To My Parents . . .

Who Always Wonder

Where I Am Headed

To My Wife . . .

Who Is Always With Me

Even When I Am Lost

To My Son . . .

Who Shows Me

Where To Go

Numerical Analysis
theory and practice
N.S. Asaithambi
Saunders, 1995
pp. iii — 80

Preface

The material in this text derives from the lecture notes of a variety of numerical analysis courses. The text is designed primarily for undergraduate and graduate students in mathematics, sciences, engineering, and computer science. It explores the mathematical and computational aspects of the subject of numerical analysis, and its major themes are the development, analysis, implementation, and intelligent use of numerical methods and related software.

The text may be used for either the one-semester numerical methods course or for the two-semester numerical analysis course. Mathematical prerequisites for its use in a methods course are at least one year of the college calculus sequence and coursework in the fundamentals of matrix algebra. Mathematical prerequisites for its use in an analysis course include at least three semesters of the college calculus sequence (through multivariable calculus), linear algebra, and an introduction to differential equations. Students should also be familiar with at least one programming language.

Although users of numerical methods come from a variety of disciplines and backgrounds, all have some goals in common. Following are the major goals of a numerical analyst (1) to design algorithms specific to an application situation, (2) to implement the algorithms in a given computing environment, and (3) to understand the purpose and limitations of available mathematical software. In order to facilitate effective use of numerical methods for solving problems, numerous mathematical software packages and systems have been developed over the past several years. However, a truly effective use of any available piece of software requires a considerable amount of computational experience with it. These three major goals could be accomplished with (a) a theoretical knowledge of the subject of numerical analysis, (b) a knowledge of available software and computing environments, and (c) a lot of computational experience.

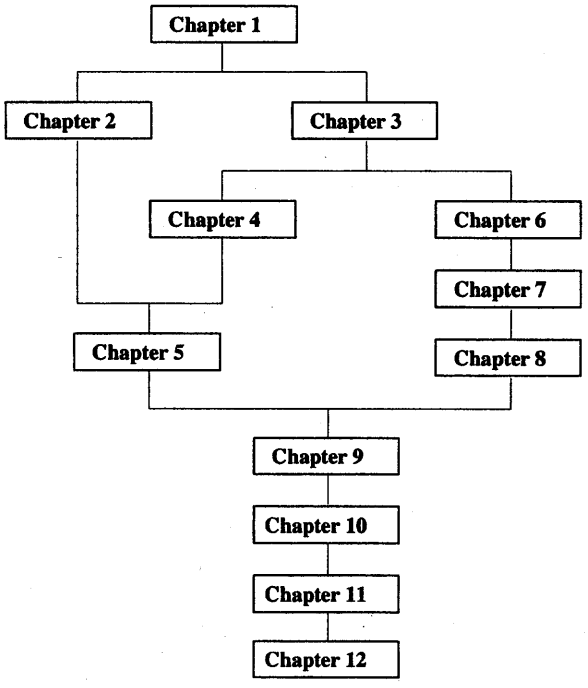
Theoretical knowledge enables the numerical analyst to understand the problem being solved so that he or she can derive, analyze, and test a numerical method for its solution. Such knowledge must also include error analysis of numerical methods and an idea of when a given numerical method will perform well or poorly. **A knowledge of the available software** and computing environments is of practical significance to the users of numerical methods and will enable them to become "intelligent users" of mathematical software. This knowledge is especially important because a standard program may not be directly useful for the solution of many real problems. An intelligent user of numerical methods should be in a position to adapt standard programs for new situations he or she encounters or to develop new methods as may be warranted. An

important aspect of computing environments is the finite precision arithmetic of actual computations as opposed to the exact arithmetic assumed in theoretical discussions. **Computational experience** bridges the gap between the theoretical discussions and the expected or unexpected behavior of numerical methods when applied to practical problems. Quite often computational experience is also helpful to determine whether a mathematical model accurately represents the physical problem of interest.

This text exposes students to the development, analysis, and implementation of numerical methods for the solution of standard mathematical problems that arise in many disciplines. Theory is emphasized as much as software development and computational experience. We hope students will gain more practical knowledge using this approach. Adequate theoretical discussion is included in order to explain the behavior of numerical methods on different kinds of problems. Students are given ample opportunity to investigate numerical methods via hand computation, program development, and experimentation. Individual instructors may encourage the use of high-quality software packages or systems such as IMSL, Mathcad, Mathematica, Matlab, and NAG for experimentation purposes. The text does not, however, discuss these items.

Organization

Sections of the text marked by a star—in the table of contents and within the text—may be skipped in the methods course with no loss of continuity. We will call these sections starred sections, and the others unstarred. The one-semester methods course may be taught using only the unstarred sections in Chapters 1–8. The entire text may be used for the two-semester analysis course. The following diagram shows the chapter dependencies within the text.



Rather than follow the syntax of a particular programming language such as Fortran or Pascal, we present algorithms in a generic notation to enable students to code the algorithms in a programming language of their own choice. Sources of mathematical software presently available for different kinds of computers are discussed in the Software Survey section at the end of each chapter. An *Instructor's Manual* is available with the text. The manual includes program listings for all the algorithms and detailed solutions to all exercises in the text. A $3\frac{1}{2}''$ disk, formatted for the DOS platform, is supplied with the *Instructor's Manual*. The disk contains the solutions to all the computer exercises and the programs corresponding to the algorithms described in the text. The programs are written in Fortran, Pascal, and C.

Acknowledgments

I am thankful to Jimmy Solomon, Charles Mastin, Paul Spikes, Charles Scarborough, and Betty Scarborough of Mississippi State University, and Richard Winchester of Lincoln University for their enthusiastic support and encouragement during the development of this textbook. I extend my thanks to the many students at Mississippi State University and Lincoln University for their comments and suggestions as the manuscript was being prepared. I also wish to thank the many capable reviewers whose valuable suggestions greatly improved the manuscript. A list of their names follows this preface.

