the Condition Number of the Impedance Matrix[a]

Basis and Transform	Number of Unknowns	Octave Level	δ	S_δ	ϵ_δ	Condition Number κ Before Threshold	Condition Number κ After Threshold
Pulse and none	64	NA	NA	0.0	2.6×10^{-5}	14.7	—
Pulse and s.o.	64	1	7.2×10^{-2}	46.8	0.70	16.7	16.4
Pulse and o.n.	64	1	7.5×10^{-3}	59.7	0.87	14.7	14.5

[a] Original impedance matrix is generated using pulse basis functions.

Table 4. Comparison of Percentage Sparsity (S_δ) and Percentage Relative Error (ϵ_δ) for Semiorthogonal and Orthonormal Wavelet Impedance Matrices as a Function of Threshold Parameter (δ) (24)

	Number of Unknowns		Threshold	Sparsity S_δ		Relative Error ϵ_δ	
Scatterer/Octave Levels	s.o.	o.n.	δ	s.o.	o.n.	s.o.	o.n.
Wire/j = 4	29	33	1×10^{-6}	34.5	24.4	3.4×10^{-3}	4.3×10^{-3}
			5×10^{-6}	48.1	34.3	3.9	1.3×10^{-3}
			1×10^{-5}	51.1	36.5	16.5	5.5×10^{-2}
Plate/j = 2, 3, 4	33	33	1×10^{-4}	51.6	28.1	1×10^{-4}	0.7
			5×10^{-4}	69.7	45.9	4.7	5.2
			1×10^{-3}	82.4	50.9	5.8	10.0

independent of the discretization step, that is, $\kappa = O(1)$ (53). The multigrid method is too involved to be discussed in this article. Readers are encouraged to look at the references provided at the end of this article.

Multiresolution aspects of wavelets have also been applied in evolution equations (57,58). In evolution problems, the space and time discretization are interrelated to gain a stable numerical scheme. The time-step must be determined from the smallest space discretization. This makes the computation quite complex. A space-time adaptive method has been introduced in Ref. 58, where wavelets have been used to adjust the space-time discretization steps locally.

BIBLIOGRAPHY

1. G. B. Arfken and H. J. Weber, *Mathematical Methods for Physicists,* San Diego, CA: Academic Press, 1995.
2. Y. Meyer, *Wavelets: Algorithms and Applications,* Philadelphia, PA: SIAM, 1993.
3. S. Mallat, A theory of multiresolution signal decomposition: The wavelet representation, *IEEE Trans. Pattern Anal. Mach. Intell.,* **11**: 674–693, 1989.
4. I. Daubechies, *Ten Lectures on Wavelets,* CBMS-NSF Ser. Appl. Math., No. 61, Philadelphia, PA: SIAM, 1992.
5. C. K. Chui, *An Introduction to Wavelets,* Boston: Academic Press, 1992.
6. C. K. Chui, *Wavelets: A Mathematical Tool for Signal Analysis,* Philadelphia, PA: SIAM, 1997.
7. G. Strang and T. Nguyen, *Wavelets and Filter Banks,* Wellesley, UK: Wellesley-Cambridge Press, 1996.
8. M. Vetterli and J. Kovačević, *Wavelets and Subband Coding,* Upper Saddle River, NJ: Prentice-Hall, 1995.
9. A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition,* San Diego, CA: Academic Press, 1992.
10. J. C. Goswami and A. K. Chan, *Fundamentals of Wavelets: Theory, Algorithms, and Applications,* New York: Wiley, 1999.
11. A. Cohen, I. Daubechies, and J. C. Feauveau, Biorthonormal bases of compactly supported wavelets, *Commun. Pure Appl. Math.,* **45**: 485–500, 1992.
12. G. M. Wing, *A Primer on Integral Equations of the First Kind,* Philadelphia, PA: SIAM, 1991.
13. J. H. Richmond, Digital solutions of the rigorous equations for scattering problems, *Proc. IEEE,* **53**: 796–804, 1965.
14. B. K. Alpert et al., Wavelet-like bases for the fast solution of second-kind integral equations, *SIAM J. Sci. Comput.,* **14**: 159–184, 1993.
15. J. Mandel, On multi-level iterative methods for integral equations of the second kind and related problems, *Numer. Math.,* **46**: 147–157, 1985.
16. J. C. Goswami, An application of wavelet bases in the spectral domain analysis of transmission line discontinuities, *Int. J. Numer. Model.,* **11**: 41–54, 1998.
17. R. F. Harrington, *Field Computation by Moment Methods,* New York: IEEE Press, 1992.
18. G. Beylkin, R. Coifman, and V. Rokhlin, Fast wavelet transform and numerical algorithms I, *Commun. Pure Appl. Math.,* **44**: 141–183, 1991.
19. R. L. Wagner, P. Otto, and W. C. Chew, Fast waveguide mode computation using wavelet-like basis functions, *IEEE Microw. Guided Wave Lett.,* **3**: 208–210, 1993.
20. H. Kim and H. Ling, On the application of fast wavelet transform to the integral-equation of electromagnetic scattering problems, *Microw. Opt. Technol. Lett.,* **6**: 168–173, 1993.
21. B. Z. Steinberg and Y. Leviatan, On the use of wavelet expansions in method of moments, *IEEE Trans. Antennas Propag.,* **41**: 610–619, 1993.
22. K. Sabetfakhri and L. P. B. Katehi, Analysis of integrated millimeter-wave and submillimeter-wave waveguides using orthonormal wavelet expansions, *IEEE Trans. Microw. Theory Tech.,* **42**: 2412–2422, 1994.
23. B. Z. Steinberg, A multiresolution theory of scattering and diffraction, *Wave Motion,* **19** (3): 213–232, 1994.
24. R. D. Nevels, J. C. Goswami, and H. Tehrani, Semi-orthogonal versus orthogonal wavelet basis sets for solving integral equations, *IEEE Trans. Antennas Propag.,* **45**: 1332–1339, 1997.
25. Z. Xiang and Y. Lu, An effective wavelet matrix transform approach for efficient solutions of electromagnetic integral equations, *IEEE Trans. Antennas Propag.,* **45**: 1205–1213, 1997.
26. Z. Baharav and Y. Leviatan, Impedance matrix compression (IMC) using iteratively selected wavelet basis, *IEEE Trans. Antennas Propag.,* **46**: 226–233, 1997.
27. R. D. Nevels and R. E. Miller, A comparison of moment impedance matrices obtained by direct and transform matrix methods using wavelet basis functions, *IEEE Trans. Antennas Propag. Soc. Int. Symp.,* 1997.
28. B. Z. Steinberg and Y. Leviatan, Periodic wavelet expansions for analysis of scattering from metallic cylinders, *IEEE Antennas Propag. Soc. Symp.,* 1994, pp. 20–23.
29. G. W. Pan and X. Zhu, The application of fast adaptive wavelet expansion method in the computation of parameter matrices of multiple lossy transmission lines, *IEEE Antennas Propag. Soc. Symp.,* 1994, pp. 29–32.
30. G. Wang, G. W. Pan, and B. K. Gilbert, A hybrid wavelet expansion and boundary element analysis for multiconductor transmis-

sion lines in multilayered dielectric media, *IEEE Trans. Microw. Theory Tech.,* **43**: 664–675, 1995.

31. W. L. Golik, Wavelet packets for fast solution of electromagnetic integral equations, *IEEE Trans. Antennas Propag.,* **46**: 618–624, 1998.
32. *Int. J. Numerical Modeling: Electron. Netw., Devices Fields,* Special Issue on Wavelets in Electromagnetics, **11**: 1998.
33. J. C. Goswami, A. K. Chan, and C. K. Chui, On solving first-kind integral equations using wavelets on a bounded interval, *IEEE Trans. Antennas Propag.,* **43**: 614–622, 1995.
34. C. K. Chui, J. C. Goswami, and A. K. Chan, Fast integral wavelet transform on a dense set of time-scale domain, *Numer. Math.,* **70**: 283–302, 1995.
35. J. C. Goswami, A. K. Chan, and C. K. Chui, On a spline-based fast integral wavelet transform algorithm, in H. L. Bertoni et al. (eds.), *Ultra-Wideband Short Pulse Electromagnetics 2,* New York: Plenum, 1995, pp. 455–463.
36. J. C. Goswami, A. K. Chan, and C. K. Chui, An application of fast integral wavelet transform to waveguide mode identification, *IEEE Trans. Microw. Theory Tech.,* **43**: 655–663, 1995.
37. E. Quak and N. Weyrich, Decomposition and reconstruction algorithms for spline wavelets on a bounded interval, *Appl. Comput. Harmonic Anal.,* **1**: 217–231, 1994.
38. A. Cohen, I. Daubechies, and P. Vial, Wavelets on the interval and fast wavelet transform, *Appl. Comput. Harmonic Anal.,* **1**: 54–81, 1993.
39. P. Auscher, Wavelets with boundary conditions on the interval, in C. K. Chui (ed.), *Wavelets: A Tutorial in Theory and Applications,* Boston: Academic Press, 1992, pp. 217–236.
40. R. F. Harrington, *Time-Harmonic Electromagnetic Fields,* New York: McGraw-Hill, 1961.
41. C. K. Chui and J. Z. Wang, High-order orthonormal scaling functions and wavelets give poor time-frequency localization, *Fourier Anal. Appl.,* **2**: 415–426, 1996.
42. W. Hackbusch, *Multigrid Methods and Applications,* New York: Springer-Verlag, 1985.
43. A. Brandt, Multi-level adaptive solutions to boundary value problems, *Math. Comput.,* **31**: 330–390, 1977.
44. W. L. Briggs, *A Multigrid Tutorial,* Philadelphia, PA: SIAM, 1987.
45. S. Dahlke and I. Weinreich, Wavelet–Galerkin methods: An adapted biorthogonal wavelet basis, *Construct. Approx.,* **9**: 237–262, 1993.
46. W. Dahmen, A. J. Kurdila, and P. Oswald (eds.), *Multiscale Wavelet Methods for Partial Differential Equations,* San Diego, CA: Academic Press, 1997.
47. H. Yserentant, On the multi-level splitting of finite element spaces, *Numer. Math.,* **49**: 379–412, 1986.
48. J. Liandrat and P. Tchamitchian, Resolution of the 1-D regularized Burgers equation using a spatial wavelet approximation, *NASA Rep. ICASE,* **90-83**: 1990.
49. R. Glowinski et al., Wavelet solution of linear and nonlinear elliptic, parabolic, and hyperbolic problems in one space dimension, in R. Glowinski and A. Lichnewsky (eds.), *Computing Methods in Applied Sciences and Engineering,* Philadelphia. PA: SIAM, 1990, pp. 55–120.
50. P. Oswald, On a hierarchical basis multilevel method with nonconforming P1 elements, *Numer. Math.,* **62**: 189–212, 1992.
51. W. Dahmen and A. Kunoth, Multilevel preconditioning, *Numer. Math.,* **63**: 315–344, 1992.
52. P. W. Hemker and H. Schippers, Multiple grid methods for the solution of Fredholm integral equations of the second kind, *Math. Comput.,* **36** (153): 215–232, 1981.
53. S. Jaffard, Wavelet methods for fast resolution of elliptic problems, *SIAM J. Numer. Anal.,* **29**: 965–986, 1992.
54. S. Jaffard and P. Laurençot, Orthonomal wavelets, analysis of operators, and applications to numerical analysis, in C. K. Chui (ed.), *Wavelets: A Tutorial in Theory and Applications,* Boston: Academic Press, 1992, pp. 543–601.
55. J. Xu and W. Shann, Galerkin–wavelet methods for two-point boundary value problems, *Numer. Math.,* **63**: 123–144, 1992.
56. C. Guerrini and M. Piraccini, Parallel Wavelet–Galerkin methods using adapted wavelet packet bases, in C. K. Chui and L. L. Schumaker (eds.), *Wavelets and Multilevel Approximation,* River Edge, NJ: World Scientific, 1995, pp. 133–142.
57. M. Krumpholz and L. P. B. Katehi, MRTD: New time-domain schemes based on multiresolution analysis, *IEEE Trans. Microw. Theory Tech.,* **44**: 555–571, 1996.
58. E. Bacry, S. Mallat, and G. Papanicolaou, A wavelet based space-time adaptive numerical method for partial differential equations, *Math. Model. Numer. Anal.,* **26**: 793–834, 1992.

JAIDEVA C. GOSWAMI
Sugar Land Product Center
RICHARD E. MILLER
ROBERT D. NEVELS
Texas A&M University

/ /

Wiley Encyclopedia of Electrical and Electronics Engineering

Wavelet Transforms
Standard Article

Stanley R. Deans[1], John J. Heine[1], Deepak Gangadharan[1], Wei Qian[1], Maria Kallergi[1], Laurence P. Clarke[1]
[1]University of South Florida, Tampa, FL

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2466
Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (379K)

Abstract

The sections in this article are

Basic Concepts

Classification of Signals

Time–Frequency Resolution

Some History

The Continuous Wavelet Transform

Discrete Wavelet Transform

Mechanics of Doing the Transform

Octave Band Tree Structure

Some Interesting Applications

Wavelets on the Internet

| | |

Copyright © 1999-2008 All Rights Reserved.

WAVELET TRANSFORMS

Functions that oscillate over time are often called waves. If the function is such that it oscillates only in a localized region and goes to zero outside the region it may be called a wavelet. Thus we say wavelets are localized waves. This is analogous to many processes in nature. Consider a sound wave that starts out at zero, builds to some maximum, and then dies out to zero. If the duration of the sound is a few seconds we say the scale for the process is on the order of seconds. Whenever examining some physical object scale plays an important role. For example, when looking at another human at a scale of about a meter you see the whole individual, but if you examine the same individual at a scale of about a centimeter you can see details such as whorled ridges that form the fingerprint.

The fundamental role of the wavelet transform is to facilitate the analysis of signals or images according to scale. Wavelets are functions with some very special mathematical properties that serve as a tool for efficiently dividing data into a sequence of frequency components without losing all information about position. This can be thought of in terms of viewing an object through different size windows. If a large window is used we see gross features, and if a small window is used we only see small detail features. There are many similarities between wavelet analysis and classical windowed Fourier analysis. The goal in the latter is to determine the local frequency content of a signal by using sine and cosine functions multiplied by a sliding window. The wavelet analysis makes use of translations and dilations of an oscillating wavelet, called the *mother wavelet,* to characterize both spatial and frequency contents of a signal. The properties of this analyzing wavelet are very different from those of sines and cosines. These differences make it possible to approximate a signal contained in a finite region or a signal with sharp changes with a few coefficients, something not possible with classical Fourier methods.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering.* Copyright © 1999 John Wiley & Sons, Inc.

Many of the principles that are the foundation for wavelet analysis emerged independently in mathematics, physics, geophysics, and engineering. In most cases the concepts came from the motivation to solve some problem that related to resolution or scale. During the last decade wavelets have been used with great success in a very wide variety of areas, including image compression, coding, signal processing, numerical analysis, turbulence, acoustics, seismology, and medical imaging.

There are some basic mathematical concepts that must be understood prior to a full explanation of the two types of wavelet transforms, the continuous transform and the discrete transform. The next section on basic concepts from linear algebra and Fourier analysis can be skipped by those who have already reached that level of mathematical sophistication.

BASIC CONCEPTS

Basis

One of the most fundamental ideas associated with many areas of mathematics is the concept of a *basis.* A simple illustration serves to get the idea across. Suppose we have a curve (waveform or signal) that looks somewhat complicated, as in Fig. 1. (In practice this could be a voltage that varies in time.) How could you explain to someone who could not see the curve just what it looks like? One possible way is to think of the complicated curve as being made up of the sum of several simple curves. The complicated curve is selected so that it is exactly the sum of two simple curves shown in Fig. 2. The simple curves are known as sine curves. These fundamental curves can be described in terms of how many times they go through a complete cycle. Note that the low-frequency curve goes through one cycle and the higher frequency curve goes through three cycles. Also note that the low-frequency curve has two times the amplitude of the high frequency curve. You could tell someone exactly how to reproduce the more complicated curve by giving the information about frequency and amplitude for the two basic curves. For those familiar with formulas for sine curves, the complicated curve is given by $y = 2\sin(2\pi x) + \sin(6\pi x)$.

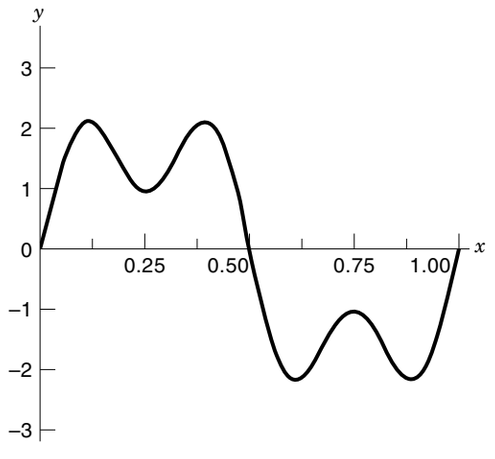


Figure 1. A complicated curve.

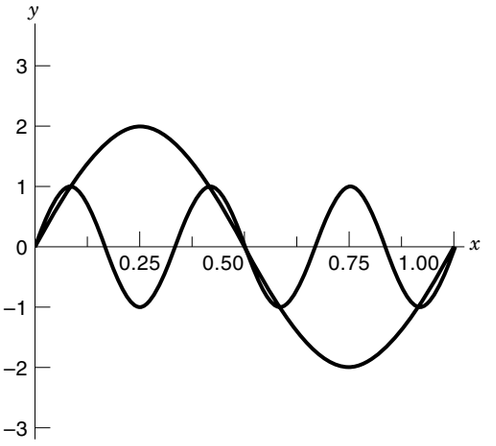


Figure 2. Two simple curves. The curve in Fig. 1 is the sum of these two curves.

Technical Definitions: Motion that repeats in equal intervals of time is called periodic. The *period* is the time required for one complete cycle or oscillation. The *frequency* is the repetition rate of a periodic process. This is the number of cycles that occur over a given interval of time. If the period is given in seconds, the frequency is in hertz, abbreviated Hz. In Fig. 2, if the x axis represents time (in seconds) the frequencies are 1 Hz and 3 Hz for the two curves.

The concept of a basis comes from an extension of the approach used to produce the curve in Fig. 1 from the sum of the curves in Fig. 2. If the basis is selected so that it is complete, an arbitrary curve can be replaced by a sum of basic curves. When this is done for periodic functions using sine or cosine curves with different frequencies and amplitudes it is called a Fourier series decomposition. If the original function is not periodic and can be defined over the entire x axis such that its area is finite, a Fourier transform is used.

The important concept here is that there is a formal way to represent a function or waveform as a sum of basic parts. Fourier analysis corresponds to the language used, and there is a prescription for calculating the coefficients in the sum. This corresponds to finding the amplitudes in Fig. 2. You might think of this as a sort of mathematical prism. The prism breaks light into various colors in much the same way the Fourier analysis breaks the complicated waveform into component parts.

When considering all sorts of waveforms an obvious question emerges. Under just what conditions is Fourier analysis the appropriate mathematical language to use to decompose the waveform? The complete answer to this question is the subject of the enormous literature on Fourier series and Fourier transforms. There are some simple answers that will suffice for our purposes. The sum in a typical Fourier series problem is an infinite sum. This means an infinite number of coefficients must be computed to represent the function. It seems we have made the problem more complicated! It turns out that in many physical situations only a few coefficients are needed to give an adequate description of the waveform. The coefficients associated with the high-frequency sines and cosines approach zero as the frequency increases. You can think about it this way: The large coefficients correspond to

the case where there is a fair match between the original function and the basic sine or cosine. If the original waveform changes slowly relative to the high-frequency oscillations there is a poor match, and consequently the coefficients are very small. More will be said about this in the sections that follow.

Orthogonality

Another concept that is essential is that of *orthogonality*. Recall from elementary geometry, if two line segments are perpendicular we say they are orthogonal. If we make vectors out of line segments by giving them properties of magnitude and direction we can determine whether they are orthogonal or not by computing their *scalar* product. This is sometimes called the *dot* or *inner* product. If the scalar product is zero they are orthogonal. Another way to think about this is that orthogonal vectors do not have any components in common, or they contain completely independent information. The same type of thing can be defined for functions; however, the rule for doing the scalar product is different. It involves doing an integral of the product of two functions. The coefficients in a Fourier series expansion can be found by computing scalar products of the original waveform multiplied by sine and cosine functions with different frequencies. The important point is that the building blocks, the sines and cosines of different frequencies, are orthogonal and complete. An important consequence is that the frequency content of the waveform can be determined in an unambiguous way. Also, an orthogonal transformation allows perfect reconstruction of the original waveform and eliminates redundancy. Generally, orthogonal transformations are more efficient and easier to use.

Sampling and the Fast Fourier Transform

In nature many waveforms are continuous functions of time. If we want to work with these signals using digital computers it is necessary to find a discrete representation. This means we have to *sample* the continuous function. There is an extremely important theorem known as the *Shannon sampling theorem* that is invoked in these situations. Proofs are given in most standard texts on Fourier analysis, for example, Bracewell (1) and Brigham (2). The theorem states that a continuous signal can be represented completely by and reconstructed perfectly from a set of measurements (samples) of its amplitude made at equally spaced times. The time interval between samples must be equal to or less than one-half the period of the highest frequency present in the signal. For example, for a typical voice signal the frequency range is from 0 Hz to 4,000 Hz. This signal must be sampled 8,000 times per second in order to describe it perfectly. In practice the idea of perfect reconstruction must be compromised. When the amplitude is sampled with real physical apparatus there must be some sort of round off. In speech transmission an error of 1% is often sufficient for practical purposes.

Another development that helped usher in the digital communication revolution is the *Fast Fourier Transform* (FFT). For n sampled points this reduces the number of computations from n^2 to $n \log n$. This is especially important for large values of n. A very interesting discussion of the FFT is given by Heideman, Johnson, and Burrus (3).

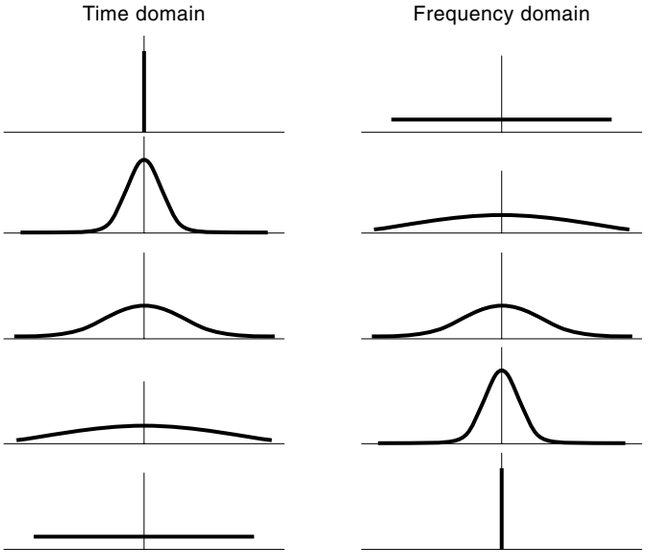


Figure 3. Signals concentrated in one domain are spread in the other domain.

Time and Frequency Domains

When we look at the signal in the time domain we have full information about the amplitude of the signal at any time. When we do the Fourier decomposition we have full information about the frequency content of the signal, but the time information is not apparent. The inverse transform yields the time information, but then the frequency spectrum is not apparent. Another way to think about this is to observe that a very sharp signal in the time domain is flat in the frequency domain. Inspection of the frequency spectrum does not tell when the sharp signal occurred in time. Some time domain and frequency domain transform pairs are shown in Fig. 3. The important point is that signals localized in time are spread in frequency and those spread in time are localized in frequency.

CLASSIFICATION OF SIGNALS

It is useful to give a broad classification of signals as *stationary, quasi-stationary,* and *nonstationary*. A signal is *stationary* if its statistical properties are invariant over the time duration of the signal. For these signals the probability of unexpected events is known in advance. If there are transient events (such as blips or discontinuities) in the signal that cannot be predicted, even with knowledge of the past, the signal is *nonstationary*. Consider viewing the signal through a window of some width; that is, look at a section of the signal. A signal is called *quasi-stationary* if the signal is stationary at the scale of the window.

The ideal tool for studying stationary signals is Fourier analysis. The study of nonstationary signals requires other techniques. One of these is the use of wavelets. An important technique for the study of quasi-stationary signals came before wavelets and will be discussed first.

The desire to maintain information about time when doing Fourier decompositions leads to the short-time Fourier transform (STFT), sometimes called the windowed Fourier trans-

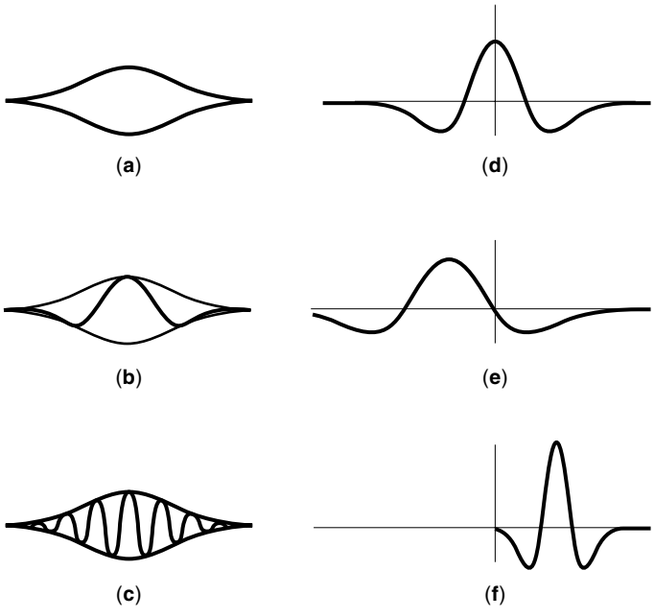


Figure 4. Comparison of STFT and wavelets. On the left, select the window and allow different frequency sinusoids to fill the window. On the right, select the mother wavelet, then translate and dilate the wavelet. The second wavelet is expanded and shifted to the left. The third wavelet is compressed and shifted to the right.

form. The idea is to select a window with fixed width and slide it along the signal. The Fourier decomposition is done for several short times of the signal rather than for the entire signal all at once. By a proper choice of the window it is possible to maintain both time and frequency information; thus, this transformation is known as a time–frequency decomposition. When the window is a Gaussian the transform is known as a Gabor transform in honor of early work done by Dennis Gabor (4). Difficulties in connection with this approach involve both orthogonality and invertibility. An introduction to the STFT and the wavelet transform is given by Rioul and Vetterli (5). The close connection with wavelets is illustrated in Fig. 4. Note that the main difference is that the functional form of the wavelet does not change, hence the name *mother wavelet.* The choices for the mother wavelet are virtually unlimited. This is in sharp contrast to Fourier analysis where the basis functions are sines and cosines.

The mother wavelet is allowed to undergo translations and dilations. It is the various translations and dilations of the mother wavelet that form the basis functions for the wavelet transform. This stretching or compressing of the wavelet changes the size of the window and allows the analysis of signals at different scales. This is in some sense like a microscope; the wide stretched out wavelets are used to give a broad approximate image of the signal while the smaller and smaller compressed wavelets can zoom in on finer and finer details.

TIME–FREQUENCY RESOLUTION

We have already seen that sharp signals in the time domain correspond to flatness in the frequency domain. If the window for the STFT is selected as in the third row of Fig. 3 then the tiles that represent the essential concentration in the time–frequency plane are squares as indicated on the left in Fig. 5. For a window fixed at one position along the time axis, going up vertically corresponds to higher and higher frequency sinusoidal curves contained within the window. The corresponding tiles for the wavelet transform are shown on the right in Fig. 5. Here the wavelet that is stretched out over the time axis (low frequency) has a narrow concentration in the frequency domain. As the wavelet is compressed (higher fre-

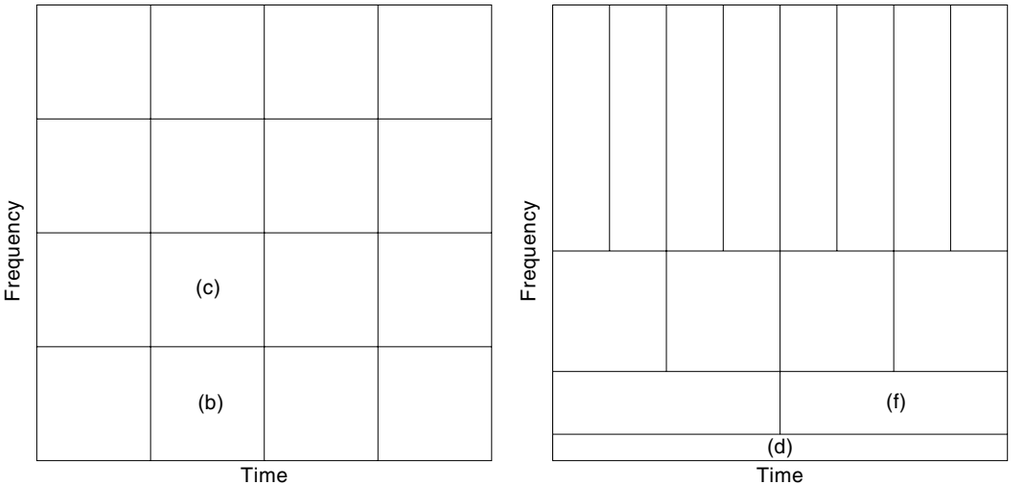


Figure 5. Left: Time and frequency resolution for STFT. Right: Time and frequency resolution for WT. The tiles indicate the region of concentration in the time–frequency plane for a basis function. As an illustration, if the tile labeled (b) corresponds to (b) in Fig. 4, then tile (c) could correspond to (c) in Fig. 4. The corresponding comparison for wavelets is for tiles labeled (d) and (f) with (d) and (f) in Fig. 4.

quencies, smaller time window) the concentration in the frequency domain is less and less concentrated. Think about this in terms of the curves in Fig. 3 where small time windows correspond to broad frequency windows. This helps in understanding how scale plays such an important role in the wavelet transform.

SOME HISTORY

There are many avenues that can be followed in trying to trace the history of wavelets. Barbara Burke Hubbard (6) has a quote in her beautiful discussion of wavelets by Yves Meyer. He says: "I have found at least 15 distinct roots of the theory, some going back to the 1930s." Seven of these sources in pure mathematics are discussed in some detail in the translation of some Meyer's (7) lecture notes. The reader with some background in harmonic analysis will find this discussion covering 70 years of mathematics fascinating: the Haar basis (1909), the Franklin orthonormal system (1927), Littlewood–Paley theory (1930), Calderón–Zygmund theory (1960–1978), and the work of Strömberg (1980). In addition to the lecture notes by Meyer, references for this background include Haar (8), Franklin (9), Hernández and Weiss (10), Edwards and Gaudry (11) for Littlewood–Paley theory, Stein (12) for Calderón–Zygmund theory, and Strömberg (13).

This work done by mathematicians is now understood as part of the history of wavelets. The term "atomic decompositions" was used in place of the term wavelets. During this period from 1910 to 1980, mathematicians from the University of Chicago (location of Zygmund and Calderón) were leaders in harmonic analysis, but apparently they did not interact very much with the experts in physics and signal processing.

In physics, the ideas underlying wavelets are present in Nobel laureate Kenneth Wilson's (14) work on the renormalization group. A review of some of Wilson's work and other uses of wavelets in physics is given by Guy Battle (15,16). Wavelet concepts also appear in the study of coherent states in quantum mechanics. This work dates from the early 1960s by Glauber (17) and Aslaksen and Klauder (18,19).

In parallel with advances in mathematics and physics there were important ideas fundamental to wavelets being developed in signal and image processing. This work was mostly in the context of discrete-time signals. As is often the case in applied science much of this work was driven by the need to solve a problem. We have already mentioned the work by Gabor who introduced concepts very close to wavelets in speech and signal processing. A technique called subband coding was proposed by Croisier, Esteban, and Galand (20) for speech and image compression. This work and related work by Esteban and Galand (21) and Crochiere, Webber, and Flanagan (22) made use of special filters known as quadrature mirror filters (QMF). This led to important work in perfect reconstruction filter banks discussed in detail by Vetterli and Kovačević (23). Other important relevant work was the development of pyramidal algorithms in image processing by Burt and Adelson (24), where images are approximated proceeding from a coarse to fine resolution. This idea is similar to the multiresolution framework currently used in connection with the discrete wavelet transform.

The important point of all of this is that the foundations of wavelet transforms were implicit in several areas of science, but those working in the various areas were not communicating outside their own field. The grand unification came as a surprise to many and is certainly one reason why this subject has become so popular. Several people made important contributions to this unification. Yves Meyer, in the foreword to the book by Hernández and Weiss (10), gives special tribute to Alex Grossmann and Stéphane Mallat.

In the early 1980s Jean Morlet, a geophysicist with the French oil company Elf-Aquitaine, coined the name wavelet in connection with analysis of data in oil prospecting [see Morlet et al. (25)]. Morlet's early work was based on extensions of the Gabor transform coupled with the fundamental idea of holding the number of oscillations in the window constant while varying the width of the window. Morlet developed empirical methods for decomposing a signal into wavelets and then reconstructing the original signal, but it was not clear how general the numerical techniques were. Morlet was referred to Alex Grossmann who had extensive experience in Fourier analysis as utilized in quantum mechanics. it took them about two years to determine that the inversion was exact, and not an approximation [Grossmann and Morlet (26)].

During 1985–1986 Stéphane Mallat (27,28), an expert in computer vision, signal processing, and applied mathematics, discovered some important connections among: (1) the quadrature mirror filters, (2) the pyramid algorithms, and (3) the orthonormal wavelet bases of Strömberg Meyer, building on the work by Mallat, constructed wavelets that are continuously differentiable but they do not have compact support. (A function with compact support vanishes outside a finite interval.) A full discussion of these Meyer wavelets is given by Ingrid Daubechies (29), where she points out that Meyer actually found this basis while trying to prove the nonexistence of such nice wavelet bases. It requires a considerable amount of work to calculate the wavelet coefficients for the Meyer wavelets, and Daubechies wanted to construct wavelets that would be easier to use. She had worked with Grossmann in France on her Ph.D. research in physics and she knew about Mallat and Meyer's work before it was published. She demanded orthogonality, compact support, and some degree of smoothness (wavelets with vanishing moments). These constraints are so much in conflict that most people doubted such a task could be accomplished. After some very intense work she had the construction by the end of March 1987. See the revealing quote on page 47 of Hubbard (6). This work is elegant and the Daubechies wavelets have become the cornerstone of wavelet applications throughout the world. The first publication on her construction is in Ref. 30. Other relevant descriptions are in Refs. 29 and 31.

This concludes an all too brief history of a topic that has roots reaching into the core of pure and applied mathematics, physics, geophysics, computer science, and engineering. The reader with interest in these matters will find the informal discussion by Ingrid Daubechies (32) both enjoyable and enlightening. In that discussion she does not cite specific references, but all of the characters in the story are identified in the bibliography or reading list for this article or in the books by Vetterli and Kovačević (23) and Daubechies (29).

THE CONTINUOUS WAVELET TRANSFORM

Families of continuous wavelets are found by shifting and scaling a "mother" wavelet $\psi(x)$

$$\psi_{a,b} = \frac{1}{\sqrt{a}}\,\psi\left(\frac{x-b}{a}\right), \qquad a, b \in \mathbb{R}, \quad a \neq 0 \tag{1}$$

The parameter a is the scale parameter, b is the shift parameter, and $\mathbb{R}$ is the set of real numbers. One possible identification for ψ is the Mexican hat function,

$$\psi(x) = \frac{2}{\sqrt{3}}\,\pi^{-1/4}(1-x^2)e^{-1/2x^2}$$

This function is the second derivative of a Gaussian $e^{-1/2x^2}$. The normalization is such that its square integrated over the real line is unity, $L^2(\mathbb{R})$ norm equal to 1. This is the function used for illustration on the right side of Fig. 4. The reason for the name comes from the image generated by a rotation around its axis of symmetry (29). Observe that for large a the basis function $\psi_{a,b}$ is a stretched out version of ψ and small a gives a contracted version.

If the basis functions are required to satisfy a completeness condition, then it is necessary for the wavelet to satisfy an "admissibility" condition (23)

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\xi)|^2}{\xi}\,d\xi < \infty$$

where $\hat{\psi}$ is the Fourier transform of ψ,

$$\hat{\psi}(\xi) = \int_{-\infty}^{\infty} \psi(x)e^{-ix\xi}\,dx$$

This means that for practical cases we must require (Let $\xi \to 0$ in the formula for the Fourier transform):

$$\int_{-\infty}^{\infty} \psi(x)\,dx = \hat{\psi}(0) = 0$$

Thus, the wavelet function cannot be a symmetric positive "bump" function like a Gaussian, but must wiggle around the x axis like a wave. The zero of the Fourier transform at the origin and the decay of the spectrum $\hat{\psi}$ at high frequencies implies that the wavelet has a bandpass behavior.

The continuous wavelet transform of a function $f(x)$ is defined by

$$\tilde{f}(a,b) = \langle \psi_{a,b}, f\rangle = \int_{-\infty}^{\infty} \psi_{a,b}(x) f(x)\,dx \tag{2}$$

for a *real* set of basis functions. The function f is recovered from the transformed function $\tilde{f}$ by the inversion formula

$$f(x) = \frac{1}{C_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{da\,db}{a^2}\,\tilde{f}(a,b)\psi_{a,b}(x) \tag{3}$$

For a proof, see Chapter 5 of Ref. 23. This last formula says that $f(x)$ can be written as a superposition of shifted and dilated wavelets.

The continuous wavelet transform has an energy conservation property that is similar to Parseval's formula for the Fourier transform. The function $f(x)$ and its continuous wavelet transform $\tilde{f}(a, b)$ satisfy

$$\int_{-\infty}^{\infty} |f(x)|^2\,dx = \frac{1}{C_\psi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{da\,db}{a^2}|\tilde{f}(a,b)|^2 \tag{4}$$

The wavelet transform has localization properties. There is a sharp time localization at high frequencies, in marked contrast with Fourier transforms. For example, the wavelet transform of a delta function centered at x_0 is

$$\frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} \psi\left(\frac{x-b}{a}\right)\delta(x-x_0)\,dx = \frac{1}{\sqrt{a}}\,\psi\left(\frac{x_0-b}{a}\right)$$

For a given scale factor a the transform is equal to a scaled and normalized wavelet centered at the location of the delta function.

DISCRETE WAVELET TRANSFORM

The wavelet transform has to be discretized for most applications. One way to approach this is to attempt to directly discretize the continuous wavelet transform and find a discrete version of the reconstruction formula given in Eq. (3). In effect this means replace $\psi_{a,b}$ by $\psi_{m,n}$ with $m, n \in \mathbb{Z}$, where $\mathbb{Z}$ is the set of integers. The appropriate replacements for a and b are (23,29)

$$a = a_0^m, \quad b = nb_0a_0^m, \quad a_0 > 1, \quad b_0 > 0$$

When this is done, it turns out that in the discrete parameter case there is no direct generalization of Eq. (3); however, for certain ψ and appropriate a_0 and b_0 there exist $\tilde{\psi}_{m,n}$ such that

$$f = \sum_{m,n}\langle \psi_{m,n}, f\rangle\tilde{\psi}_{m,n}$$

This leads to the introduction of *frames* and *dual frames*. These represent an alternative to orthonormal bases in a Hilbert space [see Heil and Walnut (33)]. This approach will not be pursued here. We refer the reader to standard references (10,23,29,33).

The approach presented here leads to the construction of orthonormal wavelet expansions for discrete sets of data. We *do not* start with a continuous wavelet and attempt to find a discrete counterpart. We make full use of multiresolution, the idea of looking at something at various scales or resolutions.

Multiresolution Analysis

In this approach another function ϕ plays a fundamental role along with the wavelet function ψ. The simplest possible system that illustrates most of the fundamental properties of these functions is the Haar scaling function and Haar wavelet. In this case the scaling function is the box function illus-

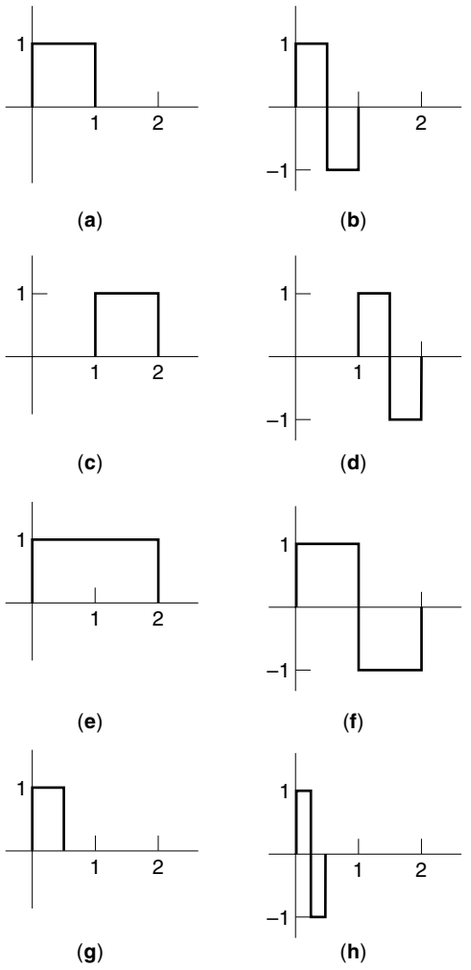


Figure 6. Scaling function ϕ and wavelet function ψ for the Haar system: (a) $\phi(x)$, (b) $\psi(x)$, level 0, basic; (c) $\phi(x-1)$, (d) $\psi(x-1)$, level 0, translated; (e) $\phi(\frac{1}{2}x)$, (f) $\psi(\frac{1}{2}x)$, level 1, basic; and (g) $\phi(2x)$, (h) $\psi(2x)$, level -1, basic.

trated in Fig. 6(a) and the corresponding wavelet is shown in Fig. 6(b). We refer to these two functions as the *level 0* functions. The fundamental idea is to construct other scaling functions and wavelets from dilations and translations of the level 0 functions. Some of these are shown in Fig. 6. Note that scaling by $x \to 2x$ corresponds to a contraction and scaling by $x \to \frac{1}{2}x$ gives an expansion. There are two scaling functions and two wavelet functions at level -1, with support on an interval of length $\frac{1}{2}$,

$$\phi(2x), \quad \phi(2x-1), \quad \psi(2x), \quad \psi(2x-1)$$

Complete families of scaling functions and the wavelets are obtained by appropriate translations and dilations. The functions $\phi(x)$ and $\psi(x)$ are the functions at level 0. The move from $\psi(x)$ to $\psi(2x)$ is a dilation operation, whereas the shift from 0 to 1 is a translation operation. Starting from ϕ and ψ the functions are shifted and compressed. The next level down (level -2) contains

$$\phi(4x), \quad \phi(4x-1), \quad \phi(4x-2), \quad \phi(4x-3);$$
$$\psi(4x), \quad \psi(4x-1), \quad \psi(4x-2), \quad \psi(4x-3)$$

Each of these functions is supported on an interval of length $\frac{1}{4}$. A continuation of this process gives infinite families of functions,

$$\phi_{j,k}(x) = 2^{-j/2}\phi(2^{-j}x-k); \quad \psi_{j,k}(x) = 2^{-j/2}\psi(2^{-j}x-k) \quad (5)$$

with $j, k \in \mathbb{Z}$. For the range of values ($j \le 0$) and ($0 \le k < 2^{-j}$), these functions form a basis over the interval [0, 1].