I wish to express my gratitude to my parents, whose many sacrifices made it possible for me to pursue higher education in the United States. I thank my wife, Sasi, and my son, Ganesh, without whose patience, understanding, and support this book would have remained just a dream.

List of Reviewers

Stephen Beck
Bloomsburg University
Barbara Bertram
Michigan Technological University
Richard T. Bumby
Rutgers University
Chris Corey
Utah State University
Philip Crooke
Vanderbilt University
Bruce Edwards
University of Florida
Raymond C. Ellis
David R. Hill
Temple University
George J. Fix
University of Texas at Arlington
Giles Maloof
Boise State University
C. Wayne Mastin
Mississippi State University
Alexander P. Morgan
General Motors Research Laboratories
Martin Rill
Shippensburg University
John Strikwerda
University of Wisconsin at Madison

I would like especially to thank Neil E. Berger of the University of Illinois at Chicago, Abdou Youssef, George Washington University, and Giles Maloof, Boise State University, for their contributions as accuracy reviewers. I would also like to express my appreciation to the editorial Staff at Saunders College Publishing, especially Jay Ricci, Bonnie Boehme, and Jay Freedman.

N. S. A.

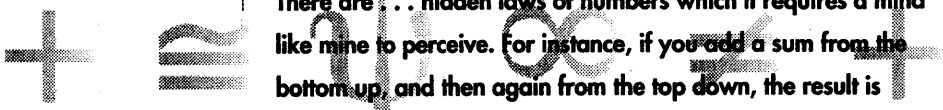
Glossary of Notation

$[a, b]$	Set of real numbers x satisfying $a \leq x \leq b$ (4)
(a, b)	Set of real numbers x satisfying $a < x < b$ (4)
$f^{(n)}(x)$	n^{th} derivative of $f(x)$ (5)
$\text{Abs}(x_A)$	Absolute error in x_A (16)
$\text{Rel}(x_A)$	Relative error in x_A (16)
$fl(x)$	Floating point representation of x (18)
$O(\cdot)$	Rate of convergence (big Oh) (33)
$\leftarrow$	Assignment operation in an algorithm (38)
ε	Error tolerance (58)
α	The actual root (58)
$x^{(k)}$	k^{th} approximation to the root (59)
x^*	Variable denoting $x^{(n)}$ (60)
e_n	Error in the n^{th} iterate (84)
p	Order of convergence (84)
A or $[a_{ij}]$	Matrix A with entries a_{ij} (128)
$\mathbf{x}$	An n-vector (132)
A^T	Transpose of A (132)
I_n or I	Identity matrix (136)
A^{-1}	Inverse of matrix A (136)
$\mathbf{0}$	Vector of all zeros (137)
$\det(A)$	Determinant of A (137)
$[A \mid \mathbf{b}]$	Augmented matrix (145)
L, U	Factors of A (164)
$\|\mathbf{x}\|$	Vector norm for $\mathbf{x}$ (196)
$\|\mathbf{x}\|_1, \|\mathbf{x}\|_2, \|\mathbf{x}\|_\infty$	The 1, 2, ∞ norms of $\mathbf{x}$ (196)
$\|\mathbf{A}\|$	Matrix norm for A (202)
$\|\mathbf{A}\|_1, \|\mathbf{A}\|_\infty$	The 1 and ∞ norms of A (204)
$\rho(A)$	Spectral radius of A (207)
$\|\mathbf{A}\|_2$	The 2-norm of A (208)
$\kappa(A)$	Condition number of A (216)
ω	Acceleration parameter for SOR (242)

$T_k(z)$	The k^{th}-degree Chebyshev polynomial (248)
$J_F(\mathbf{x})$	Jacobian of $\boldsymbol{F}$ at $\mathbf{x}$ (266)
∇g	Gradient of g (296)
$B_n(x)$	The n^{th}-degree Bernstein polynomial (311)
$L_{n,k}(x)$	The k^{th} Lagrange polynomial of degree n (317)
$f[x_i, \cdots, x_{i+k}]$	The k^{th} divided-difference of f (329)
Δ	Forward difference operator (342)
∇	Backward difference operator (344)
$H_{n,k}(x)$, $\hat{H}_{n,k}(x)$	Hermite polynomials (354)
$\langle f, g \rangle$	Inner product of f and g (412)
$\mathcal{L}_n(x)$	Legendre least-squares approximation (417)
$\mathcal{C}_n(x)$	Chebyshev least-squares approximation (435)
$\mathcal{I}_n(x)$	The optimal interpolant (436)
$DFT(\mathbf{x})$	Discrete Fourier Transform of $\mathbf{x}$ (443)
w_i	Approximation to $y(t_i)$ (561)
$\mathbf{w}_i$	Approximate solution of a system of IVPs (630)
Λ	Diagonal matrix of eigenvalues (660)
$\mathbf{z}^{(0)}$	Initial vector for power method (676)
$B(x)$	Cubic B-spline (758)
v_m^n	Approximation to $u(x_m, t_n)$ (777)
$N_j^{(i)}(x, y)$	Shape function j for triangle i (816)

Contents

CHAPTER 1 Preliminaries



There are . . . hidden laws of numbers which it requires a mind like mine to perceive. For instance, if you add a sum from the bottom up, and then again from the top down, the result is always different.

Mrs. La Touche

OUTLINE

The theory of projectiles is a subject of interest to students of elementary physics. The height $y(x)$ reached by a projectile at distance x from the point of projection is given by

$$y(x) = (\tan\alpha)\cdot x - \frac{g}{2(v_0\cos\alpha)^2}\cdot x^2, \tag{1.1}$$

where v_0 is the initial velocity and α is the angle of projection. (See Fig. 1.1.)