Important Remark: The Haar system is used for illustration purposes since it is simple and easy to understand. The important point is that all of this holds for other scaling functions and wavelets that have increasing degrees of smoothness. Some of these will be discussed and illustrated later.

Suppose we designate the space spanned by functions of the form $\phi(x-k)$, $k \in \mathbb{Z}$, by V_0 and the space spanned by functions of the form $\phi(2x-k)$, $k \in \mathbb{Z}$, by V_{-1}. Clearly, the function $\phi(x)$ can be written as

$$\phi(x) = \phi(2x) + \phi(2x-1)$$

Since functions in V_0 can be written as a linear combination of functions in V_{-1} we have the condition

$$V_0 \subset V_{-1}$$

This argument can be extended in either direction, for example

$$\phi(\tfrac{1}{2}x) = \phi(x) + \phi(x-1), \quad V_1 \subset V_0$$

An example of the projection of a function onto V_0 and V_1 is shown in Fig. 7. By continuing this process the nesting of the closed subspaces V_j follows,

$$\leftarrow \text{coarser} \ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots \text{finer} \rightarrow \quad (6)$$

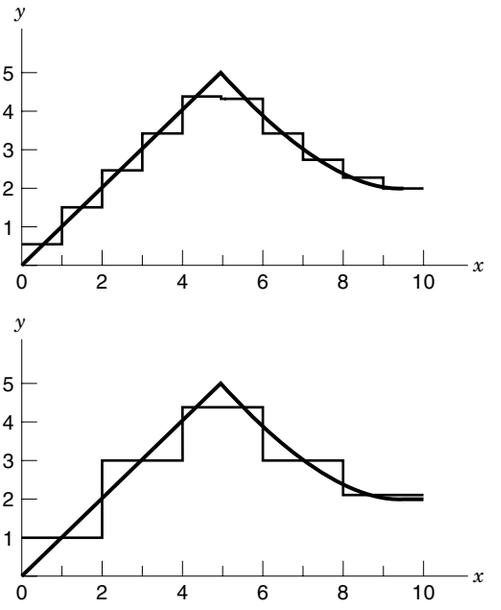


Figure 7. A function $y = f(x)$ (dots) projected onto V_0 (top), projected onto V_1 (bottom).

The nesting order is selected so that the spaces show less detail as the index increases. For example, in Fig. 7 the projection onto V_1 can be considered as a blurred version of the projection onto V_0. This is an agreement with the choice made by Daubechies (29). A caution for the reader is in order on this; about half of the wavelet literature uses the opposite convention, coupled with a change of $-j$ to $+j$ in Eq. (5). Properties of the V_j are summarized by the following definition of an orthogonal multiresolution analysis.

A *multiresolution analysis* of $L^2(\mathbb{R})$ consists of a sequence of closed subspaces V_j, for all $j \in \mathbb{Z}$, such that

(M1) $V_j \subset V_{j-1}$

(M2) $\overline{\bigcup_j V_j} = L^2(\mathbb{R})$ and $\bigcap_j V_j = \{0\}$

(M3) $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j-1}$

(M4) $f(x) \in V_0 \Leftrightarrow f(x-k) \in V_0$ for all $k \in \mathbb{Z}$

(M5) There exists a function $\phi \in V_0$ so that $\phi(x-k), k \in \mathbb{Z}$ form an orthonormal basis for V_0.

Several remarks are in order in regard to this definition. In (M2) the bar over the union is to indicate *closure.* The closure of a set is obtained by including all functions that can be obtained as limits of sequences in the set. This terminology could be replaced by saying that the union is dense in L^2. Condition (M5) is often relaxed by assuming that the set of functions $\phi(x-k)$ is a *Reisz basis* for V_0. For a full treatment of this approach, see Refs. 10 and 34.

Now let us observe that although we have the condition $V_0 \subset V_{-1}$ the basis functions in V_0 are not orthogonal to the basis functions in V_{-1},

$$\int \phi(x)\phi(2x)\,dx \neq 0 \quad \text{and} \quad \int \phi(x)\phi(2x-1)\,dx \neq 0$$

The integrals are over all x where the functions do not vanish. For the Haar case illustrated earlier this would be over the interval [0, 1]. There is a clever way to fix this. Note that we can write

$$\phi(2x) = \tfrac{1}{2}[\phi(x) + \psi(x)] \quad \text{and} \quad \phi(2x-1) = \tfrac{1}{2}[\phi(x) - \psi(x)]$$

If we designate the space spanned by the wavelets $2^{-j/2}\psi(2^{-j}x - k)$, $j, k \in \mathbb{Z}$, by W_j, it follows that the direct sum of subspaces gives

$$V_{-1} = V_0 \oplus W_0$$

Moreover, it is easy to check that basis functions in V_0 are orthogonal to basis functions in W_0. This idea can be extended in either direction

$$V_{-2} = V_{-1} \oplus W_{-1} \quad \text{and} \quad V_0 = V_1 \oplus W_1$$

The space W_j is said to be the *orthogonal complement* of V_j in V_{j-1}. In general we have

$$V_{j-1} = V_j \oplus W_j, \quad W_j \perp W_{j'}, \quad \text{if } j \neq j' \tag{7}$$

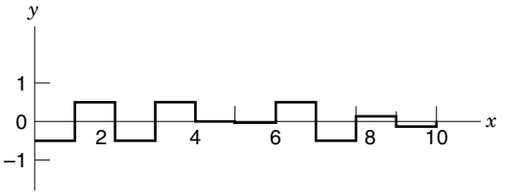


Figure 8. The detail information for the function from Fig. 7 in W_1.

If we designate the projection of $f(x)$ onto V_m by $P_m f$ and the projection of $f(x)$ onto W_m by $Q_m f$, then Eq. (7) implies that

$$P_{j-1} f = P_j f + Q_j f \tag{8}$$

If $j = 1$ this is $P_0 f = P_1 f + Q_1 f$. Projections $P_0 f$ and $P_1 f$ are shown in Fig. 7. The projection $Q_1 f$ contains the difference or *detail* information, illustrated in Fig. 8. Note that this does indeed represent Haar wavelets at level 1.

To see how the general decomposition emerges, consider

$$V_0 = V_1 \oplus W_1 = V_2 \oplus W_2 \oplus W_1 = V_3 \oplus W_3 \oplus W_2 \oplus W_1$$

This could be extended as far as desired. The general formula is

$$V_j = V_J \oplus \bigoplus_{k=0}^{J-j-1} W_{J-k} \tag{9}$$

where all subspaces on the right are orthogonal. This means that any function can be represented as a sum of detail parts plus a smoothed version of the original function. This is often expressed by saying that the function is resolved into a low-frequency part plus a sum of high-frequency parts. To see this, think of Fourier transforms. The broad part in V_J has a Fourier transform concentrated at the origin in the Fourier domain, hence low frequencies. The parts in W_j have Fourier transforms that must vanish at the origin since the area of the wavelet is 0, hence high-frequency parts.

By use of (M2) and Eq. (9) it follows that

$$L^2(\mathbb{R}) = \bigoplus_{j \in \mathbb{Z}} W_j \tag{10}$$

(Note that $V_j \to \{0\}$ as $j \to \infty$.) The collection $\{\psi_{j,k}; j, k \in \mathbb{Z}\}$ is an orthonormal basis for $L^2(\mathbb{R})$. The spaces W_j also have the scaling property (M3), so the job is to find a $\psi \in W_0$ such that the $\psi(x-k)$ form an orthonormal basis for W_0.

Orthonormal Wavelets with Compact Support

Before we embark on the task of determining other acceptable scaling and wavelet functions it may be useful to examine the Haar system more closely. Keep in mind that what we are doing applies to any functions ϕ and ψ that satisfy the multiresolution analysis and decomposition, Eq. (9).

First, we observe that the scaling function $\phi_{j,k}$ from Eq. (5) satisfies an orthogonality condition at the same scale,

$$\langle \phi_{j,k}, \phi_{j,k'} \rangle \equiv \int_{\mathbb{R}} \phi_{j,k}\phi_{j,k'}\,dx = \delta_{kk'} \tag{11}$$

Here $\delta_{kk'}$ is the Kronecker delta defined to be 1 if $k = k'$ and 0 if $k \neq k'$. The wavelets are orthogonal at the same scale and across scales,

$$\langle \psi_{j,k}, \psi_{j',k'} \rangle = \delta_{jj'}, \delta_{kk'} \tag{12}$$

The wavelets and scaling functions also satisfy

$$\langle \phi_{j,k}, \psi_{j',k'} \rangle = 0 \tag{13}$$

We focus our attention on the containment $V_0 \subset V_{-1}$ and $W_0 \subset V_{-1}$ for the Haar system,

$$\phi(x) = \phi(2x) + \phi(2x-1) \quad \psi(x) = \phi(2x) - \phi(2x-1)$$

This is a special case of general expansions for ϕ or ψ where

$$\phi(x) = \sum_{k\in\mathbb{Z}} c_k \phi(2x-k) \quad \psi(x) = \sum_{k\in\mathbb{Z}} d_k \phi(2x-k) \tag{14}$$

For the Haar system there are only two coefficients needed; namely, $c_0 = c_1 = 1$. The d coefficients are found from these. We will see later that the condition is $d_k = (-1)^k\, c_{1-k}$. One way to obtain the coefficients for more general functions ϕ and ψ is to place constraints on the coefficients c_k in the expansion of the scaling function. The method here follows the pioneer work on this by Ingrid Daubechies (30).

The expansion for $\phi(x)$ in Eq. (14) is called a *dilation equation.* If only a finite number of the coefficients are nonzero, then ϕ must vanish outside a finite interval. This gives the property of *compact support.* Suppose the nonzero coefficients are $c_m, c_{m+1}, \ldots, c_n$. If the original function ϕ has support on the interval $[a, b]$, then $\phi(2x)$ has support on the interval $[a/2, b/2]$. The shifted function $\phi(2x - k)$ has support on $[a + k/2, b + k/2]$. Since the index k goes from m to n we have

$$\phi(x) = \sum_{k=m}^{n} c_k \phi(2x-k) \tag{15}$$

The support on the left side is related to the support on the right side by

$$[a, b] = \left[\frac{a+m}{2}, \frac{b+n}{2}\right]$$

This requirement yields $a = m$ and $b = n$, hence the support is on $[m, n]$.

Example: Suppose $\phi(x)$ is the level 0 Haar box function, and let the sum go from 0 to n. If this function is substituted and used on the right side of Eq. (15) then the function on the left has support on $[0, 1 + n/2]$. If this function is now substituted on the right side the function on the left has support on $[0, 1 + 3n/4]$. If this procedure is continued the limiting case is just the interval $[0, n]$.

A *consistency condition* can be established by integrating the dilation equation. This is easy and the details that involve a change of variables ($t = 2x - k$) are left for the reader,

$$\int_{-\infty}^{\infty} \phi(x)\, dx = \int_{-\infty}^{\infty} \sum_k c_k \phi(2x-k)\, dx = \cdots = \frac{1}{2}\sum_k c_k \int_{-\infty}^{\infty} \phi(t)\, dt$$

Since the integral of the scaling function is assumed to be finite there is a requirement that

$$\sum_k c_k = 2 \tag{16}$$

A convenient choice for the normalization on ϕ is such that

$$\int_{-\infty}^{\infty} \phi(x)\, dx = 1$$

Caution: Some authors use a slightly different convention for the constants. The other popular choice is to use $c_n = \sqrt{2}\, h_n$, where h_n corresponds to the notation used by Daubechies (29).

The *orthogonality* condition in Eq. (11) leads to another important relation. The reader may wish to see Alpert (35) for details.

$$\begin{aligned}
\delta_{kl} &= \int_{-\infty}^{\infty} \phi(x-k)\phi(x-l)\, dx \\
&= \int_{-\infty}^{\infty} \sum_m c_m \phi[2(x-k)-m] \sum_n c_n \phi[2(x-l)-n]\, dx \\
&= \frac{1}{2}\sum_{m,n} c_m c_n \delta_{2k+m,2l+n} \\
&= \frac{1}{2}\sum_m c_m c_{2k-2l+m}
\end{aligned}$$

Since the sum is over all $m \in \mathbb{Z}$ we can make the change of index $m \to m + 2l$. This leads to the desired orthogonality condition

$$\sum_{m\in\mathbb{Z}} c_{2k+m} c_{2l+m} = 2\delta_{kl} \tag{17}$$

This equation ensures the orthogonality of the translates of the scaling function.

The coefficients d_k must be selected so that an orthogonality condition holds for the translates of the wavelet function $\psi(x)$. It is easy to show that this works for

$$d_k = (-1)^k c_{1-k} \tag{18}$$

The calculation makes use of Eqs. (14) and (17)

$$\begin{aligned}
&\int_{-\infty}^{\infty} \psi(x-k)\psi(x-l)\, dx \\
&\qquad = \sum_{m,n} \int_{-\infty}^{\infty} d_m \phi[2(x-k)-m] d_n \phi[2(x-l)-n]\, dx \\
&\qquad = \frac{1}{2}\sum_m d_{2k+m} d_{2l+m} \\
&\qquad = \frac{1}{2}\sum_m (-1)^{2k+m} c_{1-2k-m} (-1)^{2l+m} c_{1-2l-m} \\
&\qquad = \frac{1}{2}\sum_m c_{1-2k-m} c_{1-2k-l} \\
&\qquad = \delta_{kl}
\end{aligned}$$

Also, the choice made in Eq. (18) is adequate to establish the orthogonality

$$\int_{-\infty}^{\infty} \phi(x-k)\psi(x-l)\,dx = 0$$

This is left as an exercise; observe that you do not have to make use of Eq. (17).

The key conditions thus far are Eqs. (16), (17), and (18). These are not adequate for a unique determination of the coefficients that lead to the family of Daubechies that extend the Haar system in a natural way. The next condition relates to *approximation.*

The idea is to approximate polynomials of degree $j = 0, 1, \ldots, N-1$ as linear combinations of translates of the scaling function in V_0. Thus, we look for coefficients α such that

$$x^j = \sum_{k\in\mathbb{Z}} \alpha_{j,k}^N \phi(x-k), \quad (j = 0, 1, \ldots, N)$$

By orthogonality

$$\alpha_{j,k}^N = \int_{-\infty}^{\infty} x^j \phi(x-k)\,dx$$

The scaling function depends on N and is often written as ${}_N\phi$. Here we suppress the N and just use ϕ. Recall that only two coefficients are needed for the Haar scaling function. In this case polynomials of degree $N = 0$ can be represented with no error by scaling functions in V_0. For many situations as smoother scaling function is desired. We are looking for the conditions that must hold when we allow more than two coefficients, and require that polynomials of higher degree be represented exactly by functions in V_0.

The space V_0 is orthogonal to W_0; consequently, for $j = 0, \ldots, N-1$,

$$\int_{-\infty}^{\infty} x^j \psi(x)\,dx = 0$$

Now, use Eq. (14) along with the trick (identity)

$$x^j = \left(\frac{2x-k+k}{2}\right)^j$$

This yields

$$\int_{-\infty}^{\infty} \sum_k \left(\frac{2x-k+k}{2}\right)^j d_k \phi(2x-k)\,dx = 0$$

The general binomial expansion

$$(a+b)^j = \sum_{r=0}^{j} \binom{j}{r} a^{j-r} b^r, \quad \binom{j}{r} = \frac{j!}{r!(j-r)!}$$

can be applied to give

$$2^{-j} \sum_{r=0}^{j} \binom{j}{r} \sum_k k^{j-r} d_k \int_{-\infty}^{\infty} (2x-k)^r \phi(2x-k)\,dx = 0$$

The change of variables $2x - k \to x$ leads to

$$2^{-j-1} \sum_{r=0}^{j} \binom{j}{r} \sum_k k^{j-r} d_k \int_{-\infty}^{\infty} x^r \phi(x)\,dx = 0$$

The integral over x cannot be zero since by assumption x^r can be written as a linear combination of translates of ϕ for $r = 0, \ldots, N-1$. It follows that we must require

$$\sum_{r=0}^{j} \binom{j}{r} \sum_k k^{j-r} d_k = 0$$

hold for individual values of j from 0 to $N-1$. If you write this out for $j = 0$ then for $j = 1$, and $j = 2$ you see that the condition is

$$\sum_k k^j d_k = 0, \quad (j = 0, \ldots, N-1)$$

This is usually written in terms of the c_j coefficients from Eq. (18) with a slight modification. The index is usually shifted so the nonzero coefficients range from 0 to $2N-1$ for the Daubechies coefficients (29). This is accomplished by using the connection

$$d_k = (-1)^k c_{2N-1-k}$$

Then, the approximation condition becomes

$$\sum_{k=0}^{2N-1} (-1)^k k^j c_{2N-1-k} = 0, \quad (j = 0, \ldots, N-1) \tag{19}$$

Examples. The key equations are Eqs. (17)–(19). There are two coefficients for $N = 1$ that satisfy the conditions

$$c_0^2 + c_1^2 = 2, \quad c_0 - c_1 = 0$$

with region of support [0, 1]. These are the familiar Harr coefficients $c_0 = c_1 = 1$. For $N = 2$ we have four coefficients, known as the D4 coefficients. They satisfy orthogonality conditions

$$c_0^2 + c_1^2 + c_2^2 + c_3^2 = 2 \quad \text{and} \quad c_0 c_2 + c_1 c_3 = 0$$

and approximation conditions (for $j = 0$ and $j = 1$)

$$c_3 - c_2 + c_1 - c_0 = 0 \quad \text{and} \quad 0c_3 - 1c_2 + 2c_1 - 3c_0 = 0$$

The solution is unique up to a left–right reversal ($c_0 \leftrightarrow c_3$, $c_1 \leftrightarrow c_2$)

$$c_0 = (1+\sqrt{3})/4 \quad c_1 = (3+\sqrt{3})/4$$
$$c_2 = (3-\sqrt{3})/4 \quad c_3 = (1-\sqrt{3})/4$$

Note that Eq. (16) is also satisfied by these D4 coefficients. Keep in mind that if the other popular normalization condition is used $c_n = \sqrt{2}\,h_n$, then each of these coefficients must be divided by $\sqrt{2}$. When this is done then the sum of the squares is 1 rather than 2.

The region of support for the D4 scaling function and the wavelet function is [0, 3]. The graphs of these are shown in

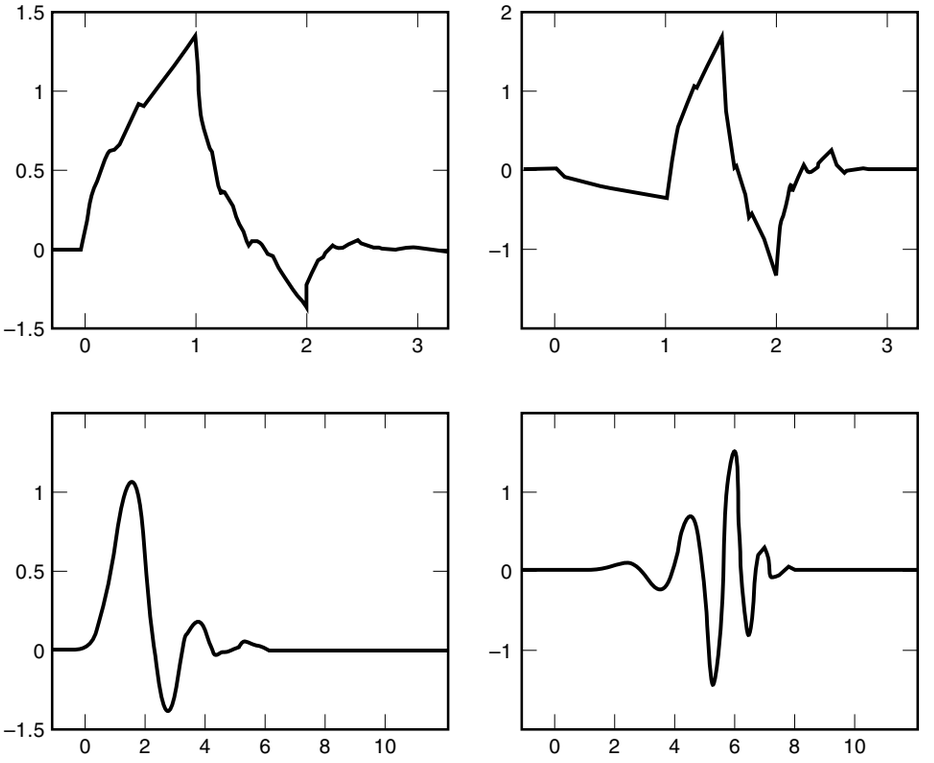


Figure 9. D4 scaling function (top left), D4 wavelet (top right), D12 scaling function (bottom left), and D12 wavelet (bottom right).

Fig. 9 across the top. The graphs for $N = 6$ (D12) are shown across the bottom. Here the region of support is [0, 11]. Note that as the number of coefficients increases the graphs get smoother and the region of support increases. Tables of coefficients for various values of N are given by Daubechies (29).

The functions in Fig. 9 are interesting, but knowing what these functions look like is absolutely unnecessary for implementation of the wavelet transform. The coefficients are all you need, coupled with an algorithm. An example is given in the section on *Mechanics of Doing Transforms,* and a method for obtaining Fig. 9 is indicated.

Other Wavelets. We have only touched the surface by indicating how to find the family of Daubechies wavelets. If the orthogonality and approximation conditions are modified, other sets of coefficients follow. For example, if you impose conditions of vanishing moments on ϕ as well as ψ then the resulting wavelets are known as *coiflets,* after a suggestion by Ronald R. Coifman of Yale University. For more information on these see Refs. 29, 36, and 37. Another example is provided by *biorthogonal wavelets.* The filter coefficients for the reconstruction are not the same as those for the decomposition, and there are two dual wavelet bases associated with two different multiresolution ladders. This leads to symmetric wavelets that are an advantage for some applications. Important references on these are by Cohen and Daubechies (38): Cohen, Daubechies, and Feauveau (39): and Vetterli and Herley (40).

Fourier Space Methods. Very powerful methods for finding wavelet coefficients are provided by Fourier techniques. These techniques can be used to find the family of Daubechies wavelets; also, they form a foundation for finding wavelets with other important properties. Here, we only indicate how this can be started and then refer the reader to some excellent references where this approach is utilized.

Start with the dilation equation for the scaling function

$$\phi(x) = \sum_k c_k \phi(2x - k)$$

The Fourier transform of this equation is

$$\hat{\phi}(\xi) = \sum_k c_k \int_{-\infty}^{\infty} \phi(2x - k) e^{-i\xi x}\, dx$$

The change of variables $t = 2x - k$ gives

$$\hat{\phi}(\xi) = \frac{1}{2} \sum_k c_k e^{-ik\xi/2} \int_{-\infty}^{\infty} \phi(t) e^{-i\xi t/2}\, dt$$

Observe that the integral is just $\hat{\phi}(\xi/2)$. This yields

$$\hat{\phi}(\xi) = m_0\left(\frac{\xi}{2}\right) \hat{\phi}\left(\frac{\xi}{2}\right)$$

where, in keeping with the notation of Daubechies (29), we define

$$m_0(\xi) \equiv \frac{1}{2} \sum_k c_k e^{-ik\xi}$$

Note that $m_0(0) = 1$ follows from

$$m_0(0) = \frac{1}{2} \sum_k c_k e^0 = \frac{1}{2} \sum_k c_k = 1$$

If we make the replacement $\xi \to \xi/2$ then

$$\hat{\phi}\left(\frac{\xi}{2}\right) = m_0\left(\frac{\xi}{4}\right)\hat{\phi}\left(\frac{\xi}{4}\right)$$

and

$$\hat{\phi}(\xi) = m_0\left(\frac{\xi}{2}\right)m_0\left(\frac{\xi}{4}\right)\hat{\phi}\left(\frac{\xi}{4}\right) = \left[\prod_{j=1}^{2} m_0\left(\frac{\xi}{2^j}\right)\right]\hat{\phi}\left(\frac{\xi}{2^2}\right)$$

Clearly this procedure can be continued to give

$$\hat{\phi}(\xi) = \left[\prod_{j=1}^{N} m_0\left(\frac{\xi}{2^j}\right)\right]\hat{\phi}\left(\frac{\xi}{2^N}\right)$$

As $N \to \infty$, $\hat{\phi}(\xi/2^N) \to \hat{\phi}(0) = 1$, since the area under the scaling function is normalized to 1. This means that as $N \to \infty$ the infinite product goes to the Fourier transform of the scaling function,

$$\hat{\phi}(\xi) = \prod_{j=1}^{\infty} m_0\left(\frac{\xi}{2^j}\right)$$

Example. Let us investigate how this works for the box function, the scaling function for the Haar case. If $c_0 = c_1 = 1$, then

$$m_0\left(\frac{\xi}{2}\right) = \frac{1}{2}\left(1 + e^{-i\xi/2}\right)$$

and

$$\begin{aligned} m_0\left(\frac{\xi}{2}\right)m_0\left(\frac{\xi}{4}\right) &= \frac{1}{2^2}(1 + e^{-i\xi/2})(1 + e^{i\xi/4}) \\ &= \frac{1}{2^2}(1 + e^{-i\xi/4} + e^{-2i\xi/4} + e^{-3i\xi/4}) \end{aligned}$$

The part in parenthesis on the right is just the sum of $2^2 = 4$ terms of a geometric series where the first term is 1 and the ratio term r is $e^{-i\xi/4}$. The sum of n terms is given by $(1 - r^n)/(1 - r)$. Thus

$$m_0\left(\frac{\xi}{2}\right)m_0\left(\frac{\xi}{4}\right) = \frac{1}{2^2}\frac{1 - e^{i\xi}}{1 - e^{-i\xi/4}}$$

In the general case where there are 2^j terms, the result is

$$m_0\left(\frac{\xi}{2}\right)\ldots m_0\left(\frac{\xi}{2^j}\right) = \frac{1}{2^j}\frac{1 - e^{i\xi}}{1 - e^{-\xi/2^j}}$$

Now let $2^{-j} = x$, then

$$2^j(1 - e^{-i\xi/2^j}) = \frac{1 - \cos x\xi}{x} + i\frac{\sin x\xi}{x}$$

In the limit as $j \to \infty$, and $x \to 0$ we get $i\xi$. It follows that

$$\prod_{j=1}^{\infty} m_0\left(\frac{\xi}{2^j}\right) = \frac{1 - e^{-i\xi}}{i\xi}$$

This is just the Fourier transform of ϕ where ϕ is the box function,

$$\hat{\phi}(\xi) = \int_{-\infty}^{\infty} \phi(x)e^{-i\xi x}\,dx = \int_0^1 e^{-\xi x}\,dx = \frac{1 - e^{-i\xi}}{i\xi}$$

just as expected.

A rich resource of information about wavelets comes from using Fourier techniques. The books by Hernández and Weiss (10), Vetterli and Kovačević (23), Strang and Nguyen (41), and Daubechies (29) are excellent sources.

MECHANICS OF DOING THE TRANSFORM

This example of how the wavelet transform can be implemented using matrices will be of value to those who wish to acquire an intuitive understanding about how the transform works. This is for illustration only, since in practice efficient code may not be written in matrix form. This example is for the simplest case, the Haar; however, the extension to smoother cases is easy and we will indicate how following this example. The following operations are illustrated:

1. Generate the wavelet coefficients with down sampling.
2. Show how this is a dual filter operation with a shrinking matrix and signal.
3. Mechanics of the reconstruction, the inverse transform.

There is a pyramidal structure to the procedure. At each level the detail information is stored, while the smooth information may be transformed at the next higher scale. One way to indicate this is shown in Fig. 10, where we have carried the transform through three stages.

Let the transpose of the original signal vector for an eight-point transform be designated by

$$[16, 32, 64, 16, 6, 32, 16, 8]$$

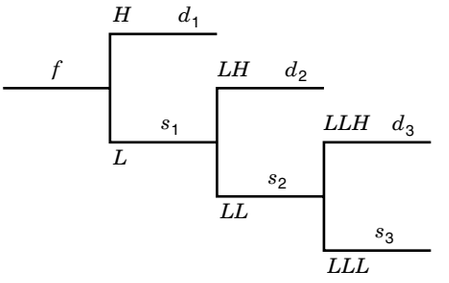


Figure 10. Pyramidal decomposition of a signal. The low- and high-pass parts are indicated by L and H. The corresponding smooth and detail parts are designated by s and d with subscripts indicating the level.

The full smoothing operator (the low pass part) with $c_0 = c_1 = 1$ is given by

$$S = \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The shift on the last row is to take into account edge effects, and is essential to insure that the inversion is exact. The highpass operator associated with the detail is given by

$$H = \frac{1}{2}\begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Now we calculate the detail and smooth coefficients that lie in W_1 and V_1, respectively

$$d_1 = Hf \downarrow = [-8, -16, 24, 4, -12, 8, 4, -4] \downarrow = [-8, 24, -12, 4]$$

and

$$s_1 = Sf \downarrow = [24, 48, 40, 12, 20, 24, 12, 12] \downarrow = [24, 40, 20, 12]$$

The use of the down arrow is to indicate down sampling. Every other value is discarded. You might think information has been lost by doing this, but note that you started with eight independent values in the signal and after down sampling you still have eight independent values, four detail and four smooth coefficients. These can be used to recover the original values. The next step is to contract the matrices S and H,

$$S = \frac{1}{2}\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \qquad H = \frac{1}{2}\begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

The coefficients for W_2 and V_2 follow by applying these new contracted matrices to the s_1 vector,

$$d_2 = Hs_1 \downarrow = [-8, 10, 4, -6] \downarrow = [-8, 4]$$
$$s_2 = Ss_1 \downarrow = [32, 30, 16, 16] \downarrow = [32, 16]$$

Once again we contract S and H,

$$S = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \qquad H = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

The last two coefficients are found as before

$$d_3 = Hs_2 \downarrow = [8, -8] \downarrow = [8], \qquad s_3 = Ss_2 \downarrow = [24, 24] \downarrow = [24]$$

This completes the eight-point transform. The eight points in the signal vector have been transformed by the Haar wavelet transform to eight points,

$$d_1 = [-8, 24, -12, 4] \quad d_2 = [-8, 4] \quad d_3 = [8] \quad s_3 = [24]$$

The inverse transform must start with the wavelet coefficients and end with the original signal coefficients. This is done by a clever reversal of the directions in Fig. 10, with a sum used to go from two branches on the right to one on the left, at a vertex where three lines meet. Here is how it works. Use the transpose of S and H without the factor of $\frac{1}{2}$ at each step and insert zeros where there were discarded values. This upsampling is indicated by the up arrow.

$$S^\dagger = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \qquad H^\dagger = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$
$$S^\dagger s_3 \uparrow = S^\dagger \begin{pmatrix} 24 \\ 0 \end{pmatrix} = \begin{pmatrix} 24 \\ 24 \end{pmatrix} \qquad H^\dagger d_3 \uparrow = H^\dagger \begin{pmatrix} 8 \\ 0 \end{pmatrix} = \begin{pmatrix} 8 \\ -8 \end{pmatrix}$$

The s_2 signal is recovered by addition

$$\begin{pmatrix} 24 \\ 24 \end{pmatrix} + \begin{pmatrix} 8 \\ -8 \end{pmatrix} = \begin{pmatrix} 32 \\ 16 \end{pmatrix}$$

At the next step we have

$$S^\dagger = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \qquad H^\dagger = \begin{pmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$
$$S^\dagger s_2 \uparrow = S^\dagger \begin{pmatrix} 32 \\ 0 \\ 16 \\ 0 \end{pmatrix} = \begin{pmatrix} 32 \\ 32 \\ 16 \\ 16 \end{pmatrix}$$
$$H^\dagger d_2 \uparrow = H^\dagger \begin{pmatrix} -8 \\ 0 \\ 4 \\ 0 \end{pmatrix} = \begin{pmatrix} -8 \\ 8 \\ 4 \\ -4 \end{pmatrix}$$

Again by addition we recover the s_1 signal,

$$\begin{pmatrix} 32 \\ 32 \\ 16 \\ 16 \end{pmatrix} + \begin{pmatrix} -8 \\ 8 \\ 4 \\ -4 \end{pmatrix} = \begin{pmatrix} 24 \\ 40 \\ 20 \\ 12 \end{pmatrix}$$

In the final step we are back to the full matrices

$$S^{\dagger} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$H^{\dagger} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

The up sampling gives

$$S^{\dagger} s_1 \uparrow = S^{\dagger} \begin{pmatrix} 24 \\ 0 \\ 40 \\ 0 \\ 20 \\ 0 \\ 12 \\ 0 \end{pmatrix} = \begin{pmatrix} 24 \\ 24 \\ 40 \\ 40 \\ 20 \\ 20 \\ 12 \\ 12 \end{pmatrix} \qquad H^{\dagger} d_1 \uparrow = H^{\dagger} \begin{pmatrix} -8 \\ 0 \\ 24 \\ 0 \\ -12 \\ 0 \\ 4 \\ 0 \end{pmatrix} = \begin{pmatrix} -8 \\ 8 \\ 24 \\ -24 \\ -12 \\ 12 \\ 4 \\ -4 \end{pmatrix}$$

The original signal vector is recovered by addition

$$f = \begin{pmatrix} 24 \\ 24 \\ 40 \\ 40 \\ 20 \\ 20 \\ 12 \\ 12 \end{pmatrix} + \begin{pmatrix} -8 \\ 8 \\ 24 \\ -24 \\ -12 \\ 12 \\ 4 \\ -4 \end{pmatrix} = \begin{pmatrix} 16 \\ 32 \\ 64 \\ 16 \\ 8 \\ 32 \\ 16 \\ 8 \end{pmatrix}$$

This concludes the Haar example; however, some additional things should be observed. It is possible to combine the matrix multiplication and the up and down sampling. For a discussion of this see Strang and Nguyen (41). Also, one can combine the operations of finding the d and s parts along with the down sampling. A practical example of this is contained in Ref. 42, section 13.10, for the Daubechies D4 wavelet with four coefficients. In addition to Ref. 42 other sources of code for efficient implementation of the forward and inverse wavelet transform include Bruce and Gao (43), and Cody (44,45). Also, see the section on Wavelet Resources on the Internet. Finally, note that if we start with

$$d_1 = [0, 0, 0, 0] \qquad d_2 = [0, 0] \qquad d_3 = [1] \qquad s_3 = [0]$$

and apply the inverse transform we get back the wavelet function

$$[1, 1, 1, 1, -1, -1, -1, -1]$$

This is one way to obtain the wavelets illustrated in Fig. 9. We simply run a unit vector, made up of 0's except for a 1 in a single location through the inverse transform.

OCTAVE BAND TREE STRUCTURE

The type of division of the spectrum for the tree structure of Fig. 10 is known as a *dyadic* or *octave* band. The part labeled s is the low-pass part and the part labeled d is the high-pass part. At each level of the tree the lower half of the spectrum is split into two equal bands. In Fourier space this can be represented by Fig. 11. For an extensive discussion of tree structures and the corresponding frequency band splits, see Akansu and Haddad (46). Another important type of tree structure for wavelet analysis is that used in connection with wavelet packets and best basis algorithms pioneered by Coifman and Wickerhauser (47,48) and Wickerhauser (49). In this type of tree there is an option along both the high-pass and low-pass branches to send the signal through more high-pass and low-pass filters. This is part of an important and extensive area of wavelet theory known as *adaptive wavelet transform methods.* For a full discussion we refer the reader to Refs. 47–49 and the Reading List.

An extension of the octave band tree structure to 2-D was suggested by Burt and Adelson (24). The technique goes by the name of the *Laplacian pyramid.* The multiresolution analysis can be extended to 2-D for functions $f(x, y)$, for details see Daubechies (29). We define a scaling function of two variables and three wavelets. These come from tensor products of horizontal and vertical 1-D wavelets. Here superscripts s, h, v, and d refer to smooth, horizontal, vertical, and diagonal, respectively.

$$\begin{aligned} \Phi^{s}(x, y) &= \phi(x)\phi(y) \quad L(x)L(y) \\ \Psi^{h}(x, y) &= \phi(x)\psi(y) \quad L(x)H(y) \\ \Psi^{v}(x, y) &= \psi(x)\phi(y) \quad H(x)L(y) \\ \Psi^{d}(x, y) &= \psi(x)\psi(y) \quad H(x)H(y) \end{aligned}$$

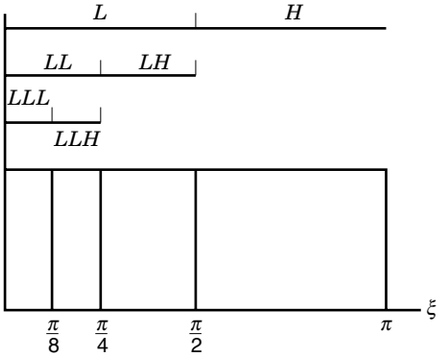


Figure 11. Relation of positive part of frequency spectrum to ideal high- and low-pass parts from Fig. 10.

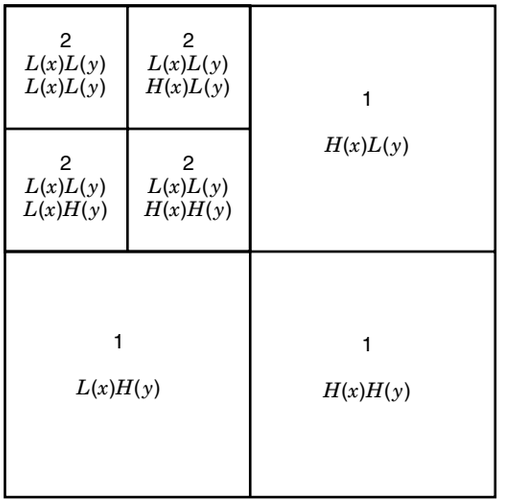


Figure 12. Decomposition of the 2-D transform into two levels. To go to the next level the low-pass part in the upper left is further broken down just as going from level 1 to level 2.

This leads to a decomposition at levels 1 and 2 illustrated by Fig. 12. At level 2 the smooth part from level 1 is further divided to produce the parts in the upper left corner. To go to level 3 the upper left smooth–smooth part would be further broken down as in going from level 1 to level 2. This can go on as far as is practical. In an image, horizontal edges show prominently in the Ψ^h part, vertical edges in the Ψ^v part, and diagonal edges in the Ψ^d part. See Fig. 13 and the discussion and images in Chapter 10 of Ref. 29.

SOME INTERESTING APPLICATIONS

The range of fields, both pure and applied, where wavelets have had an impact is wide. The disciplines include mathematics, physics, geophysics, fluid dynamics, engineering, computer science, and medicine. The broad list of topics include Fourier analysis, approximation theory, numerical analysis, functional analysis, operator theory, group representations, fractals, turbulence, signal processing, image processing, medical imaging, various types of compressions, speech and audio, image, and video. In this section we give a brief introduction to some of these applications and provide the reader with references to current literature for further study.

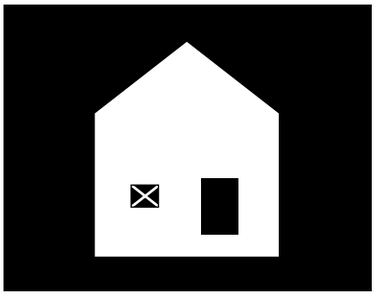
Compression

In many cases a digitized image contains more information than is needed to convey the message the image carries. In these cases we want to remove some of the information in the original image without degrading the quality too much; this is called *lossy compression*. This modified image can be stored more economically and can be transmitted more rapidly, using less bandwidth over a communications channel. Wavelets have been used for these kinds of problems with striking success. We illustrate this in Fig. 14. The original image is upper left. To obtain the image upper right we performed a wavelet transform of the original image, kept 25% of the coefficients with the largest magnitude, replaced the other 75% with zeros, then did the inverse transform. The resulting image is clearly degraded, but not significantly. On the lower left we did the same thing with 6.25% of the coefficients, and on the lower right with only 1.56%.

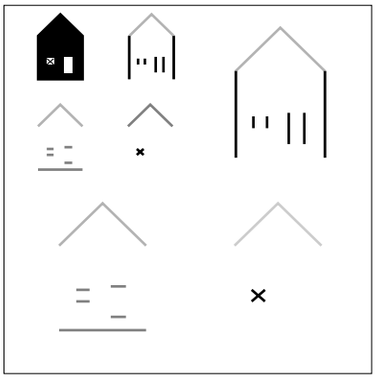
There is an enormous amount of literature on compression. Here we suggest only a few recent articles. These contain guidance to earlier work. Uses of wavelet transform maxima in signal and image processing are described by Mallat and Zhong (50) and Mallat (51). Some new ideas on optimal compression are discussed by Hsiao, Jawerth, Lucier, and Yu (52) and DeVore, Jawerth, and Lucier (53). See Ref. 23 for a general discussion of video compression, and speech and audio compression. Acoustic signal compression with wavelet packets and a comparison of compression methods are given by Wickerhauser (54,55), and some general theorems on optimal bases for data compression are developed by Donoho (56).

Turbulence

Wavelet analysis has provided a new means for examining the structure of turbulent flow. They are especially useful when it is important to obtain some information about the spatial structure of the flow. Some of the pioneer work in this area along with a comparison of older methods is in the review by Marie Farge (57). Also, see the paper on wavelets and turbulence by Farge, Kevlahan, Perrier, and Goirand (58) for a discussion of the main applications of wavelets and wavelet packets to analyze, model, and compute turbulent flows. Wavelet spectra of buoyant atmospheric turbulence are analyzed by Mayer, Hudgins, and Friehe (59) and an experimen-



(**a**)



(**b**)

Figure 13. (a) House to be decomposed using wavelet transform. (b) Decomposition through level 2. The gray scale has been reversed and rescaled to emphasize the important features. Note the prominence of the vertical, horizontal, and diagonal parts in the appropriate locations.