Corresponding to $\alpha = \pi/3$, $v_0 = 100$ m/s, and $g = 9.8$ m/s^2, the height $y(x)$ for $x = 500$ determined by (1.1) is

$$y(500) = 500\sqrt{3} - \frac{9.8}{2(100\cdot 0.5)^2}\cdot(500)^2 \approx 376 \text{ m}.$$

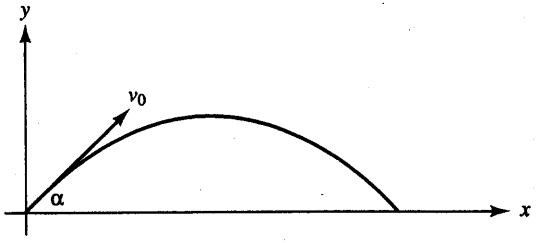


Figure 1.1 Locus of a projectile.

It is an important task of a physicist to determine how small inaccuracies in the parameters involved in the calculations affect the final result. This is frequently called *sensitivity analysis* or *perturbation analysis.* For instance, consider decreasing v_0 by 2% in the present calculation. Then, $v_0 = 98$ m/s, and the height $y(x)$ for $x = 500$ determined by (1.1) is

$$y(500) = 500\sqrt{3} - \frac{9.8}{2(98 \cdot 0.5)^2} \cdot (500)^2 \approx 356 \text{ m}.$$

Therefore, the result has decreased roughly by 5.32%. On the other hand, if we let $v_0 = 100$, and consider decreasing α by 2%, we have $\alpha = 0.98\pi/3$, $\tan\alpha \approx 1.6512$, $\cos\alpha \approx 0.5180$, and

$$y(500) = 500(1.6512) - \frac{9.8}{2(100 \cdot 0.5180)^2} \cdot (500)^2 \approx 369 \text{ m}.$$

Note that this corresponds to a decrease of only 1.86%. Thus, in this example, a small change in v_0 affects the result much more than does a small change in α.

When there are many parameters involved, it may not be practical to consider the effect of "changing" one parameter at a time. Moreover, in physical studies the "changes" in many parameters may occur simultaneously as measurement errors or uncertainties. The study of effects of small changes in the data arises naturally in the design of numerical methods for solving mathematical problems.

The subject of numerical analysis deals with the design, implementation, testing, and analysis of numerical methods. In this book, we will consider a variety of mathematical problems for which numerical methods will be developed and analyzed. This chapter contains a brief review of selected topics from elementary calculus, along with an introduction to the basic ideas related to error analysis, computer arithmetic, propagation of errors, convergence and stability of computations, algorithms, and programming.

1.1 Theory

1.1.1 REVIEW OF CALCULUS

This book assumes that you are familiar with the basic concepts of real and complex number systems, limits and continuity, sequences and series, and differentiation and integration, which are normally covered in the undergraduate calculus and analytical geometry sequence. A short review of some of these topics is given in Appendix A. For the last three chapters, a knowledge of ordinary and partial differential equations is assumed. This section contains a review of some basic theorems from calculus that will be used frequently in the book. Most of the results are presented unreferenced and are available in standard undergraduate calculus textbooks (for example, see Berkey [4]).

THEOREM

1.1 (Intermediate-Value Theorem)

Suppose $f(x)$ is a continuous function on the interval $[a, b]$, and K is any number between $f(a)$ and $f(b)$. Then there exists a number $c \in [a, b]$ such that $f(c) = K$. (See Fig. 1.2.) □

A formal proof of this theorem is not usually given in the basic calculus course, but is available in most advanced calculus texts (for example, see Fulks [7]). However, this result is intuitively clear. For example, consider the altitude $H(t)$ of an airplane at time t minutes after takeoff. Clearly, at $t = 0$ we must have $H = 0$. Because of the physical situation, it is clear that $H(t)$ is a continuous function of t. Hence, if H is 10,000 feet at $t = 10$ and 20,000 feet at $t = 15$, then the plane must reach every altitude between 10,000 feet and 20,000 feet between $t = 10$ and $t = 15$.

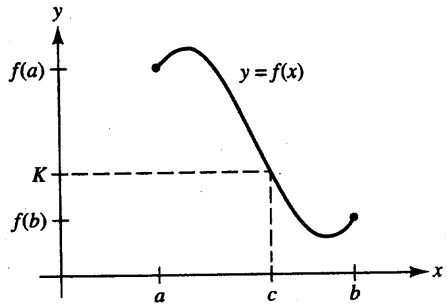


Figure 1.2 Intermediate-value theorem.

THEOREM

1.2 (Rolle's Theorem)

Suppose $f(x)$ is a continuous function on the interval $[a, b]$ and f is differentiable on (a, b). If $f(a) = f(b)$, then there exists at least one number $c \in (a, b)$ such that $f'(c) = 0$. (See Fig. 1.3.) □

We now illustrate the use of the two theorems with some examples. Suppose $f(x)$ is a given function. A number x^* for which $f(x^*) = 0$ is called a solution of the equation $f(x) = 0$, or more commonly a **root** of $f(x) = 0$. The task of locating a root of $f(x) = 0$, which arises quite often in various contexts, is called the **rootfinding** problem. The simplest way to solve the rootfinding problem is to sketch the graph of $f(x)$ and locate x^* as a point at which the graph of $f(x)$ intersects the x-axis. In other words, we must search for x^* on the x-axis. Where do we start searching, and for how long? Theorems 1.1 and 1.2 may be used to obtain some clues!

EXAMPLE 1.1

Show that the equation $x^3 - 2x - 5 = 0$ has a solution in the interval $[2, 3]$.

SOLUTION

Let $f(x) = x^3 - 2x - 5$, and note that $f(x)$ is continuous on the interval $[2, 3]$. Then $-1 = f(2) \leq 0 \leq f(3) = 16$. With $K = 0$ in Theorem 1.1, we may conclude that there is a number $c \in [2, 3]$ such that $f(c) = 0$. ▲

EXAMPLE 1.2

Show that the equation $x^3 + 4x + k = 0$, where k is any real number, has exactly one real root.