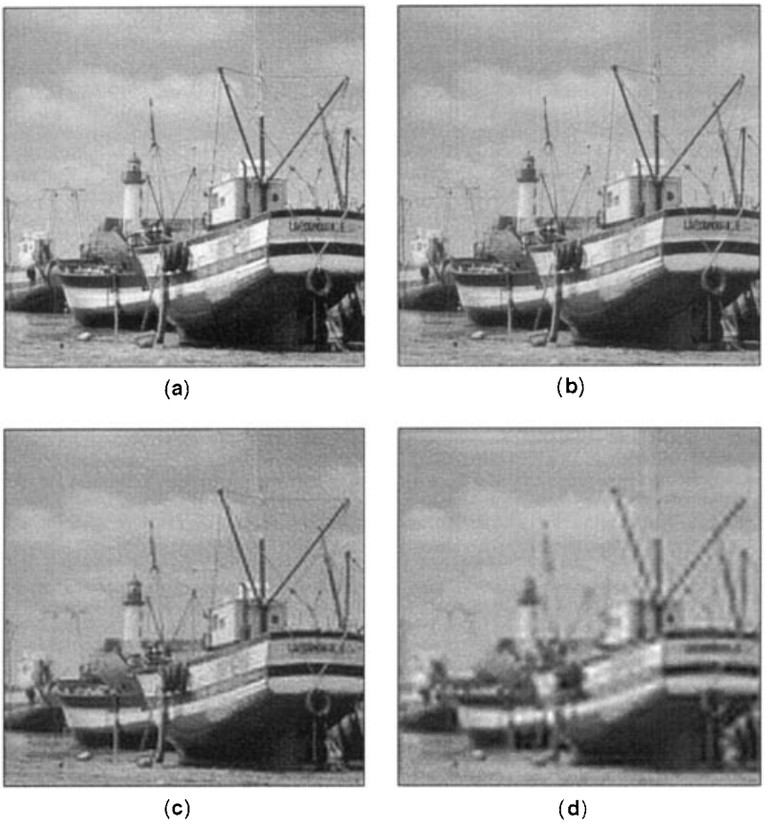


Figure 14. Boat figure to illustrate compression: (a) original, (b) use largest 25%, (c) use largest 6.25%, (d) use largest 1.56%. The degradation can be seen as fewer and fewer coefficients are used to reconstruct the boat.

tal study of inhomogeneous turbulence in the lower troposphere using wavelet analysis is discussed by Druilhet et al. (60). Wickerhauser et al. (61) compare methods for compression of a two-dimensional turbulent flow, and find that the wavelet packet representation is superior to the local cosine representation.

Fractals

The wavelet transform is valuable for the efficient representation of scale-invariant signals. Fractal geometry is being used more and more to describe processes that do not fit naturally into traditional Euclidean geometry. Many fractals of interest have structure that is *similar* on different scales. These properties of wavelets and fractals lead to important foundations for scale-invariant signal models. The discrete wavelet transform algorithm is a key component for practical processing of scale-invariant signals, and for estimating fractal dimensions. Signal processing with fractals using wavelets is an emerging area, one that is exciting with much work remaining to be done. An important resource in this field is the book by Wornell (62). The local self-similarity aspect of fractals and the analysis through wavelet transforms is discussed by Holschneider (34), Chapter 4. These two references and the brief review by Hazewinkel (63) provide guidance to the rich literature in this field.

Medicine and Biology

Wavelets are playing an important role in many areas of medicine and biology. A review of one-dimensional processing in bio-acoustics, electrocardiography (ECG), and electroencephalography (EEG) is given by Unser and Aldroubi (64). This article also contains a brief review of biomedical image processing. Applications of importance include: *noise reduction* in magnetic resonance images (65) using methods systematized by Donoho and Johnstone (66,67) and DeVore and Lucier (68), *image enhancement and segmentation* in digital mammography to accentuate and detect image features that are clinically relevant (69–71), and *image restoration* to restore degradation due to photon scattering and collimator photon penetration with the gamma camera (72). A general strategy for extraction of microcalcification clusters in digitized mammograms making use of wavelets is outlined by DeVore, Lucier, and Yang (73), and a multiresolution statistical method for the identification of normal mammograms with respect to microcalcifications has been developed (74); the key to the method is the recognition of the statistical properties of the various levels of the wavelet decomposition. In *computer-assisted tomography* (CT) the Radon transform (75) is fundamental to the algorithms for reconstruction from projections. Several authors have successfully combined wavelet methods with Radon methods to obtain improved algorithms for certain areas of CT (see Refs. 64 and 76 and citations therein). Other medical applications are to *magnetic resonance imaging* (MRI) (77) and functional neuroimaging using *positron emission tomography* (PET) and *functional MRI* (fMRI). A review is given by Unser and Aldroubi (64).

Others

There are several other areas of application where the wavelet transform plays a key role. We have added a reading list

at the end of the bibliography. A quick survey of this list will provide guidance to good starting points for various applications.

WAVELETS ON THE INTERNET

An increasing amount of wavelet resources are available on the Internet. Preprints of academic papers are available on the Internet long before they appear in print. Many researchers maintain Internet sites where they post their papers, software, and tutorial guides. In fact the World Wide Web (Web), the graphical interface of the Internet, was created by Tim Berners-Lee while he was at the CERN particle physics laboratory in Geneva. The particle physics community has pioneered in the use of the Internet and the Web in the exchange of ideas, abstracts, and papers since 1991.

A similar effort has been made by Wim Sweldens who founded the *Wavelet Digest* in 1992. The *Wavelet Digest* is a free monthly newsletter, edited by Sweldens, available to subscribers by e-mail. One can browse through past issues of the digest at the Wavelet Digest home page (http://www.wavelet.org/wavelet/index.html). The *Wavelet Digest* carries announcements of papers, books, conferences, and seminars in the field of wavelets. It is also a forum for subscribers to ask questions they have about wavelets. Given the wide reach of the *Wavelet Digest* someone is likely to have an answer for almost any question.

The Collection of Computer Science Bibliographies at the Department of Computer Science of the University of Karlsruhe, Germany has a *Bibliographies on Wavelets* (http://liinwww.ira.uka.de/bibliography/Theory/Wavelets/). This is a fairly comprehensive collection of references, although many of them are not available on the Web.

The MathSoft Wavelet Resources page (http://www.mathsoft.com/wavelets.html) has a list of links to preprints and papers available on the Web.

Most wavelet pages on the Web have links to other wavelet resources on the Web. The wavelet page maintained by Andreas Uhl at the Department of Mathematics at the University of Salzburg, Austria (http://www.mat.sbg.ac.at/~uhl/wav.html) has a useful list of links to wavelet pages, and the Amara Graps wavelet page (http://www.amara.com/current/wavelet.html) has a comprehensive list of wavelet links. Also the Amara Graps page provides a list of wavelet software available on the Internet along with a brief description of each software listed.

The WaveLab software for Matlab written by David Donoho, Iain Johnstone, Jonathan Buckheit, and Shaobing Chen at the Stanford University Statistics Department along with Jeffrey Scargle at NASA-Ames Research Center is available at (http://stat.stanford.edu/~wavelab/). This site includes Macintosh, Unix, and PC versions of the software including instructions on how to download and install the software.

MathWorks, the creators of Matlab, have introduced the Wavelet Toolbox. The Wavelet Toolbox is written by Michael Misiti, Yves Misiti, Georges Oppenheim, and Jean-Michel Poggi, who are all members of "Laboratoire de Mathmatiques," Orsay-Paris 11 University, France. The book *Wavelets and Filter Banks,* by Gilbert Strang and Truong Nguyen (41), comes with the Toolbox. The exercises and examples in the book complement the Wavelet Toolbox. The Toolbox has both graphical user interface (GUI) and command line routines. Information on the toolbox can be found at the Matlab Wavelet Toolbox site (http://www.mathworks.com/products/wavelettbx.shtml).

An example of a wavelet application in the real world can be found at the Federal Bureau of Investigation (FBI) fingerprint image compression standard website (http://www.c3.lanl.gov/~brislawn/FBI/FBI.html). The FBI selected a wavelet standard for digitized fingerprints, and this site gives some of the reasons behind the choice.

Another interesting site on the Web is the Jelena Kovačević Bell Labs Wavelet Group page which includes a link to wavelet related Java applets (http://cm.bell-labs.com/who/jelena/Wavelet/w_applets.html).

This list is far from comprehensive. The interested reader can find wavelet-related links at these sites or by searching on any of the Internet search engines. There are some things one must keep in mind while browsing the Web. Not all information on the Web has been screened by any rigorous peer-review process. One must check the provenance of the information on the Web. Also note that Web addresses are not permanent. The author of a page may graduate or change jobs and the site could be removed. The Internet is a useful resource for any serious researcher. One cannot only get a lot of useful information on the Web, but one can contact other researchers to exchange ideas, data, and programs.

BIBLIOGRAPHY

1. R. N. Bracewell, *The Fourier Transform and Its Applications,* 3rd ed., New York: McGraw-Hill, 1986.
2. E. O. Brigham, *The Fast Fourier Transform and Its Applications,* Engelwood Cliffs, NJ: Prentice-Hall, 1988.
3. M. T. Heideman, D. H. Johnson, and C. S. Burrus, Gauss and the history of the fast Fourier transform, *IEEE Acoust. Speech Signal Process. Mag.,* **1** (4): 14–21, 1984.
4. D. Gabor, Theory of communication, *J. IEE,* **93**: 429–457, 1946.
5. O. Rioul and M. Vetterli, Wavelets and signal processing, *IEEE Signal Process. Mag.,* **8** (4): 14–38, October 1991.
6. B. B. Hubbard, *The World According to Wavelets,* Wellesley, MA: A K Peters, 1996.
7. Y. Meyer, *Wavelets Algorithms & Applications,* Philadelphia: SIAM, 1993.
8. A. Haar, Zur theorie der orthogonalen funktionen-systeme, *Math. Ann.,* **69**: 331–371, 1910.
9. P. Franklin, A set of continuous orthogonal functions, *Math. Ann.,* **100**: 522–529, 1928.
10. E. Hernández and G. Weiss, *A First Course on Wavelets,* Boca Raton, FL: CRC Press, 1996.
11. R. E. Edwards and G. I. Gaudry, *Littlewood-Paley and Multiplier Theory,* Berlin: Springer-Verlag, 1977.
12. E M. Stein, *Harmonic Analysis: Real-Variable Methods, Orthogonality, and Oscillatory Integrals,* Princeton, NJ: Princeton University Press, 1993.
13. J.-O. Strömberg, A modified Franklin system and higher-order spline systems on $\mathbb{R}^n$ as unconditional bases for Hardy spaces, in W. Beckner et al. (eds.), *Conference on Harmonic Analysis in Honor of Antoni Zygmund,* Belmont CA: Wadsworth, 1983, vol. 2, pp. 475–494.
14. K. G. Wilson, Renormalization group and critical phenomens, II, Phase-space cell analysis of critical behavior, *Phys. Rev. B,* **4**: 3184–3205, 1971.

15. G. Battle, Wavelets: A renormalization group point of view, in M. B. Ruskai et al. (eds.), *Wavelets and Their Applications,* Boston: Jones and Bartlett, 1992, pp. 323–349.
16. G. Battle, Wavelet refinement of the Wilson recursion formula, in L. L. Schumaker and G. Webb (eds.), *Recent Advances in Wavelet Analysis,* San Diego: Academic Press, 1994, pp. 87–118.
17. R. J. Glauber, Coherent and incoherent states of the radiation field, *Phys. Rev.,* **131**: 2766–2788, 1963.
18. E. W. Aslaksen and J. R. Klauder, Unitary representations of the affine group, *J. Math. Phys.,* **9**: 206–211, 1968.
19. E. W. Aslaksen and J. R. Klauder, Continuous representation theory using the affine group, *J. Math. Phys.,* **10**: 2267–2275, 1969.
20. A. Crosier, D. Esteban, and C. Galand, Perfect channel splitting by use of interpolation/decimation/tree decomposition techniques, *Int. Conf. Inform. Sci. Syst.,* Patras, Greece, 1976, pp. 443–446.
21. D. Esteban and C. Galand, Application of quadrature mirror filters to split band voice coding schemes, *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.,* 1977, pp. 191–195.
22. R. E. Crochiere, S. A. Webber, and J. L. Flanagan, Digital coding of speech sub-bands, *Bell Syst. Tech. J.,* **55**: 1069–1085, 1976.
23. M. Vetterli and J. Kovačević, *Wavelets and Subband Coding,* Englewood Cliffs, NJ: Prentice-Hall, 1995.
24. P. J. Burt and E. H. Adelson, The Laplacian pyramid as a compact image code, *IEEE Trans. Commun.,* **31**: 532–540, 1983.
25. J. Morlet et al., Wave propagation and sampling theory—Part II: Sampling theory and complex waves, *Geophysics,* **47**: 222–236, 1982.
26. A. Grossmann and J. Morlet, Decomposition of Hardy functions into square integrable wavelets of constant shape, *SIAM J. Math. Anal.,* **15**: 723–736, 1984.
27. S. G. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Trans. Pattern Anal. Mach. Intell.,* **11**: 674–693, 1989.
28. S. G. Mallat, Multiresolution approximations and wavelet orthogonal bases of $L^2(\mathbb{R})$, *Trans. Amer. Math. Soc.,* **315**: 69–87, 1989.
29. I. Daubechies,, *Ten Lectures on Wavelets,* Philadelphia: SIAM, 1992.
30. I. Daubechies, Orthonormal bases of compactly supported wavelets, *Comm. Pure Appl. Math.,* **41**: 909–996, 1988.
31. I. Daubechies, The wavelet transform, time-frequency localization and signal analysis, *IEEE Trans. Inf. Theory,* **36**: 961–1005, 1990.
32. I. Daubechies, Where do wavelets come from?—A personal point of view, *Proc. IEEE,* **84**: 510–513, 1996.
33. C. E. Heil and D. F. Walnut, Continuous and discrete wavelet transforms, *SIAM Rev.,* **31**: 628–666, 1989.
34. M. Holschneider, *Wavelets an Analysis Tool,* Oxford: Clarendon Press, 1995.
35. B. K. Alpert, Wavelets and other bases for fast numerical linear algebra, in C. K. Chui (ed.), *Wavelets: A Tutorial in Theory and Applications,* San Diego: Academic Press, 1992, pp. 181–216.
36. I. Daubechies, Orthonormal bases of compactly supported wavelets II. Variations on a theme, *SIAM J. Math. Anal.,* **24**: 499–519, 1993.
37. G. Beylkin, R. Coifman, and V. Rokhlin, Fast wavelet transforms and numerical algorithms I, *Comm. Pure Appl. Math.,* **44**: 141–183, 1991.
38. A. Cohen and I. Daubechies, A stability criterion for biorthogonal wavelet bases of compactly supported wavelets, *Duke Math. J.,* **68**: 313–335, 1992.
39. A. Cohen, I. Daubechies, and J.-C. Feauveau, Biorthogonal bases of compactly supported wavelets, *Comm. Pure Appl. Math.,* **45**: 485–560, 1992.
40. M. Vetterli and C. Herley, Wavelets and filter banks: Theory and design, *IEEE Trans. Signal Process.,* **40**: 2207–2232, 1992.
41. G. Strang and T. Nguyen, *Wavelets and Filter Banks,* Wellesley, MA: Wellesley-Cambridge Press, 1996.
42. W. H. Press et al., *Numerical Recipes in C,* 2nd ed., Cambridge: Cambridge University Press, 1992.
43. A. Bruce and H.-Y. Gao, *Applied Wavelet Analysis with S-PLUS,* New York: Springer, 1996.
44. M. A. Cody, The fast wavelet transform, *Dr. Dobb's J.,* **17** (4): 16–28, 1992.
45. M. A. Cody, The wavelet packet transform, *Dr. Dobb's J.,* **19** (4): 44–54, 1994.
46. A N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition,* San Diego: Academic Press, 1992.
47. R. R. Coifman and M. V. Wickerhauser, Entropy-based algorithms for best basis selection, *IEEE Trans. Inf. Theory,* **38**: 713–718, 1992.
48. R. R. Coifman and M. V. Wickerhauser, Wavelets and adapted waveform analysis, in J. J. Benedetto and M. W. Frazier (eds.), *Wavelets: Mathematics and Applications,* Boca Raton, FL: CRC Press, 1994, pp. 399–423.
49. M. V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software,* Wellesley, MA: A. K. Peters, 1994.
50. S. Mallat and S. Zhong, Wavelet transform maxima and multiscale edges, in M. B. Ruskai et al. (eds.), *Wavelets and Their Applications,* Boston: Jones and Bartlett, 1992, pp. 67–104.
51. S. Mallat, Wavelets for a vision, *Proc. IEEE,* **84**: 604–614, 1996.
52. C.-C. Hsiao et al., Near optimal compression of orthonormal wavelet expansions, in J. J. Benedetto and M. W. Frazier (eds.), *Wavelets: Mathematics and Applications,* Boca Raton, FL: CRC Press, 1994, pp. 425–446.
53. R. A. DeVore, B. Jawerth, and B. J. Lucier, Image compression throught wavelet transform coding, *IEEE Trans. Inf. Theory,* **38**: 719–746, 1992.
54. M. V. Wickerhauser, Acoustic signal compression with wavelet packets, in C. K. Chui (ed.), *Wavelets: A Tutorial in Theory and Applications,* San Diego: Academic Press, 1992, pp. 679–700.
55. M. V. Wickerhauser, Comparison of picture compression methods: Wavelet, wavelet packet, and local cosine transform coding, in C. K. Chui, L. Montefusco, and L. Puccio, (eds.), *Wavelets: Theory, Algorithms, and Applications,* San Diego: Academic Press, 1994, pp. 585–621.
56. D. L. Donoho, Unconditional bases are optimal bases for data compression and for statistical estimation, *Appl. Computat. Harmonic Anal.,* **1**: 100–115, 1993.
57. M. Farge, Wavelet transforms and their applications to turbulence, *Annu. Rev. Fluid Mech.,* **24**: 395–457, 1992.
58. M. Farge et al., Wavelets and turbulence, *Proc. IEEE,* **84**: 639–669, 1996.
59. M. E. Mayer, L. Hudgins, and C. A. Friehe, Wavelet spectra of buoyant atmospheric turbulence, in C. K. Chui, L. Montefusco, and L. Puccio, (eds.), *Wavelets: Theory, Algorithms, and Applications,* San Diego: Academic Press, 1994, pp. 533–541.
60. A. Druilhet et al., Experimental study of inhomogeneous turbulence in the lower troposphere by wavelet analysis, in C. K. Chui, L. Montefusco, and L. Puccio, (eds.), *Wavelets: Theory, Algorithms, and Applications,* San Diego: Academic Press, 1994, pp. 543–559.
61. M. V. Wickerhauser et al., Efficiency comparison of wavelet packet and adapted local cosine bases for compression of a two-dimensional turbulent flow, in C. K. Chui, L. Montefusco, and L.

Puccio, (eds.), *Wavelets: Theory, Algorithms, and Applications,* San Diego: Academic Press, 1994, pp. 509–531.

62. G. W. Wornell, *Signal Processing with Fractals: A Wavelet-Based Approach,* Upper Saddle River, NJ: Prentice Hall, 1966.
63. M. Hazewinkel, Wavelets understand fractals, in T. H. Koornwinder (ed.), *Wavelets: An Elementary Treatment of Theory and Applications,* River Edge, NJ: World Scientific, 1995, pp. 207–219.
64. M. Unser and A. Aldroubi, A review of wavelets in biomedical applications, *Proc. IEEE,* **84**: 626–638, 1996.
65. J. B. Weaver et al., Filtering noise from images with wavelet transforms, *Magn. Reson. Med.,* **24**: 288–295, 1991.
66. D. L. Donoho, De-noising by soft-thresholding, *IEEE Trans. Inf. Theory,* **41**: 613–627, 1995.
67. D. L. Donoho and I. M. Johnstone, Ideal spatial adaptation via wavelet shrinkage, *Biometrika,* **81**: 425–455, 1994.
68. R. A. DeVore and B. J. Lucier, Fast wavelet techniques for near-optimal image processing, *Proc. IEEE Military Commun. Conf.,* New York: IEEE, 1992, pp. 48.3.1–48.3.7.
69. J. Fan and A. Laine, Multiscale contrast enhancement and denoising in digital radiographs, in A. Aldroubi and M. Unser (eds.), *Wavelets in Medicine and Biology,* Boca Raton, FL: CRC Press, 1996, pp. 163–189.
70. W. Qian et al., Computer assisted diagnosis for digital mammography, *IEEE Eng. Med. Biol. Mag.,* **14**: 561–569, 1995.
71. W. Qian et al., Tree structured wavelet transform segmentation of microcalcifications in digital mammography, *Med. Phys.,* **22**: 1247–1254, 1995.
72. W. Qian and L. P. Clarke, Wavelet-based neural network with fuzzy-logic adaptivity for nuclear image restoration, *Proc. IEEE,* **84**: 1458–1473, 1996.
73. R. A. DeVore, B. Lucier, and Z. Yang, Feature extraction in digital mammography, in A. Aldroubi and M. Unser (eds.), *Wavelets in Medicine and Biology,* Boca Raton, FL: CRC Press, 1996, pp. 145–161.
74. J. J. Heine et al., Multiresolution statistical analysis of high-resolution digital mammograms, *IEEE Trans. Med. Imaging,* **16**: 503–515, 1997.
75. S. R. Deans, *The Radon Transform and Some of Its Applications,* New York: Wiley, 1983. Malabar, FL: Krieger, 1993.
76. F. Rashid-Farrokhi et al., Wavelet-based multiresolution local tomography, *IEEE Trans. Image Process.,* **6**: 1412–1430, 1997.
77. D. M. Healy, Jr. and J. B. Weaver, Adapted wavelet techniques for encoding magnetic resonance images, in A. Aldroubi and M. Unser (eds.), *Wavelets in Medicine and Biology,* Boca Raton, FL: CRC Press, 1996, pp. 297–352.