SOLUTION

Let $f(x) = x^3 + 4x + k$. For large positive x, $f(x)$ is positive, and for large negative x, $f(x)$ is negative. Hence, by Theorem 1.1, there exists a number c such that $f(c) = 0$; *i.e.*, there is at least one real root for the equation $x^3 + 4x + k = 0$.

Suppose $f(x) = 0$ has more than one real root. Consider two such roots, say a and b. Then, $f(x)$ is a continuous function on the interval $[a, b]$ with $f(a) = f(b) = 0$.

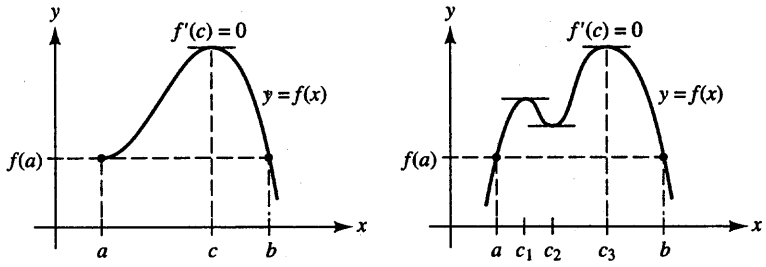


Figure 1.3 Rolle's theorem.

Then, by Theorem 1.2 there must be a point $c \in (a, b)$ such that $f'(c) = 0$. However, since $f(x) = x^3 + 4x + k$, at any real number c we have $f'(c) = 3c^2 + 4$, which is always positive—a contradiction. Therefore, $f(x) = 0$ must have only one real root. ▲

Note that Theorems 1.1 and 1.2 provide only clues regarding the roots of $f(x) = 0$; they do not provide any means for obtaining the roots. We will consider numerical methods for solving the rootfinding problem in detail in Chapter 2.

By applying Theorem 1.2 successively to $f, f', \cdots, f^{(n-1)}$ we obtain the following theorem.

THEOREM

1.3 (Generalized Rolle's Theorem)

Suppose $f(x)$ is a continuous function on the interval $[a, b]$, and f is n times differentiable on (a, b). If $f(x) = 0$ at the $n + 1$ distinct numbers $x_0, x_1, \cdots, x_n$ in $[a, b]$, then there exists a number $c \in (a, b)$ such that $f^{(n)}(c) = 0$. □

EXAMPLE 1.3

Suppose $f(x)$ and $g(x)$ are two functions that are n times continuously differentiable on the interval $[a, b]$. Suppose there exist $n + 1$ distinct numbers $x_0, x_1, \cdots, x_n$ in $[a, b]$ such that $f(x_i) = g(x_i)$ for $i = 0, 1, \cdots, n$. Then show that there exists a number $c \in (a, b)$ such that $f^{(n)}(c) = g^{(n)}(c)$.

SOLUTION

Define $h(x) = f(x) - g(x)$. Then $h(x)$ is n times continuously differentiable on $[a, b]$ and $h(x) = 0$ at the $n + 1$ distinct points $x_0, x_1, \cdots, x_n$. Then, by Theorem 1.3, it follows that there exists a number $c \in (a, b)$ such that $h^{(n)}(c) = 0$. That is, $f^{(n)}(c) = g^{(n)}(c)$. ▲

In the analysis of numerical methods, the behavior of errors in computations is important. Usually we obtain expressions for errors that involve values of certain functions or their derivatives at some unknown points. The following theorems are often useful for obtaining bounds or estimates for such error expressions.

THEOREM

1.4 (Extreme-Value Theorem)

Suppose $f(x)$ is a continuous function on the interval $[a, b]$. Then there exist numbers c_1 and c_2 in $[a, b]$ with the property that for all x in $[a, b]$, $f(c_1) \leq f(x) \leq f(c_2)$. □

The proofs for Theorems 1.5 through 1.8 are given in Appendix A.

THEOREM

1.5 (Weighted Mean-Value Theorem for Sums)

Suppose $f(x)$ is a continuous function on the interval $[a, b]$. Let $x_1, x_2, \cdots, x_n$ be points in $[a, b]$, and let $w_1, w_2, \cdots w_n$ be real numbers ***all of one sign****. Then there exists a*

number $c \in [a, b]$ *such that*

$$\sum_{j=1}^{n} w_j f(x_j) = f(c) \sum_{j=1}^{n} w_j.$$

□

EXAMPLE 1.4

Let $f(x)$ be continuous on $[a, b]$. Let $S = \sum_{j=1}^{n} f(x_j)$, where $x_1, x_2, \cdots, x_n \in [a, b]$. Show that there exists a number $c \in [a, b]$ such that $S = n f(c)$.

SOLUTION Note that Theorem 1.5 applies to this situation with $w_j = 1$ for $j = 1, 2, \cdots, n$. ▲

THEOREM

1.6 (Integral Mean-Value Theorem)

Suppose $f(x)$ *is a continuous function on the interval* $[a, b]$, $w(x)$ *is an integrable function on* $[a, b]$, i.e.,

$$\int_a^b w(x)dx < \infty,$$

and $w(x)$ ***does not change sign*** *on* $[a, b]$. *Then there exists a number* $c \in [a, b]$ *such that*

$$\int_a^b w(x) f(x)dx = f(c) \int_a^b w(x)dx.$$

□

Remark For $w(x) \equiv 1$, Theorem 1.6 guarantees the existence of a number $c \in [a, b]$ such that

$$f(c) = \frac{1}{b-a} \int_a^b f(x)\, dx. \tag{1.2}$$

The right member of (1.2) is usually referred to as the **average value** of the function $f(x)$. (See Fig. 1.4.) Note that the hypothesis that $w(x)$ is of one sign is essential in

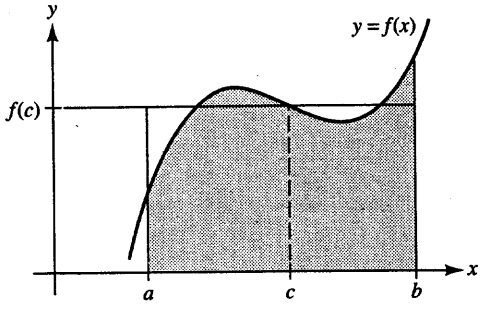


Figure 1.4 Average value.

Theorem 1.6. For example, the theorem does not hold for $f(x) = x$ and $w(x) = x$ on the interval $[-1, 1]$. ▮

EXAMPLE 1.5

Show that $\int_1^2 (1-x)(x-2)\,f(x)\,dx = \frac{1}{6}f(c)$ for some number $c \in [1, 2]$.

SOLUTION Theorem 1.6 applies with $w(x) = (1-x)(x-2)$. Notice that $w(x) \geq 0$ on $[1, 2]$. Therefore,

$$\int_1^2 (1-x)(x-2)\,f(x)dx = f(c)\int_1^2 (1-x)(x-2)\,dx = \frac{1}{6}f(c). \quad \blacktriangle$$

EXAMPLE 1.6

Consider $\int_1^3 w(x)\,f(x)\,dx$. Can Theorem 1.6 be directly applied to this integral with $w(x) = (x-1)(x-2)(x-3)$ to obtain a result similar to Example 1.5?

SOLUTION No, Theorem 1.6 does not apply to this integral because the function $w(x) = (x-1)(x-2)(x-3)$ changes sign in the interval $[1, 3]$. ▲

Taylor's Theorem and the associated Taylor series are among the most important tools in numerical analysis. The theorem gives a simple method for approximating functions $f(x)$ by polynomials. The proof of Taylor's Theorem makes use of the fundamental theorem of calculus. The Mean-Value Theorem for Derivatives is a particular instance of Taylor's Theorem and will be presented first.

THEOREM

1.7 (Mean-Value Theorem for Derivatives)

Suppose $f(x)$ is a continuous function on the interval $[a, b]$, and f is differentiable on (a, b). Then there exists a number $c \in (a, b)$ such that

$$f'(c) = \frac{f(b) - f(a)}{b - a.} \qquad \text{(See Fig. 1.5.)} \quad \square$$

EXAMPLE 1.7

Find the value of the number c guaranteed by Theorem 1.7 (Mean-Value Theorem for Derivatives) for the function $f(x) = x + \frac{1}{x}$ over the interval $[2, 3]$.

SOLUTION $f(x)$ is continuous on $[2, 3]$ and differentiable on $(2, 3)$—hence Theorem 1.7 applies. There is a number c in $(2, 3)$ such that

$$1 - \frac{1}{c^2} = f'(c) = \frac{f(3) - f(2)}{3 - 2} = \left(3 + \frac{1}{3}\right) - \left(2 + \frac{1}{2}\right) = \frac{5}{6}.$$

Hence, $c^2 = 6$ and $c = \sqrt{6}$. (The value corresponding to the negative sign of the square root does not lie in $(2, 3)$ and therefore is not guaranteed by Theorem 1.7.) ▲

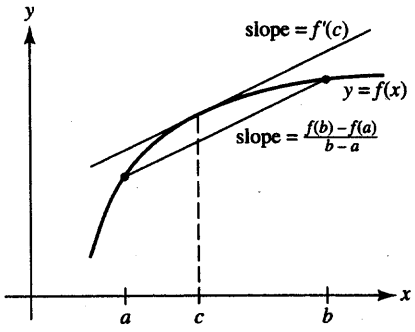


Figure 1.5 Mean-value theorem for derivatives.

Successive application of the fundamental theorem of calculus to $f, f', \cdots, f^{(n)}$, and a careful integration by parts result in Taylor's Theorem.

THEOREM

1.8 (Taylor's Theorem)

Suppose $f(x)$ has $n+1$ continuous derivatives on $[a, b]$, and x_0 is some point in $[a, b]$. Then, for all $x \in [a, b]$, there exists $\xi(x)$ in the interval containing x_0 and x satisfying

$$f(x) = P_n(x) + R_{n+1}(x),$$

where

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

$$= \sum_{k=0}^{n} (x - x_0)^k \frac{f^{(k)}(x_0)}{k!}$$

and

$$R_{n+1}(x) = (x - x_0)^{n+1} \frac{f^{(n+1)}(\xi(x))}{(n+1)!}.$$

□

$P_n(x)$ is called the **n^{th}-degree Taylor polynomial** for f about x_0 and $R_{n+1}(x)$ is called the **remainder term** associated with $P_n(x)$. The infinite series obtained by taking the limit of $P_n(x)$ as $n \to \infty$ is called the **Taylor series** for f about x_0. When $x_0 = 0$, the Taylor polynomial and series are often referred to as the **Maclaurin polynomial** and the **Maclaurin series,** respectively. The remainder term is also called the **truncation error,** indicating the error involved in using a truncated or finite summation to approximate the sum of an infinite series.

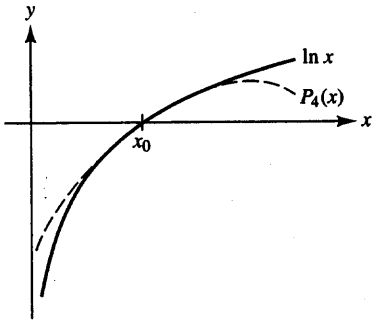


Figure 1.6 Taylor's theorem.

EXAMPLE 1.8

Use Taylor's Theorem (Theorem 1.8) to determine the Taylor polynomial of degree n for $f(x) = \ln x$ about $x_0 = 1$. Determine also the remainder term.