Reading List

In addition to Refs. 10, 23, 29, 34, 41, 46, 49, and 62 other possible texts are listed.

A. C. Cohen and R. D. Ryan, *Wavelets and Multiscale Signal Processing,* London: Chapman & Hall, 1995.

C. K. Chui, *An Introduction to Wavelets,* San Diego: Academic Press, 1992.

G. Kaiser, *A Friendly Guide to Wavelets,* Boston: Birkhäuser, 1994.

Y. Meyer, *Wavelets and Operators,* Cambridge: Cambridge University Press, 1992.

R. T. Ogden, *Essential Wavelets for Statistical Applications and Data Analysis,* Boston: Birkhäuser, 1997.

L. Prasad and S. S. Iyengar, *Wavelet Analysis with Applications to Image Processing,* Boca Raton, FL: CRC Press, 1997.

B. W. Suter, *Multirate and Wavelet Signal Processing,* Boca Raton, FL: Academic Press, 1998.

G. W. Walter, *Wavelets and Other Orthogonal Systems with Applications,* Boca Raton, FL: CRC Press, 1994.

P. Wojtaszczyk, *A Mathematical Introduction to Wavelets,* Cambridge: Cambridge University Press, 1997.

This list contains useful sources for applications. Most of these were not cited in the bibliography, but in a few cases there is overlap.

M. Akay (ed.), *Time Frequency and Wavelets in Biomedical Signal Processing,* New York: IEEE Press, 1998.

This is excellent for engineers and applied scientists. It covers time–frequency analysis methods with biomedical applications; wavelets, wavelet packets, and matching pursuits with biomedical applications; wavelets and medical imaging; and wavelets, neural networks, and fractals.

A. Aldroubi and M. Unser (eds.), *Wavelets in Medicine and Biology,* Boca Raton, FL: CRC Press, 1996.

This covers many applications in medicine and biology. The main topics are wavelet transform: theory and implementation, wavelets in medical imaging and tomography, wavelets and biomedical signal processing, wavelets and mathematical models in biology.

J. J. Benedetto and M. W. Frazier (eds.), *Wavelets: Mathematics and Applications,* Boca Raton, FL: CRC Press, 1994.

This is good for both foundations and applications. It contains core material, wavelets and signal processing, and wavelets and partial differential operators.

E. Foufoula-Georgiou and P. Kumar (eds.), *Wavelets in Geophysics,* San Diego: Academic Press, 1994.

A brief summary of wavelets is followed with applications directed toward turbulence and geophysics.

C. K. Chui (ed.), *Wavelets: A Tutorial in Theory and Applications,* San Diego: Academic Press, 1992.

This reference contains many articles on foundations for applications. There are sections on orthogonal wavelets, semi-orthogonal and nonorthogonal wavelets, wavelet-like local bases, multivariate scaling functions and wavelets, short-time Fourier and window-Radon transforms, theory of sampling and interpolation, and applications to numerical analysis and signal processing.

L. L. Schumaker and G. Webb (eds.), *Recent Advances in Wavelet Analysis,* San Diego: Academic Press, 1994.

The articles here are mainly on recent advances related to mathematical properties of wavelets.

C. K. Chui, L. Montefusco, and L. Puccio (eds.), *Wavelets: Theory, Algorithms, and Applications,* San Diego: Academic Press, 1994.

Several fundamentals are covered. These include multiresolution and multilevel analysis, wavelet transforms, spline wavelets, other mathematical tools for time–frequency analysis, wavelets and fractals, numerical methods and algorithms, and applications.

W. Dahmen, A. Kurdila, and P. Oswald (eds.), *Multiscale Wavelet Methods for Partial Differential Equations,* San Diego: Academic Press, 1997.

This will be of interest to people working with processes involving differential and integral equations, fast algorithms, software tools, numerical experiments, turbulence, and wavelet analysis of partial differential operators.

M. Farge, J. C. R. Hunt, and J. C. Vassilicos (eds.), *Wavelets, Fractals, and Fourier Transforms,* Oxford: Clarendon Press, 1993.

The papers here are based on the proceedings of a conference held at Newnham College, Cambridge in December 1990.

M. B. Ruskai et al. (eds.), *Wavelets and Their Applications,* Boston: Jones and Bartlett, 1992.

There are some very useful articles in this reference. It includes signal analysis, numerical analysis, wavelets and quantum mechanics, and theoretical developments.

J. M. Combes, A. Grossmann, and Ph. Tchmitchian (eds.), *Wavelets: Time–Frequency Methods and Phase Space,* 2nd ed., Berlin: Springer-Verlag, 1990.

The conference proceedings of a conference held at Marseille, France in 1988 are contained here. This brought together an interdisciplinary mix of participants, including many major contributors to the development of wavelet methods.

Y. Meyer (ed.), *Wavelets and Applications,* Berlin: Springer-Verlag, 1992.

The proceedings of an international conference on wavelets held at Marseille are in this volume. This conference along with the previous one illustrates and captures some of the flavor and excitement of time.

A. Antoniadis and G. Oppenheim (eds.), *Wavelets and Statistics,* Lecture Notes in Statistics 103, New York: Springer-Verlag, 1995.

This contains the proceedings of a conference on wavelets and statistics held at Villard de Lans, France in 1994.

T. H. Koornwinder (ed.), *Wavelets: An Elementary Treatment of Theory and Applications,* River Edge, NJ: World Scientific, 1993.

This series of articles provides a good introduction to wavelets. It is available in paperback and could serve as a text for a one-semester course.

Finally, there are a few important issues of journals that have been devoted entirely to wavelets, and applications. These include: *IEEE Trans. Inf. Theory,* **38**: March 1992, Part II of two parts; *IEEE Trans. Signal Process.,* **41**: December 1993; *Proc. IEEE,* **84**: April 1996; *Ann. Biomed. Eng.* **23** (5), 1995. A fairly new journal is *Applied and Computational Harmonic Analysis,* started in 1993; many articles in this journal are devoted to wavelets and applications.

STANLEY R. DEANS
JOHN J. HEINE
DEEPAK GANGADHARAN
WEI QIAN
MARIA KALLERGI
LAURENCE P. CLARKE
University of South Florida

WAVE SCATTERING, ELECTROMAGNETIC. See ELECTROMAGNETIC SUBSURFACE REMOTE SENSING.

WEATHER. See METEOROLOGICAL RADAR.

WEBCASTING. See BROADCASTING VIA INTERNET.

WEB COMPUTING. See NETWORK COMPUTING.

WEBER FUNCTIONS. See BESSEL FUNCTIONS.

WEB PROGRAMMING. See JAVA, JAVASCRIPT, HOT JAVA.

WEB SERVICES. See INTERNET COMPANIES.

/ /

z transforms

Standard Article

Dennis M. Sullivan[1]

[1]University of Idaho, Moscow, ID

Copyright © 1999 by John Wiley & Sons, Inc. All rights reserved.

: 10.1002/047134608X.W2467

Article Online Posting Date: December 27, 1999

Abstract | Full Text: HTML PDF (201K)

Abstract

The sections in this article are

Definition of the Z Transform

Convolution Using the Z Transform

Convolution of Sampled Signals

Properties of Z Transforms

The Inverse Z Transform

Stability

Alternative Methods to Formulate the Z Transform

An Example from Electromagnetic Simulation

Summary

| | |

Copyright © 1999-2008 All Rights Reserved.

Z TRANSFORMS

One of the most useful techniques in engineering analysis is transforming a problem from the time domain to the frequency domain. Using a Fourier transform, differential equations are changed to difference equations, often substantially simplifying the analysis. When dealing with linear systems, the relationship between the input and output is a convolution integral. However, this reduces to simple multiplication in the frequency domain. When dealing with transient signals, it is often more convenient to use the Laplace transform, but the principle is the same.

The Laplace and Fourier transforms are appropriate for analog signals. When dealing with digital signals, the Z transform is used. The reasons for using it are analogous: (1) complicated difference equations in the time domain become algebraic equations in the Z domain, and (2) the relationship between the input and output of a linear system is a multiplication in the Z domain instead of a convolution in the sampled time domain.

DEFINITION OF THE Z TRANSFORM

The Z transform is extremely useful when dealing with functions in the sampled time domain; that is, instead of the function $x(t)$, we have a function of the type

$$x(n) = \sum_{n=-\infty}^{\infty} x(t)\delta(t - nT)$$

where T is a uniform time interval. The Z transform is defined by

$$Z[x(n)] = X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{1}$$

A power of z is associated with each delay interval T. Suppose we have a function $x(n)$, as given in Fig. 1(a)

$$x(n) = \delta(t) + 0.5\delta(t - T) + 0.25\delta(t - 2T) \tag{2}$$

In the Z domain, this is written as

$$X(z) = 1 + 0.5z^{-1} + 0.25z^{-2} \tag{3}$$

which is shown in Fig. 1(b). Note that the first term is merely 1, because $z^{-0} = 1$.

In its simplest form, the z^{-1} operator associated with the Z transform can be thought of as a delay operator. So if

$$y(n) = x(n - 1)$$

then the Z transform of this equation would be written as

$$Y(z) = z^{-1}X(z)$$

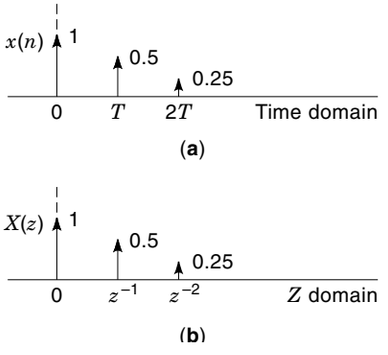


Figure 1. Graph of Eq. (2) in (a) the sampled time domain and (b) the Z domain. Notice that the Z domain is exactly the same as the sampled time domain, except that delays by the time interval T are indicated by the z^{-1} operator.

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

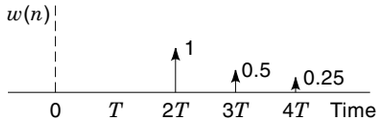


Figure 2. Graph of Eq. (4) after it has been put in the time domain. Since $W(z) = z^{-2}X(z)$, the graph of $w(n)$ is the same as $x(n)$ in Fig. 1, except it has been delayed by $2T$, as indicated by the z^{-2}.

and if

$$w(n) = y(n-1)$$

the Z transform of $w(n)$ is

$$W(z) = z^{-1}Y(z) = z^{-1}\left(z^{-1}X(z)\right) = z^{-2}X(z) \tag{4}$$

If $W(z)$ is defined as in Eq. (3), and $X(z)$ as defined in Eq. (2) then

$$W(z) = z^{-2}1 + 0.5z^{-3} + 0.25z^{-4}$$

which in turn, going back to the sampled time domain (Fig. 2), means that

$$w(n) = \delta(t-2T) + 0.5 \cdot \delta(t-3T) + 0.25 \cdot \delta(t-4T)$$

We can add two Z transforms, merely by making sure like powers of z are lumped together. For instance, adding $X(z)$ of Eq. (2) and $W(z)$ of Eq. (3) gives

$$\begin{aligned} X(z) + W(z) &= [1 + 0.5z^{-1} + 0.25z^{-2}] + [1z^{-2} + 0.5z^{-3} + 0.25z^{-4}] \\ &= 1 + 0.5z^{-1} + 1.25z^{-2} + 0.5z^{-3} + 0.25z^{-4} \end{aligned}$$

Going back to the sampled time domain,

$$\begin{aligned} x(n) + w(n) = {} & \delta(t) + 0.5 \cdot \delta(t-T) + 1.25 \cdot \delta(t-2T) \\ & + 0.5 \cdot \delta(t-3T) + 0.25 \cdot \delta(t-4T) \end{aligned}$$

which could have been obtained by adding $x(n)$ and $w(n)$, being sure to keep like delta (δ) terms together.

CONVOLUTION USING THE Z TRANSFORM

Figure 3 illustrates a simple linear system. In this example, suppose $h(n)$ is a system that adds the present value of $x(n)$ to its previous value $x(n-1)$ and outputs it as the new value of $y(n)$:

$$y(n) = x(n) + x(n-1) \tag{5}$$

This function $h(n)$ is referred to as the "impulse response" for the following reason: if the input $x(n)$ is an impulse $\delta(n)$ then

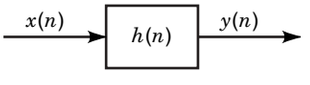


Figure 3. A simple discrete linear system.

the output is

$$\begin{aligned} y(n) &= 1 \qquad \text{for } n = 0 \text{ or } 1 \\ y(n) &= 0 \qquad \text{otherwise} \end{aligned}$$

See Ref. 1 or 2. Therefore,

$$h(n) = \delta(n) + \delta(n-1) \tag{6}$$

A more mathematically concise expression is

$$y(n) = \sum_{i=0}^{1} x(n-i) \cdot h(i) \tag{7}$$

Equation 7 is a discrete convolution. Notice that the index i only ranges between 0 and 1 because in Eq. (6) $h(n)$ has only these two terms. This could be generalized for any upper limit to infinity.

If, instead of an impulse, we use the values of $x(n)$ from Eq. (2), $y(n)$ is calculated from Eq. (7):

$$y(n) = \delta(t) + 1.5 \cdot \delta(t-T) + 0.75 \cdot \delta(t-2T) + 0.25 \cdot \delta(t-3T) \tag{8}$$

This process was made tractable only by the small number of terms used in this example.

As an alternative approach, take the Z transforms of $x(n)$ and $h(n)$:

$$\begin{aligned} X(z) &= 1 + 0.5 \cdot z^{-1} + 0.25 \cdot z^{-2} \\ H(z) &= 1 + z^{-1} \end{aligned}$$

Multiplying the two together gives

$$\begin{aligned} H(z) \cdot X(z) &= (1 + 0.5 \cdot z^{-1} + 0.25 \cdot z^{-2}) \cdot (1 + z^{-1}) \\ &= 1 + 1.5 \cdot z^{-1} + 0.75 \cdot z^{-2} + 0.25 \cdot z^{-2} \end{aligned}$$

Going back to the time domain gives the $y(n)$ of Eq. (8). This illustrates the powerful "convolution theorem": Convolution in the discrete time domain becomes multiplication in the Z domain. The $H(z)$, which is the Z transform of the impulse response, is referred to as the transfer function. The proof follows.

Starting with the definition of convolution in the discrete time domain,

$$y(n) = \sum_{i=0}^{\infty} h(n-i) \cdot x(i) \tag{9}$$

take the Z transform of both sides

$$\sum_{n=0}^{\infty} y(n)z^{-n} = \sum_{n=0}^{\infty} \sum_{i=0}^{\infty} h(n-i) \cdot x(i)z^{-n}$$

and then interchange the summation signs

$$Y(z) = \sum_{i=0}^{\infty} x(i) \sum_{n=0}^{\infty} h(n-i) \cdot z^{-n}$$

Finally, multiplying by $z^{-i} \cdot z^{i}$ gives

$$Y(z) = \sum_{i=0}^{\infty} x(i)z^{-1} \sum_{n=0}^{\infty} h(n-i) \cdot z^{-n+i}$$

and using the parameter $m = n - i$

$$Y(z) = \sum_{i=0}^{\infty} x(i) \cdot z^{-i} \sum_{m=0}^{\infty} h(m) \cdot z^{-m}$$

gives

$$Y(z) = H(z) \cdot Y(z) \tag{10}$$

Note that Eq. (1) is usually referred to as the *bilateral Z Transform* because it is defined for both positive and negative n. However, we will almost always use causal functions, so the summation will be over the positive n's.

Example

As an example, suppose the impulse response in Fig. 3 is an exponentially decaying function

$$h(n) = Ae^{-nT/t_0} \qquad n = 0, 1, 2, 3, \ldots \tag{11}$$

and the input is the discretized unit step function

$$u(n) = 1 \qquad n = 0, 1, 2, 3, \ldots \tag{12}$$

Since

$$\sum_{n=0}^{\infty} a^{-n} = \frac{1}{1 - a^{-1}} \quad \text{when} \quad a \leq 1$$

$H(z)$ can be calculated

$$Z[h(n)] = H(z) = A \sum_{n=0}^{\infty} [e^{T/t_0} z]^{-n} = \frac{A}{1 - e^{-T/t_0} z^{-1}} \tag{13}$$

and similarly

$$Z[u(n)] = U(z) = \frac{1}{1 - z^{-1}}$$

Or we could simply refer to a table of Z transforms, such as Table 1. The desired convolution is

$$\begin{aligned} Y(z) = H(z) \cdot U(z) &= \frac{A}{1 - e^{-T/t_0} z^{-1}} \cdot \frac{1}{1 - z^{-1}} \\ &= \frac{A}{1 - (1 + e^{-T/t_0}) z^{-1} + e^{-T/t_0} z^{-2}} \end{aligned} \tag{14}$$

To get a solution in the time domain, we take the partial fraction expansion of $Y(z)$

$$Y(z) = \frac{A}{1 - e^{-T/t_0} z^{-1}} \cdot \frac{1}{1 - z^{-1}} = A \cdot \left[\frac{B}{1 - e^{-T/t_0} z^{-1}} + \frac{C}{1 - z^{-1}} \right] \tag{15}$$

It turns out that

$$B = -\frac{e^{-T/t_0}}{1 - e^{-T/t_0}} \quad \text{and} \quad C = \frac{1}{1 - e^{-T/t_0}}$$

(The partial fraction expansion technique will be explained later.) Now the two terms in Eq. (15) can be taken back into the sampled time domain by finding the time domain terms corresponding to the two Z domain terms in Table 1, giving

$$y(n) = \frac{A}{1 - e^{-T/t_0}} [1 - e^{-(n+1)T/t_0}] \quad n = 0, 1, 2, 3 \ldots \tag{16}$$

Equation (16) is an analytic solution. An alternative approach exists. Consider Eq. (14) as a purely algebraic problem where we are solving for $Y(z)$:

$$\begin{aligned} [1 - (1 + e^{-T/t_0}) z^{-1} - e^{-T/t_0} z^{-2}] \cdot Y(z) &= A \\ Y(z) = (1 + e^{-T/t_0}) z^{-1} Y(z) - e^{-T/t_0} z^{-2} Y(z) + A \end{aligned} \tag{17}$$

Then remembering that the z^{-1} is an operator that just means a delay of one, we can go back to the sampled time domain

$$y(n) = (1 + e^{-T/t_0}) y(n-1) - e^{-T/t_0} y(n-2) + A \cdot \delta(n) \tag{18}$$

Note the following: The A term of Eq. (17) became $A \cdot \delta(n)$ because a constant in the Z domain is a delta function in the

Table 1. Transforms Among the Time, Frequency, Sampled-Time, and Z Domains

Time Domain	Frequency Domain	Sampled Time Domain	Z Domain
$\delta(t)$	1	$\delta(n)$	1
$u(t)$	$\frac{1}{j\omega}$	$u(n)$	$\frac{1}{1 - z^{-1}}$
$tu(t)$	$\frac{1}{(j\omega)^2}$	$n \cdot u(n)$	$\frac{z^{-1}}{(1 - z^{-1})^2}$
$e^{-\alpha t} \cdot u(t)$	$\frac{1}{\alpha + j\omega}$	$e^{-\alpha nT} \cdot u(n)$	$\frac{1}{1 - z^{-1} e^{-\alpha \cdot T}}$
$e^{-\alpha t} \sin(\beta t) \cdot u(t)$	$\frac{\beta}{(\alpha^2 + \beta^2) + j2\alpha\omega - \omega^2}$	$e^{-\alpha nT} \sin(\beta nT) \cdot u(n)$	$\frac{e^{-\alpha \cdot T} \cdot \sin(\beta T) \cdot z^{-1}}{1 - 2e^{-\alpha \cdot T} \cdot \cos(\beta T) \cdot z^{-1} + e^{-2\alpha \cdot T} \cdot z^{-2}}$
$e^{-\alpha t} \cos(\beta t) \cdot u(t)$	$\frac{\alpha + j\omega}{(\alpha^2 + \beta^2) + j2\alpha\omega - \omega^2}$	$e^{-\alpha nT} \cos(\beta nT) \cdot u(n)$	$\frac{1 - e^{-\alpha \cdot T} \cdot \cos(\beta T) \cdot z^{-1}}{1 - 2e^{-\alpha \cdot T} \cdot \cos(\beta T) \cdot z^{-1} + e^{-2\alpha \cdot T} \cdot z^{-2}}$

```
DELTA = 1
DO N=0,NMAX
  Y(N) = (1 + EXP(−T/T0) ) * Y(N−1) − EXP(−T/T0)*Y(N−2) + A*DELTA
  DELTA = 0
END DO
```

(a)

```
X = 1
DO N=0,NMAX
    Y(N) = EXP(−T/T0) * Y(N−1) + A* X
END DO
```

(b)

Figure 4. Computer codes to convolve an exponentially decaying function and the unit step function: (a) implementation of Eq. (18); (b) implementation of Eq. (19). The two codes are apparently of different form, but give identical results.

time domain (Table 1). To convince yourself of this, substitute $\delta(n)$ for $x(n)$ in Eq. (1), the definition of the Z transform. Only the $n = 0$ term survives, that is, a constant. It is not obvious, but Eq. (18) is equivalent to Eq. (16). Equation (16) is an analytic solution, while Eq. (18) is more appropriate when calculating the solution iteratively. A computer code to calculate Eq. (17) is given in Fig. 4(a).

There is yet another alternative approach. In the previous example, we specified the input $x(n)$ in Fig. 1, as $u(n)$. This is the step function, sometimes referred to as the Heaviside function. Suppose $x(n)$ is left as an unspecified function. Then

$$Y(z) = H(z)X(z) = \frac{A}{1 - e^{-T/t_0}z^{-1}} \cdot X(z)$$

and following the same process

$$\begin{aligned} Y(z)(1 - e^{-T/t_0}z^{-1}) &= A \cdot X(z) \\ Y(z) &= e^{-T/t_0}z^{-1}Y(z) + A \cdot X(z) \\ y(n) &= e^{-T/t_0}y(n-1) + A \cdot x(n) \end{aligned} \tag{19}$$

Now the appropriate computer code is given by Fig. 4(b). The result is identical to that generated in Fig. 4(a); however, this is the more general form. The function x is just specified as 1, and assuming that $N = 0$ corresponds to $t = 0$, X is the step function. However, we could replace X with any function and it would be convolved with the exponential. In fact, this is a simple one pole digital filter.

CONVOLUTION OF SAMPLED SIGNALS

In dealing with discrete functions, there are actually two types of problems: (1) the discrete functions are sequences of numbers, or (2) the discrete functions are sampled versions of continuous functions. The key point separating the two is whether or not the time interval between samples is an issue. In the previous section, we treated the first problem; now we will look at the second.

In Fig. 3, suppose we started with continuous functions $x(t)$, $h(t)$, and $y(t)$ instead of $x(n)$, $h(n)$, and $y(n)$, respectively. The convolution in the time domain is

$$y(t) = \int_0^\infty h(\tau) \cdot x(t - \tau)\, d\tau \tag{20}$$

where it has once again been assumed that the system response $h(t)$ is causal. Suppose that in order to simulate it on a computer, this problem had to be implemented in the discrete domain. The finite difference approximation of the integral in Eq. (20) is

$$\begin{aligned} y(n) &\cong \sum_{i=0}^{n} h(n-i) \cdot x(i) \cdot T \\ &= T \sum_{i=-\infty}^{n} h(n-i) \cdot x(i) \end{aligned} \tag{21}$$

where T is the time interval between samples. Taking the Z transform of both sides

$$\sum_{n=0}^{\infty} y(n)z^{-n} = T \sum_{n=0}^{\infty} \sum_{i-0}^{n} h(n-i) \cdot x(i)z^{-n}$$

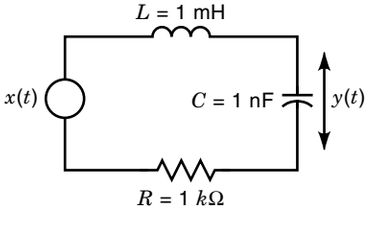
the development becomes identical to the previous section, except we obtain the extra T:

$$Y(z) = T \cdot X(z) \cdot H(z) \tag{22}$$

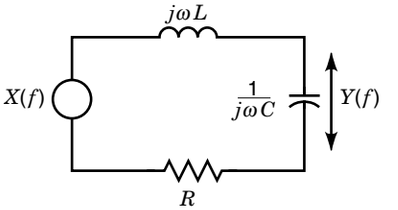
Simulation of a Two Pole Digital Filter

In this section, we will design a digital filter equivalent to the *RLC* circuit in Fig. 5(a). It will be convenient to start in the frequency domain [Fig. 5(b)], from which we obtain the following transfer function:

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{1/j\omega C}{j\omega L + 1/j\omega C + R} = \frac{1/LC}{1/LC + j\omega R/L - \omega^2} \tag{23}$$



(a)



(b)

Figure 5. (a) An *RLC* circuit; (b) Fourier components of the *RLC* circuit.

To get this into a recognizable form, we will use the following change of parameters:

$$\alpha = \frac{R}{2L} = 0.5 \times 10^6 \qquad \beta = \sqrt{\frac{1}{LC} - \alpha^2} = 0.866 \times 10^6$$
$$\gamma\beta = \frac{1}{LC} \Rightarrow \gamma = 1.155 \times 10^6$$

The Z transform can now be read from the frequency domain expression in Table 1

$$\begin{aligned} H(z) &= Z\left[\gamma \frac{\beta}{(\alpha^2+\beta^2)+j2\alpha\omega-\omega^2}\right] \\ &= 1.155 \times 10^6 \frac{e^{-\alpha T}\sin(\beta \cdot T)\cdot z^{-1}}{1-2e^{-\alpha T}\cdot\cos(\beta\cdot T)\cdot z^{-1}+e^{-2\alpha T}} \end{aligned} \tag{24}$$

At first, we may be somewhat startled to see the magnitude of the multiplier resulting from the γ term. But remember, when it is convolved with another function, it will be multiplied by T! Since $\beta = 0.866 \times 10^6$, we will want T to be much smaller, so choose $T = 10^{-7}$. Notice now that

$$e^{-\alpha\cdot T} = e^{-0.05} = 0.951$$
$$e^{-2\alpha\cdot T} = e^{-.1} = 0.904$$

and

$$\sin(\beta\cdot T) = \sin(0.086) = 0.086$$
$$\cos(\beta\cdot T) = \cos(0.086) = 0.9963$$

Now take the convolution of $h(n)$ with an unknown function $x(n)$.

$$\begin{aligned} Y(z) &= H(z)X(z)T \\ &= \frac{1.155\times10^6\cdot(0.951)\cdot(0.086)\cdot10^{-7}}{1-2\cdot(0.951)\cdot(0.9963)\cdot z^{-1}+(0.904)z^{-2}} z^{-1}X(z) \\ &= \frac{0.0094}{1-1.895z^{-1}+0.9044z^{-2}} z^{-1}X(z) \end{aligned} \tag{25}$$

$$\begin{aligned} Y(z) &= 1.895\cdot z^{-1}Y(z) - 0.9044z^{-2}Y(z) + 0.0094\cdot z^{-1}X(z) \\ y(n) &= 1.895\cdot y(n-1) - 0.9044\cdot y(n-2) + 0.0094\cdot x(n-1) \end{aligned} \tag{26}$$

Note that the z^{-1} in the numerator of $H(z)$ in Eq. (25) resulted in the $x(n-1)$, that is, a delay in the input in Eq. (26). The computer code in Fig. 6 implements Eq. (26).

Alternatively, we may be asked to build the digital equivalent of the analog filter shown in Fig. 5. This is done in Fig. 7. Note the use of delay registers (marked D), which hold the values for one clock cycle, changing $y(n)$ to $y(n-1)$ and $y(n-1)$ to $y(n-2)$. These, along with the input, are multiplied by their respective scaling factors and summed to give the new value of $y(n)$ at every clock cycle, as per Eq. (26).

```
T = 1E−7
GAMMA = 1.155e6
DO N=0,NMAX
    Y(N) = 2*EXP(−ALPHA*T)*COS(BETA*T)* Y(N−1)
    − EXP(−2*ALPHA*T) * Y(N−2)+ T*GAMMA*SIN(BETA*T)* X(N−1)
END DO
```

Figure 6. Simulation of the *RLC* circuit in Fig. 5. *X*(*N*) is the (as yet unspecified) input.

Sum of Two Parallel Systems

A system diagram is given in Fig. 8. The two transfer functions are

$$H_1(\omega) = \frac{1}{j\omega+\alpha_1} \tag{27}$$

and

$$H_2(\omega) = \frac{1}{j\omega+\alpha_2} \tag{28}$$

Suppose we want to design a digital simulation of this system. The overall transfer function of the system is given by

$$Y(\omega) = [H_1(\omega) + H_2(\omega)]\cdot X(\omega) \tag{29}$$

Going to the Z domain gives

$$\begin{aligned} Y(z) &= [H_1(z)+H_2(z)]\cdot X(z)\cdot T \\ &= \left[\frac{1}{1-e^{-\alpha_1 T}z^{-1}} + \frac{1}{1-e^{-\alpha_2 T}z^{-1}}\right]\cdot T\cdot X(z) \\ &= \left[\frac{2-(e^{-\alpha_1 T}+e^{-\alpha_2 T})z^{-1}}{1-(e^{-\alpha_1 T}+e^{-\alpha_2 T})z^{-1}+z^{-2}}\right]\cdot T\cdot X(z) \end{aligned} \tag{30}$$

from which we get

$$\begin{aligned} Y(z) &= (e^{-\alpha_1 T}+e^{-\alpha_2 T})z^{-1}Y(z) - z^{-2}Y(z) \\ &\quad + 2\cdot T\cdot X(z) - T\cdot(e^{-\alpha_1 T}+e^{-\alpha_2 T})z^{-1}X(z) \end{aligned} \tag{31}$$

Note that two terms of the input are used with $X(z)$ corresponding to $x(n)$, and $z^{-1}X(z)$ corresponding to $x(n-1)$. This does not present any particular difficulty.

Going back to Eq. (30), instead of cross multiplying, suppose we define two auxiliary functions

$$S_1(z) = \frac{T}{1-e^{-\alpha_1 T}z^{-1}}\cdot X(z)$$
$$S_2(z) = \frac{T}{1-e^{-\alpha_2 T}z^{-1}}\cdot X(z)$$

So instead of the results of Eq. (31), we get

$$S_1(z) = e^{-\alpha_1 T}z^{-1}S_1(z) + T\cdot X(z) \tag{32a}$$
$$S_2(z) = e^{-\alpha_2 T}z^{-1}S_2(z) + T\cdot X(z) \tag{32b}$$
$$Y(z) = S_1(z) + S_2(z) \tag{32c}$$

The results of Eqs. (32) present a simpler formulation. This is a method of defining auxiliary parameters so that several small calculations are being made instead of one large one, often an easier process. If, for instance, H_1 and H_2 were each second-order systems, the cross multiplication similar to Eq. (31) would produce a fourth order system. It would be far better to define two-second order auxiliary parameters and make

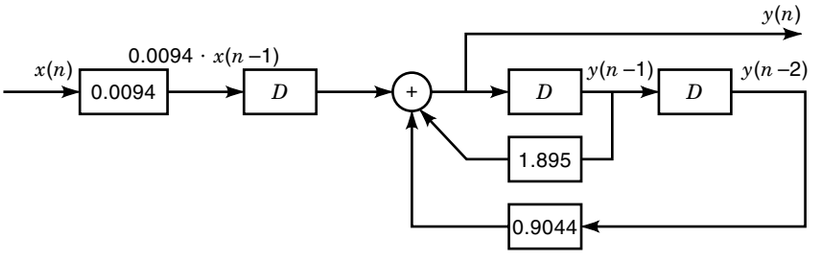


Figure 7. Digital hardware implementation of the analog filter in Fig. 5. The boxes marked D represent delay registers.

two second-order calculations, similar to Eq. (32a) and Eq. (32b). Figure 9 is the digital simulation of the transfer function.

PROPERTIES OF *Z* TRANSFORMS

The definition of the Z transform has been given along with some examples of how the Z transform may be used. In solving these examples, we used the important convolution theorem. Here are some other important properties of Z transforms.

Linearity

If $F_1(z) = Z[f_1(n)]$ and $F_2(z) = Z[f_2(n)]$, then $Z[\alpha f_1(n) + \beta f_2(n)] = \alpha F_1(z) + \beta F_2(z)$

Proof

$$Z[\alpha \cdot f_1(n) + \beta \cdot f_2(n)] = \sum_{n=-\infty}^{\infty} (\alpha \cdot f_1(n) + \beta \cdot f_2(n)) z^{-n}$$

$$= \alpha \sum_{n=-\infty}^{\infty} f_1(n) + \beta \sum_{n=-\infty}^{\infty} f_2(n) = \alpha \cdot F_1(z) + \beta \cdot F_2(z)$$

Time-Shifting

If $F(z) = Z[f(n)]$, then $Z[f(n - m)] = z^{-m}F(z)$

Proof

$$Z[f(n-m)] = \sum_{n=-\infty}^{\infty} f(n-m) z^{-n}$$

$$= z^{-m} \sum_{n=\infty}^{\infty} f(n-m) z^{-(n-m)}$$

$$= z^{-m} \sum_{l=-\infty}^{\infty} f(l) z^{-(l)} = z^{-m} F(z) \quad l = n - m$$

Note that this proof was for the general case of a noncausal function $f(n)$. If $f(n)$ is causal, or of a finite duration, then additional terms may appear (3).

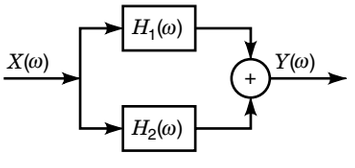


Figure 8. A system made up of two parallel sub systems.

Initial Value Theorem

If $F(z) = Z[f(n)]$, then $f(0) = \lim_{z \Rightarrow \infty} F(z)$

Proof The proof comes directly from the definition of the Z transform in Eq. (1). As $z \Rightarrow \infty$, all terms vanish except $f(0)$, which proves the theorem.

Final Value Theorem

If $F(z) = Z[f(n)]$, and $(z - 1)F(z)$ has no poles on or outside the unit circle (*see the section on stability*), then

$$f(\infty) = \lim_{z \to 1} (z-1)F(z)$$

This theorem can be extremely helpful, because it gives us the steady state value without solving for the entire sequence in the time domain. However, the proof is nontrivial. The interested reader should see Ref. 3.

There are numerous other theorems, some of which are listed in Table 2. More extensive lists are available in the literature (1–4).

THE INVERSE *Z* TRANSFORM

Like all frequency domain transforms, there is an inverse Z transform given by

$$f(n) = \frac{1}{2\pi j} \int_{\Gamma} F(z) z^{n-1} dz \tag{33}$$

This involves contour integration in the complex plane and is rarely of practical use (2). Instead, we will use much the same approach we used to get the forward Z transform: get the ex-

```
DO N=0,NMAX
      S1(N) = EXP(−ALPHA1*T)*S1(N−1) + T*X(N)
      S2(N) = EXP(−ALPHA2*T)*S2(N−1) + T*X(N)
      Y(N)  = S1(N) + S2(N)
END DO
```

Figure 9. Computer program to calculate the response of the system in Fig. 8. The responses S1 and S2 are the responses of the two parallel paths in Fig. 8.

Table 2. Properties of the Z Transform

	Sampled Time Domain	Z Domain
	$f(n)$	$F(z)$
Linearity	$\alpha \cdot f(n) + \beta \cdot g(n)$	$\alpha \cdot F(z) + \beta \cdot F(z)$
Time shift	$f(n - m)$	$z^{-m}F(z)$
Initial value	$f(0)$	$\lim_{z\to 0} F(z)$
Final value	$f(\infty)$	$\lim_{z\to 1}(z - 1) \cdot F(z)$
Integration	$\sum_{n=0}^{N} f(n)$	$\frac{1}{1 - z^{-1}} F(z)$
Convolution	$\sum_{n=0}^{\infty} f(n)h(m - n)$	$F(z) \cdot H(z)$
Complex convolution	$f(n) \cdot g(n)$	$\frac{1}{2\pi j}\oint_C G(v)F\left(\frac{z}{v}\right) v^{-1}\,\delta v$

pression in a form we recognize, then look it up in a table. This involves the well-known partial fraction expansion.

Partial Fraction Expansion

Recall the previous example in which we were convolving two sequences, one an exponentially decaying function of Eq. (11)

$$h(n) = Ae^{-nT/t_0} \qquad n = 0, 1, 2, 3, \ldots \tag{11}$$

and the other the discretized unit step function of Eq. (12)

$$u(n) = 1 \qquad n = 0, 1, 2, 3, \ldots \tag{12}$$

Their Z transforms are expressed in Eq. (13)

$$Z[h(n)] = H(z) = \frac{A}{1 - e^{-T/t_0}z^{-1}} \tag{13}$$

and

$$Z[u(n)] = U(z) = \frac{1}{1 - z^{-1}}$$

The desired convolution is

$$Y(z) = H(z)U(z) = \frac{A}{1 - e^{-T/t_0}z^{-1}} \cdot \frac{1}{1 - z^{-1}} \tag{34}$$

To get the inverse Z transform, it is preferable to work with z instead of z^{-1}, so multiply the numerator and denominator by z^2:

$$Y(z) = \frac{A}{z - e^{-T/t_0}} \cdot \frac{1}{z - 1} z^2$$

For reasons that will be apparent a little later, divide both sides by z:

$$\frac{Y(z)}{z} = \frac{A}{z - e^{-T/t_0}} \cdot \frac{1}{z - 1} z \tag{35}$$

Take the partial fraction expansion of $Y(z)/z$:

$$\frac{Y(z)}{z} = \frac{B}{z - e^{-T/t_0}} + \frac{C}{z - 1} \tag{36}$$

Now solve for B and C by multiplying through by their respective denominators and evaluating at the location of the pole in the Z domain:

$$B = (z - e^{-T/t_0})\left[\frac{1}{z - e^{-T/t_0}} \cdot \frac{Az}{z - 1}\right]_{z=e^{-T/t_0}} = \frac{Ae^{-T/t_0}}{e^{-T/t_0} - 1} \tag{37a}$$

$$C = (z - 1)\left[\frac{1}{z - e^{-T/t_0}} \cdot \frac{Az}{z - 1}\right]_{z=1} = \frac{A}{1 - e^{-T/t_0}} \tag{37b}$$

Putting these values in Eq. (36), we arrive at

$$\frac{Y(z)}{z} = \left[\frac{1}{z - 1} - \frac{e^{-T/t_0}}{z - e^{-T/t_0}}\right] \frac{A}{1 - e^{-T/t_0}} \tag{38}$$

The reason for dividing by z before we did the expansion is now apparent: we can multiply both sides of Eq. (38) by z, and on the right side, we can divide the numerators and denominators by z giving

$$Y(z) = \left[\frac{1}{1 - z^{-1}} - \frac{e^{-T/t_0}}{1 - e^{-T/t_0}z^{-1}}\right] \frac{A}{1 - e^{-T/t_0}}$$

Now all the Z terms are in a format that exist in Table 1, and we get

$$\begin{aligned} y(n) &= \frac{A}{1 - e^{-T/t_0}}[1 - e^{-T/t_0}e^{-nT/t_0}]u(n) \\ &= \frac{A}{1 - e^{-T/t_0}}[1 - e^{-(n+1)T/t_0}]u(n) \end{aligned} \tag{39}$$

Partial Fraction Expansion of Multiple Roots

If multiple roots exist at one location, a modification to the partial fraction expansion is needed. Suppose we have

$$\begin{aligned} \frac{Y(z)}{z} &= \left[\frac{N(z)}{(z - p_1)(z - p_2)^r}\right] \\ &= \frac{k_1}{z - p_1} + \frac{k_{21}}{(z - p_2)} + \frac{k_{22}}{(z - p_2)^2} + \cdots + \frac{k_{2r}}{(z - p_2)^r} \end{aligned}$$

The term k_1 is calculated as explained above. The other terms are calculated by the formula

$$k_{rj} = \frac{1}{(r - j)!}\frac{d^{r-j}}{dz^{r-j}}[(z - p_2)^r Y(z)]_{z=p_2}$$

As an example, suppose we convolve the one pole filter of the previous example with a ramp function,

$$x(n) = nu(n)$$

Convolving this with $h(n)$, and going to the Z domain gives

$$Y(z) = \frac{A}{1 - e^{-T/t_0}z^{-1}} \cdot \frac{z^{-1}}{(1 - z^{-1})^2} T \tag{40}$$

or

$$\frac{Y(z)}{z} = \frac{A}{z - e^{-T/t_0}} \cdot \frac{1}{(z-1)^2} Tz$$
$$= \frac{k_1}{z - e^{-T/t_0}} + \frac{k_{22}}{(z-1)^2} + \frac{k_{21}}{z-1}$$
$$k_1 = \left.\frac{ATz}{(z-1)^2}\right|_{z=e^{-T/t_0}} = \frac{ATe^{-T/t_0}}{(e^{-T/t_0}-1)^2}$$
$$k_{22} = \left.\frac{ATz}{(z-e^{-T/t_0})}\right|_{z=1} = \frac{AT}{(1-e^{-T/t_0})} = \frac{AT(1-e^{-T/t_0})}{(1-e^{-T/t_0})^2}$$
$$k_{21} = \frac{d}{dz}\left[\frac{ATz}{(z-e^{-T/t_0})}\right]_{z=1} = AT\left[\frac{(z-e^{-T/t_0}) - z(1)}{(z-e^{-T/t_0})^2}\right]_{z=1}$$
$$= \frac{-ATe^{-T/t_0}}{(1-e^{-T/t_0})^2}$$

Now the inverse of Eq. (40) is

$$y(n) = \frac{ATe^{-T/t_0}}{(e^{-T/t_0}-1)^2}[(e^{-nT/T_0}-1) + (e^{-T/t_0}-1)n]u(n)$$

Cross Multiplication

There is an alternative to the partial fraction expansion method that is often easier to implement, particularly when dealing with complex roots. We start by illustrating its use on an earlier problem and then move to an example with complex roots. Starting with Eq. (36) and cross multiplying the terms in the denominators, we can set this equal to Eq. (35)

$$\frac{Y(z)}{z} = \left[\frac{B(z-1) + C(z-e^{-T/t_0})}{(z-e^{-T/t_0})(z-1)}\right] = \frac{ATz}{(z-e^{-T/t_0})(z-1)}$$

Equating the numerators gives

$$B(z-1) + C(z-e^{-T/t_0}) = ATz$$

and then, by equating like powers of z

$$B = -Ce^{-T/t_0}$$

and

$$C(1-e^{-T/t_0})z = ATz$$

which leads to the same values as Eq. (37). The advantage to this approach will become apparent with more complicated expressions.