SOLUTION With $f(x) = \ln x$ and $x_0 = 1$, one obtains $f(x_0) = 0$, $f'(x_0) = 1$, $f''(x_0) = -1$, $f'''(x_0) = 2$, $f^{(iv)}(x_0) = -6$, etc. In general,

$$f^{(n)}(x_0) = (-1)^{n-1}(n-1)!$$

Hence, $\ln x = P_n(x) + R_{n+1}(x)$ with

$$P_n(x) = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} + \cdots + (-1)^{n-1}\frac{(x-1)^n}{n},$$

$$R_{n+1}(x) = (-1)^n \frac{(x-1)^{n+1}}{n+1}\frac{1}{\xi^{n+1}},$$

for some ξ between 1 and x. (See Fig. 1.6.) ▲

Consider using the preceding Taylor polynomial to approximate ln 1.1. With $n = 4$ and $x = 1.1$, we obtain

$$\ln 1.1 = 0.1 - \frac{(0.1)^2}{2} + \frac{(0.1)^3}{3} - \frac{(0.1)^4}{4} + \frac{(0.1)^5}{5}\frac{1}{\xi^5},$$

$$= 0.095308333 + \frac{1}{5} \times 10^{-5}/\xi^5$$

for some ξ between 1 and 1.1. It is clear that $\frac{1}{5} \times 10^{-5}/\xi^5 \leq \frac{1}{5} \times 10^{-5} = 2 \times 10^{-6}$. We may then consider 0.095308333 as an approximation to ln 1.1 with an error less than 2×10^{-6}. The exact value of ln 1.1 is 0.095310179, showing that the actual difference between the exact and the approximate values is $|0.095310179 - 0.095308333| \approx 1.85 \times 10^{-6}$, which is consistent with the bound obtained above.

EXAMPLE 1.9

Determine the degree n that will assure an accuracy of 10^{-3} when ln 1.5 is approximated by $P_n(1.5)$ using the result of Example 1.8.

SOLUTION Since $R_{n+1}(x) = (-1)^n \dfrac{(x-1)^{n+1}}{n+1} \dfrac{1}{\xi^{n+1}}$ for some ξ between 1 and x, we wish to achieve

$$|R_{n+1}(1.5)| \leq \frac{(0.5)^{n+1}}{n+1} \leq 10^{-3}$$

which is possible for $n \geq 7$. That is, $P_7(1.5)$ yields an approximation to within 10^{-3} for ln 1.5. ▲

It is also possible to express a function of two variables by using a polynomial and a remainder term. Let $f(x, y)$ and all of its partial derivatives of orders less than or equal to $n+1$ be continuous in some neighborhood of the point (x_0, y_0). Let

$$x = x_0 + \alpha, \quad y = y_0 + \beta, \quad F(t) = f(x_0 + t\alpha, y_0 + t\beta) \text{ for } 0 \leq t \leq 1.$$

Thus the function $f(x, y)$ in two variables has been expressed in terms of $F(t)$, a function in the one variable t. Since $f(x, y)$ and all of its partial derivatives of orders less than or equal to $n+1$ are continuous, $F(t)$ and all its derivatives with respect to t of orders less than or equal to $n+1$ are continuous as well (as functions of t). Further, since $t = 0$ corresponds to the point (x_0, y_0), we may use Theorem 1.8 (Taylor's Theorem) to express $F(t)$ as a polynomial in t (around $t = 0$) along with a remainder term. Finally, since $F(1) = f(x, y)$, putting $t = 1$ in the resulting expressions will yield the polynomial and remainder term for $f(x, y)$.

We have

$$\begin{aligned} F(t) &= F(0) + tF'(0) + \frac{t^2}{2!}F''(0) + \cdots + \frac{t^n}{n!}F^{(n)}(0) + \frac{t^{n+1}}{(n+1)!}F^{(n+1)}(\theta) \\ &= \sum_{k=0}^{n} \frac{t^k}{k!}F^{(k)}(0) + \frac{t^{n+1}}{(n+1)!}F^{(n+1)}(\theta) \end{aligned}$$

for some θ with $0 \leq \theta \leq t$. Therefore,

$$f(x, y) = F(1) = \sum_{k=0}^{n} \frac{1}{k!}F^{(k)}(0) + \frac{1}{(n+1)!}F^{(n+1)}(\theta)$$

for some θ with $0 \leq \theta \leq 1$. It is easily seen that $F(0) = f(x_0, y_0)$. Next, let us examine the derivatives of $F(t)$. We have

$$F'(t) = \alpha \frac{\partial f}{\partial x}(x_0 + t\alpha, y_0 + t\beta) + \beta \frac{\partial f}{\partial y}(x_0 + t\alpha, y_0 + t\beta),$$

which yields

$$\begin{aligned} F'(0) &= \alpha \frac{\partial f}{\partial x}(x_0, y_0) + \beta \frac{\partial f}{\partial y}(x_0, y_0) \\ &= \left[\left(\alpha \frac{\partial}{\partial x} + \beta \frac{\partial}{\partial y}\right) f\right](x_0, y_0) \\ &= \left[\left((x - x_0)\frac{\partial}{\partial x} + (y - y_0)\frac{\partial}{\partial y}\right) f\right](x_0, y_0). \end{aligned}$$

Similarly,

$$\begin{aligned} F''(t) &= \alpha^2 \frac{\partial^2}{\partial x^2} f(x_0 + t\alpha, y_0 + t\beta) + 2\alpha\beta \frac{\partial^2}{\partial x \partial y} f(x_0 + t\alpha, y_0 + t\beta) \\ &\quad + \beta^2 \frac{\partial^2}{\partial y^2} f(x_0 + t\alpha, y_0 + t\beta) \\ &= \left[\left(\alpha \frac{\partial}{\partial x} + \beta \frac{\partial}{\partial y}\right)^2 f\right](x_0 + t\alpha, y_0 + t\beta). \end{aligned}$$

In the above simplification, we have used the idea that

$$\left(\frac{\partial}{\partial x}\right)^i \left(\frac{\partial}{\partial y}\right)^j = \frac{\partial^{i+j}}{\partial x^i \partial y^j}.$$

Therefore,

$$\begin{aligned} F''(0) &= \left[\left(\alpha \frac{\partial}{\partial x} + \beta \frac{\partial}{\partial y}\right)^2 f\right](x_0, y_0) \\ &= \left[\left((x - x_0)\frac{\partial}{\partial x} + (y - y_0)\frac{\partial}{\partial y}\right)^2 f\right](x_0, y_0). \end{aligned}$$

Then, by induction it follows that

$$\begin{aligned} F^{(k)}(0) &= \left[\left((x - x_0)\frac{\partial}{\partial x} + (y - y_0)\frac{\partial}{\partial y}\right)^k f\right](x_0, y_0) \\ F^{(n+1)}(\theta) &= \left[\left((x - x_0)\frac{\partial}{\partial x} + (y - y_0)\frac{\partial}{\partial y}\right)^{n+1} f\right](\xi, \eta), \end{aligned}$$

with $\xi = x_0 + \theta(x - x_0)$, and $\eta = y_0 + \theta(y - y_0)$ for some θ satisfying $0 \leq \theta \leq 1$. It is clear that ξ is between x_0 and x, and η is between y_0 and y. Combining these ideas, we obtain the following theorem.

THEOREM

1.9 (Taylor's Theorem in Two Dimensions)

Suppose $f(x, y)$ is $n+1$ times continuously differentiable for all (x, y) in some neighbourhood D of a point (x_0, y_0). Then, for any $(x, y) \in D$, we have

$$f(x, y) = P_n(x, y) + R_{n+1}(x, y),$$

where

$$P_n(x, y) = \sum_{k=0}^{n} \frac{1}{k!} \left[\left((x - x_0)\frac{\partial}{\partial x} + (y - y_0)\frac{\partial}{\partial y} \right)^k f \right](x_0, y_0) \tag{1.3}$$

$$R_{n+1}(x, y) = \frac{1}{(n+1)!} \left[\left((x - x_0)\frac{\partial}{\partial x} + (y - y_0)\frac{\partial}{\partial y} \right)^{n+1} f \right](\xi, \eta) \tag{1.4}$$

for some ξ between x_0 and x, and some η between y_0 and y. □

Note that the binomial theorem yields

$$\left((x - x_0)\frac{\partial}{\partial x} + (y - y_0)\frac{\partial}{\partial y} \right)^k = \sum_{j=0}^{k} \binom{k}{j} (x - x_0)^{k-j} (y - y_0)^j \frac{\partial^k}{\partial x^{k-j} \partial y^j}.$$

Thus we may rewrite (1.3) and (1.4) as

$$P_n(x, y) = \sum_{k=0}^{n} \frac{1}{k!} \sum_{j=0}^{k} \binom{k}{j} (x - x_0)^{k-j} (y - y_0)^j \frac{\partial^k f}{\partial x^{k-j} \partial y^j}(x_0, y_0), \tag{1.5a}$$

$$R_{n+1}(x, y) = \frac{1}{(n+1)!} \sum_{j=0}^{n+1} \binom{n+1}{j} (x - x_0)^{n+1-j} (y - y_0)^j \frac{\partial^{n+1} f}{\partial x^{n+1-j} \partial y^j}(\xi, \eta). \tag{1.5b}$$

EXAMPLE 1.10

Determine the Taylor polynomial of degree 2 for

$$f(x, y) = \cos \pi(x + y)$$

around $(0, 1)$. Determine the remainder term.

SOLUTION

From (1.5a) we have

$$\begin{aligned} P_2(x, y) = f(x_0, y_0) &+ (x - x_0)\frac{\partial f}{\partial x}(x_0, y_0) + (y - y_0)\frac{\partial f}{\partial y}(x_0, y_0) \\ &+ \frac{1}{2!}\left[(x - x_0)^2 \frac{\partial^2 f}{\partial x^2}(x_0, y_0) + 2(x - x_0)(y - y_0)\frac{\partial^2 f}{\partial x \partial y}(x_0, y_0) \right. \\ &\left. + (y - y_0)^2 \frac{\partial^2 f}{\partial y^2}(x_0, y_0) \right]. \end{aligned} \tag{1.6a}$$

With $f(x, y) = \cos \pi(x + y)$ and $(x_0, y_0) = (0, 1)$ we have $x - x_0 = x$, $y - y_0 = y - 1$, and

$$f(x_0, y_0) = -1, \qquad \frac{\partial^2 f}{\partial x^2}(x_0, y_0) = \pi^2,$$
$$\frac{\partial f}{\partial x}(x_0, y_0) = 0, \qquad \frac{\partial^2 f}{\partial x \partial y}(x_0, y_0) = \pi^2,$$
$$\frac{\partial f}{\partial y}(x_0, y_0) = 0, \qquad \frac{\partial^2 f}{\partial y^2}(x_0, y_0) = \pi^2.$$

Then, from (1.6a) it follows that

$$P_2(x, y) = -1 + \frac{\pi^2}{2}(x + y - 1)^2.$$

For the remainder term, we use (1.5b) to write

$$R_3(x, y) = \frac{1}{3!}\left[(x - x_0)^3 \frac{\partial^3 f}{\partial x^3}(\xi, \eta) + 3(x - x_0)^2(y - y_0)\frac{\partial^3 f}{\partial x^2 \partial y}(\xi, \eta)\right.$$
$$\left. + 3(x - x_0)(y - y_0)^2 \frac{\partial^3 f}{\partial x \partial y^2}(\xi, \eta) + (y - y_0)^3 \frac{\partial^3 f}{\partial y^3}(\xi, \eta)\right]. \quad \textbf{(1.6b)}$$