Suppose we go back to the *RLC* filter problem and calculate the expression for the response to a unit step function. Eq. (25) becomes

$$Y(z) = \frac{0.0094z^{-1}}{1 - 1.895 \cdot z^{-1} + 0.9044z^{-2}} \frac{1}{1-z^{-1}}$$

Expanding this out into the separate terms, we get the following general form

$$Y(z) = \frac{0.0094z}{z^2 - 1.895 \cdot z^1 + 0.9044} \cdot \frac{z}{z-1}$$
$$= \frac{K_1 z^2 + K_2 z}{z^2 - 1.895 \cdot z^1 + 0.9044} + \frac{K_3 z}{z-1} \quad (41)$$

It is best to start by solving for K_3. First divide through by $z/(z-1)$ and solve:

$$K_3 = \left[\frac{0.0094z}{z^2 - 1.895 \cdot z^1 + 0.9044}\right]_{z=1}$$
$$= \frac{0.0094}{1 - 1.895 + 0.9044} = \frac{0.0094}{0.0094} = 1$$

Hereafter, however, we are reduced to cross multiplying

$$\frac{0.0094z}{z^2 - 1.895 \cdot z + 0.9044} \cdot \frac{z}{z-1}$$
$$= \frac{(K_1 z^2 + K_2 z)(z-1) + z(z^2 - 1.902 \cdot z + 0.9044)}{(z^2 - 1.895 \cdot z + 0.9044)(z-1)}$$

Equating like powers of z in the numerator gives

$$0 = K_1 z^3 + z^3 \Rightarrow K_1 = -1$$
$$0.0094z^2 = K_2 z^2 + K_1 z^2 - 1.902 \cdot z^2 \Rightarrow K_2 = 0.9044$$

and the accuracy can be checked by the remaining equation

$$0 = -K_2 z + 0.9044z$$

Now plugging these numbers back into Eq. (43) gives

$$Y(z) = \frac{0.0094z}{z^2 - 1.895 \cdot z^1 + 0.9044} \cdot \frac{z}{z-1}$$
$$= \frac{-z^2 + 0.9044z}{z^2 - 1.895 \cdot z^1 + 0.9044} + \frac{z}{z-1}$$
$$= \frac{1}{1-z^{-1}} - \frac{1 - 0.9044z^{-1}}{z - 1.895 \cdot z^{-1} + 0.9044z^{-2}}$$
$$= \frac{1}{1-z^{-1}} - \frac{1 - 0.9044z^{-1}}{z - 2 \cdot e^{-\alpha T}\cos(\beta T) \cdot z^{-1} + e^{-2\alpha T}z^{-2}}$$

This is starting to look like something from Table 1, but the last term resembling the decaying cosine isn't quite there. The following manipulation is required:

$$Y(z) = \frac{1}{1-z^{-1}} - \frac{1 - 0.9044z^{-1}}{z - 2 \cdot e^{-\alpha T}\cos(\beta T) \cdot z^{-1} + e^{-2\alpha T}z^{-2}}$$
$$= \frac{1}{1-z^{-1}} - \frac{1 - e^{-\alpha T}\cos(\beta T)z^{-1}}{z - 2 \cdot e^{-\alpha T}\cos(\beta T) \cdot z^{-1} + e^{-2\alpha T}z^{-2}}$$
$$+ \frac{0.521 \cdot e^{-\alpha T}\sin(\beta T)z^{-1}}{z - 2 \cdot e^{-\alpha T}\cos(\beta T) \cdot z^{-1} + e^{-2\alpha T}z^{-2}}$$

Notice that it was necessary to break the last term into two parts to give two terms that can be found in Table 1

$$y(n) = [1 - e^{-\alpha nT}\cos(\beta nT) - 5.21 \cdot e^{-\alpha nT}\sin(\beta nT)] \quad (42)$$

STABILITY

In dealing with the transfer function $H(z)$ of a discrete system, a key issue is the stability of this system. By stability we are saying that the output remains bounded for any bounded input (3). An Nth order causal system has a transfer

function which can be expressed as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 z^N + b_1 z^{N-1} + \cdots + b_{N-1} z + b_N}{a_0 z^N + a_1 z^{N-1} + \cdots + a_{N-1} z + a_N} \tag{43}$$

In determining the response to an input $X(z)$, the output can be expressed in the following manner after factoring the denominator into its roots and taking the partial fraction expansion

$$Y(z) = \frac{c_1 z}{z - p_1} + \frac{c_2 z}{z - p_2} + \cdots + \frac{c_N z}{z - p_N} + Y_I(z) \tag{44}$$

(For this example, it is assumed there are no repeated roots.) $Y_I(z)$ contains only those terms that originated from the poles of the input $X(z)$. Taking the inverse transform of Eq. (44) gives

$$y(n) = c_1 p_1^n + c_2 p_2^n + \cdots + c_N p_N^n + y_I(n) \tag{45}$$

Each of the poles of Eq. (44) produced an exponential term in Eq. (45). As long as the magnitude of each of the poles in Eq. (44) is less than one, then each of the terms in Eq. (45) is exponentially decaying. The term $y_I(n)$ originated from the input, which we have already assumed is bounded. Therefore, looking at Fig. 10, we can say that the system is stable if each pole is inside the unit circle in the complex Z plane.

The *RLC* filter that we analyzed earlier had the following transfer function:

$$H(z) = 1.155 \times 10^6 \frac{e^{-\alpha T} \sin(\beta \cdot T) \cdot z^{-1}}{1 - 2e^{-\alpha T} \cos(\beta \cdot T) \cdot z^{-1} + e^{-2\alpha T} z^{-2}}$$
$$= \frac{0.0818 z^{-1}}{1 - 1.895 z^{-1} + 0.904 z^{-2}}$$

The denominator has its roots at $z = 0.9475 + j\ .079$ and $0.9475 - j\ .079$, forming a complex conjugate pair. Most important is that $|z| = 0.9508$; both roots have a magnitude of less than one and lie inside the unit circle.

ALTERNATIVE METHODS TO FORMULATE THE Z TRANSFORM

We have seen examples in which problems were stated in the frequency domain and we solved them in the sampled time domain. Our approach has been to take the partial fraction expansion of the frequency domain expression, find the corresponding Z transforms, and solve the problem in the Z domain. Our success depended upon the ability to manipulate the frequency domain expression in a form that could be found in a table like Table 1. In this section, we present some alternatives.

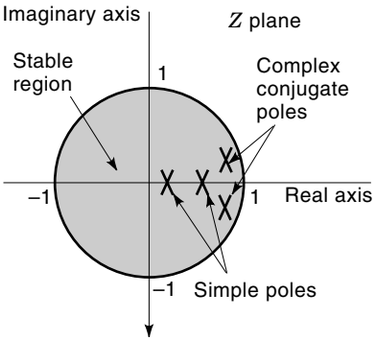


Figure 10. Complex Z plane. All poles must be inside the unit circle (shaded area) to insure stability.

Backward Rectangular Approximation

Fourier Transform Theory tells us that a $j\omega$ in the frequency domain becomes a derivative in the time domain (1,2). In going from the time domain to the sampled time domain, the derivative may be approximated by

$$\frac{df(t)}{dt} \cong \frac{f[nT] - f[(n-1)T]}{T}$$

Taking the Z transform of the right side gives

$$Z\left\{\frac{f[nT] - f[(n-1)T]}{T}\right\} = \frac{F(z) - z^{-1}F(z)}{T} = \frac{1 - z^{-1}}{T} F(z)$$

Suppose this is taken one step further and the transition from the frequency domain to the Z domain is made by simply replacing

$$j\omega \Rightarrow \frac{1 - z^{-1}}{T} \tag{46}$$

As an example, the transition from the frequency domain to the Z domain for the one pole filter becomes

$$\frac{1}{\alpha + j\omega} \Rightarrow \frac{1}{\alpha + \dfrac{1 - z^{-1}}{T}} = \frac{T}{\alpha T + 1 - z^{-1}} \tag{47}$$

At first glance, this does not seem in any way, shape, or form to represent the Z transform in Table 1. An approximation that is useful here and elsewhere is

$$\frac{1}{1 + \delta} \cong e^{-\delta} \qquad \text{if} \qquad \delta \ll 1$$

Utilizing this approximation Eq. (47) becomes

$$\frac{T}{\alpha T + 1 - z^{-1}} = \frac{T/(1 + \alpha T)}{1 - z^{-1}/(1 + \alpha T)} \cong \frac{T e^{-\alpha T}}{1 - e^{-\alpha T} z^{-1}} \tag{48}$$

There are two points worth noting. First, the factor T that we usually add in the convolution theorem is already there because the substitution of Eq. (48) is essentially Riemann Integration, that is, approximating an integral by a summation at specific intervals of size T. Furthermore, the amplitude has been changed by a factor of

$$\frac{1}{1 + \alpha T} \cong e^{-\alpha T}$$

Once again, if αT is small, this term will be very close to 1. This means that more accuracy can be obtained by making T smaller.

The practical reasoning for this approach is a little clearer if we go back to the *RLC* circuit of Fig. 5, which had the trans-

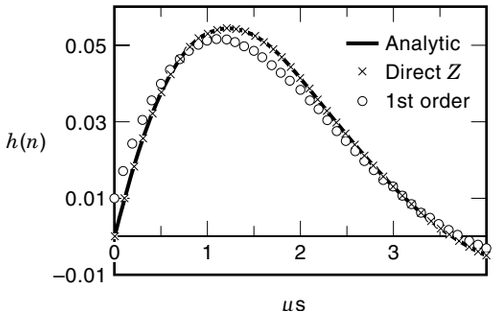


Figure 11. Impulse response of the second-order *RLC* circuit as calculated analytically (dashed curve), by the direct Z transforms ($\times$), and by the first-order backward rectangular approximation ($\bigcirc$). Notice that the latter is not as accurate. However, it can be made arbitrarily close to the analytic solution by using smaller time steps.

fer function

$$H(\omega) = \frac{\omega_0^2}{\omega_0^2 + j\omega 2\delta_0\omega_0 - \omega^2}$$

where

$$\omega_0 = 1/LC \qquad \delta_0 = R/2L\omega_0$$

Instead of having to transform this to the form in Table 1, replace each $j\omega$ with $(1 - z^{-1})/T$:

$$\begin{aligned} H(z) &= \frac{\omega_0^2}{\omega_0^2 + \left(\dfrac{1-z^{-1}}{T}\right) 2\delta_0\omega_0 + \left(\dfrac{1-z^{-1}}{T}\right)^2} \\ &= \frac{\omega_0^2 T^2}{\omega_0^2 T^2 + (1-z^{-1})2\delta_0\omega_0 T + (1 - 2z^{-1} + z^{-2})} \\ &= \frac{\omega_0^2 T^2}{(\omega_0^2 T^2 + 2\delta_0\omega_0 T + 1) - 2(1 + \delta_0\omega_0 T)z^{-1} + z^{-2}} \end{aligned} \tag{49}$$

The resemblance to the Z transform of Table 1 is not as obvious, but it is there. Figure 11 shows the impulse response using Eq. (49) compared to the results obtained from Eq. (26). These plots were made using a time step of 0.1 μs. If the time step is reduced to 0.01 μs, the results correspond almost exactly.

Trapezoidal Approximation (Bilinear Transform)

Equation (46) can be improved upon by the following transformation:

$$j\omega \Rightarrow \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \tag{50}$$

This is referred to as the bilinear transform. While the use of Eq. (46) can be thought of as approximating the time domain convolution integral with rectangular step Reimann integration, Eq. (50) represents trapezoidal integration. Equation (50) is preferred by theoreticians in the signal processing field because it is unconditionally stable, whereas Eq. (46) is not (3). From a somewhat more intuitive view, it is more accurate because trapezoidal integration is more accurate than rectangular integration. The disadvantage is obvious: the order of the system in the Z domain is doubled!

As a simple example, take the Z transform of the one pole function using the bilateral transform (4)

$$\begin{aligned} \frac{1}{\alpha + j\omega} &= \frac{1}{\alpha + \dfrac{2}{T}\dfrac{1-z^{-1}}{1+z^{-1}}} = \frac{(1+z^{-1})\cdot T}{(1+z^{-1})\cdot\alpha T + 2\cdot(1-z^{-1})} \\ &= \frac{(1+z^{-1})\cdot T}{(\alpha T + 2) + (\alpha T - 2)z^{-1}} = \frac{(1+z^{-1})\cdot \dfrac{T}{2+\alpha T}}{1 + \dfrac{1-\alpha T/2}{1+\alpha T/2}z^{-1}} \end{aligned} \tag{51}$$

Figure 12 compares the formulation of Eq. (48) with the new bilateral formulation from Eq. (51) and with the analytic formulation of Eq. (22). Clearly Eq. (51) is more accurate, after the first pulse. (Notice that $1 + z^{-1}$ in the numerator of Eq. (51) means that the impulse response is calculated by averaging the impulse over the first two time steps.)

The different approaches used to change from the Fourier domain to the Z domain can be summarized as follows: The direct transform, that is, converting from terms in the frequency domain to those in the Z domain by looking them up in a table, is the most accurate and also gives the lowest order expression in the Z domain. The disadvantage of the direct transform is that it often requires a partial fraction expansion. For third and higher order systems, this is not trivial. Using direct substitution via either the backward approximation or the bilateral transform is usually easier, however, these are approximations. The bilateral transform is better than the backward approximation, at the cost of increased complexity.

Most authors describe these transforms starting from the Laplace domain (3,4). The concepts are the same, but begin with s instead of $j\omega$.

AN EXAMPLE FROM ELECTROMAGNETIC SIMULATION

The past decade has seen a dramatic increase in the use of computer simulation methods for a wide variety of applications in electromagnetics. In particular, time domain methods, such as the finite-difference time-domain (FDTD) method have become more popular because of their flexibility and

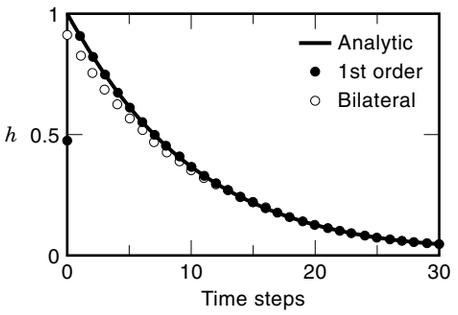


Figure 12. Impulse response of the single-pole filter as calculated analytically (dashed curve), by the first-order backward rectangular approximation ($\bigcirc$), and by the second-order bilateral approximation ($\bullet$). The bilateral is more accurate, but requires twice as many terms.

their efficiency in utilizing the power of today's computers (5). In this example, it will be demonstrated that the Z transform can be used in formulating the FDTD simulation for complex materials (6,7).

The time dependent Maxwell's equations are given by

$$\frac{dD}{dt} = \nabla \times H \tag{52a}$$

$$D(\omega) = \epsilon_0 \cdot \epsilon_r^*(\omega) \cdot E(\omega) \tag{52b}$$

$$\frac{dH}{dt} = \frac{1}{\mu_0} \nabla \times E \tag{52c}$$

where μ_0 is the permeability and ϵ_0 is the permittivity of free space. It will be assumed that we are dealing with nonmagnetic materials. However, the relationship between the flux density D and the electric field E can be an extremely complicated function of frequency.

In implementing the FDTD method, Eqs. (52a) and (52c) are formulated using spatial and temporal difference approximations. This is straightforward and is described extensively in the literature (5). However, we still need a method of calculating E from D. We will regard this as a digital filtering problem, and utilize the Z transforms.

Suppose we are simulating a medium described by the following complex dielectric constant

$$\epsilon_r^*(\omega) = \epsilon_r + \frac{\sigma}{j\omega\epsilon_0} + \epsilon_1 \frac{\omega_0}{(\omega_0^2 + \alpha^2) + j2\alpha\omega + \omega^2} \tag{53}$$

Inserting Eq. (53) into Eq. (52b) and taking the Z transforms, we get

$$\begin{aligned} D(z) = {} & \epsilon_r E(z) + \frac{\sigma \cdot T/\epsilon_0}{1 - z^{-1}} \cdot E(z) \\ & + \epsilon_1 \frac{e^{-\alpha T} \cdot \sin(\omega_0 T) \cdot T \cdot z^{-1}}{1 - 2e^{-\alpha T} \cos(\omega_0 T) \cdot z^{-1} + e^{-2\alpha T} z^{-2}} \cdot E(z) \end{aligned} \tag{54}$$

It will prove worthwhile to define two auxiliary parameters:

$$I(z) = \frac{\sigma \cdot T/\epsilon_0}{1 - z^{-1}} \cdot E(z) \tag{55a}$$

$$S(z) = \frac{\epsilon_1 \cdot e^{-\alpha T} \cdot \sin(\omega_0 T) \cdot T \cdot E(z)}{1 - 2e^{-\alpha T} \cos(\omega_0 T) \cdot z^{-1} + e^{-2\alpha T} z^{-2}} \tag{55b}$$

Now Eq. (54) becomes

$$D(z) = \epsilon_r E(z) + I(z) + z^{-1} S(z) \tag{56}$$

Once we have calculated $E(z)$, $I(z)$ and $S(z)$ can be calculated from Eqs. (55a) and (55b):

$$I(z) = z^{-1} I(z) + \frac{\sigma \cdot T}{\epsilon_0} E(z) \tag{57a}$$

$$\begin{aligned} S(z) = {} & 2e^{-\alpha T} \cos(\omega_0 T) \cdot z^{-1} S(z) \\ & - e^{-2\alpha T} z^{-2} S(z) + \epsilon_1 \cdot e^{-\alpha T} \cdot \sin(\omega_0 T) \cdot T \cdot E(z) \end{aligned} \tag{57b}$$

The trouble is this: in calculating $E(z)$ in Eq. (56), we need the previous value of $S(z)$, the present value of $D(z)$, and the present value of $I(z)$. The present value of $D(z)$ is not a problem because in the order in which the algorithm is implemented, it has already been calculated in Eq. (52a). However, the present value of $I(z)$ requires the present value of $E(z)$. This problem is circumvented by simply replacing $I(z)$ with its expanded version in Eq. (57a). Now Eq. (56) becomes

$$D(z) = \epsilon_r E(z) + z^{-1} I(z) + \frac{\sigma \cdot T}{\epsilon_0} \cdot E(z) + z^{-1} S(z)$$

from which $E(z)$ can be calculated by

$$E(z) = \frac{D(z) - z^{-1} I(z) - z^{-1} S(z)}{\epsilon_r + \sigma \cdot T/\epsilon_0} \epsilon_r E(z) \tag{58}$$

$$I(z) = z^{-1} I(z) + \frac{\sigma \cdot T}{\epsilon_0} \cdot E(z) \tag{57a}$$

$$\begin{aligned} S(z) = {} & 2e^{-\alpha T} \cos(\omega_0 T) \cdot z^{-1} S(z) \\ & - e^{-2\alpha T} z^{-2} S(z) + \epsilon_1 \cdot e^{-\alpha T} \cdot \sin(\omega_0 T) \cdot T \cdot E(z) \end{aligned} \tag{57b}$$

Note that $S(z)$ did not have to be expanded out because it already had a z^{-1} in the numerator.

SUMMARY

The Z transform plays the same role in discrete time that the Laplace and Fourier transforms play in continuous time. As shown earlier in this article, it can be used to analyze discrete time equations, develop iterative solutions to discrete time equations, or design digital circuitry to calculate a solution in hardware. It has even found application in electromagnetic simulation.

The theory and examples in this article are by no means complete. However, they illustrate the power and flexibility of the Z transform as one of the essential tools in electrical engineering today.

BIBLIOGRAPHY

1. R. E. Ziemer, W. H. Tranter, and D. R. Fannin, *Signals and Systems—Continuous and Discrete,* New York: Macmillan, 1983.
2. C. L. Phillips and J. M. Parr, *Signals, Systems, and Transforms,* Upper Saddle River, NJ: Prentice-Hall, 1995.
3. E. P. Cunningham, *Digital Filtering: An Introduction,* Princeton, NJ: Houghton Mifflin, 1992.
4. A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1975.
5. A. Taflove, *Computational Electrodynamics—the Finite-Difference Time-Domain Method,* Norwood, MA: Artech House, 1995.
6. D. M. Sullivan, A frequency-dependent FDTD method using Z transforms, *IEEE Trans. Antennas Propag.,* **AP-40**: 1223–1230, 1992.
7. D. M. Sullivan, Z transform theory and the FDTD method, *IEEE Trans. Antennas Propag.,* **AP-44**: 28–34, 1996.

DENNIS M. SULLIVAN
University of Idaho