Note that we require all the third order partial derivatives to be evaluated at the point (ξ, η) for some ξ between x_0 and x and some η between y_0 and y. We also have $\xi = x_0 + \theta(x - x_0) = \theta x$, and $\eta = y_0 + \theta(y - y_0) = 1 + \theta(y - 1)$, with $0 \le \theta \le 1$. Note that each of the third order partial derivatives equals $\pi^3 \sin \pi(\xi + \eta) = \pi^3 \sin \pi(1 + \theta(x + y - 1))$. Then, from (1.6b) it follows that

$$R_3(x, y) = \frac{\pi^3}{3!}(x + y - 1)^3 \sin \pi(1 + \theta(x + y - 1)).$$

▲

EXERCISE SET 1.1.1

1. Show that the equation $x = 2^{-x}$ has a solution in the interval $[0, 1]$.
2. Show that the equation $\frac{3}{2}x^5 = 7^x$ has a solution in the interval $[2, 3]$.
3. Show that the function $f(x) = x^4 - 4x^2 - 20x$ has a relative minimum in the interval $[2, 3]$.
4. If $f(x) = (x - 2)\cos \frac{\pi x}{2}$, show that $f'(x) = 0$ for some $x \in [1, 2]$. **Do not** differentiate $f(x)$.
5. Show that $f'(x) = 0$ for some $x \in [0, 2]$, if $f(x) = 2x - 1 + 2\cos \frac{\pi x}{2}$. **Do not** differentiate $f(x)$.
6. Suppose $f(x)$ is a continuous function on the interval $[a, b]$, and f' exists on (a, b). Show that if $f'(x) \neq 0$ for all $x \in (a, b)$, then there can exist at most one point p such that $f(p) = 0$.
7. Show that the equation $x = 2^{-x}$ has exactly one real solution.

8. Show that the equation $2x^5 + 3x^3 + 2x + k = 0$ has exactly one real solution, regardless of the value of the constant k.
9. Show that there is a number $c \in [0, 4]$ such that: $f''(c) = 0$ for $f(x) = (x - 1)(x - 2)(x - 3)$. **Do not** differentiate $f(x)$.
10. Show that there is a number $c \in [-2, 3]$ such that: $f''(c) = 0$ for $f(x) = x \sin(x - 2) \ln(x + 3)$. **Do not** differentiate $f(x)$.
11. In Theorem 1.6 (Integral Mean-Value Theorem), let $w(x) = e^{-x}$, $f(x) = x^2$, and $[a, b] = [0, 1]$. Find the number c guaranteed by the theorem and verify that c lies in $(0, 1)$.
12. Suppose $g(x)$ is continuous on $[a, b]$. Then show that

$$\int_a^b (x - a)(b - x)^3 g(x)\, dx = \frac{(b - a)^5}{20} g(c) \qquad \text{for some } c \in [a, b].$$

13. In each of the following, determine whether the hypotheses of the Mean-Value Theorem for Derivatives (Theorem 1.7) are satisfied or not. If satisfied, determine the number c guaranteed by the theorem. If not, explain why.
    a. $f(x) = \sqrt{1 - x^2}$, $[a, b] = [-1, 1]$
    b. $f(x) = |x - 2|$, $[a, b] = [1, 3]$
    c. $f(x) = x^{3/2}$, $[a, b] = [-8, 8]$
    d. $f(x) = \dfrac{1}{x + 1}$, $[a, b] = [1, 2]$
14. Ineach of the following, determine whether the hypotheses of the Mean-Value Theorem for Derivatives (Theorem 1.7) are satisfied or not. If satisfied, determine the number c guaranteed by the theorem. If not, explain why.
    a. $f(x) = x^{4/5}$, $[a, b] = [-1, 32]$
    b. $f(x) = x^{4/5}$, $[a, b] = [0, 32]$
    c. $f(x) = |x - 2|$, $[a, b] = [2, 3]$
15. Use the Mean-Value Theorem for Derivatives to establish the following inequalities.
    a. $|\sin x - \sin y| \le |x - y|$
    b. $|e^x - e^y| \le |x - y|$ for all $x, y \le 0$
16. Use the Mean-Value Theorem for Derivatives to establish the following inequalities.
    a. $|x - y| \le |\tan x - \tan y|$ for $-\dfrac{\pi}{2} < x, y < \dfrac{\pi}{2}$
    b. $my^{m-1}(x - y) \le x^m - y^m \le mx^{m-1}(x - y)$ for $0 \le y \le x$, $m \ge 1$
17. Find the Taylor polynomial of degree four for $f(x) = e^x \sin x$ around $x_0 = 0$. Show the remainder term.
18. Find the Taylor polynomial of degree four for $f(x) = e^x \cos x$ around $x_0 = 0$. Show the remainder term.
19. Use the Taylor polynomials of degree three and four for $f(x) = \sqrt{1 + x}$ around $x_0 = 0$ to approximate $\sqrt{1.1}$. Obtain an error bound for your approximation in each case.
20. Use the Taylor polynomials of degree three and four for $f(x) = (1 + x)^{3/2}$ around $x_0 = 0$ to approximate $(1.1)^{3/2}$. Obtain an error bound for your approximation in each case.

21. Find the degree of the Taylor polynomial around $x_0 = 0$ that could be used to approximate e^x for all x in the interval [0, 1] to an accuracy of 5 decimal places.
22. Find the degree of the Taylor polynomial around $x_0 = 0$ that could be used to approximate e^{2x} for all x in the interval [0,1] to an accuracy of 5 decimal places.
23. Expand $f(x, y) = \cos x \sin y$ as a Taylor polynomial of degree two around $(x_0, y_0) = (0, \pi/2)$.
24. Make use of Taylor's theorem for functions of two variables to determine linear and quadratic approximations first to (a) $f(x, y) = (1 + x - y)^{1/3}$ and then to (b) $f(x, y) = \sqrt{(1 + 2x)/(1 + y)}$ for small values of x and y.

1.1.2 ROUND-OFF ERROR

In any computation, errors could result from several sources. The most common sources are human errors, or mistakes and blunders. Clearly, mistakes and blunders are outside the scope of numerical analysis! The major sources of errors of interest to numerical analysts are (i) uncertainty in data, (ii) round-off, and (iii) mathematical truncation.

Laboratory measurements using instruments with specified precision generally result in data containing uncertainty or experimental errors. The study of how such an initial uncertainty or error in the data propagates in the course of a computation is central to numerical analysis. Errors arising from the machine representation of real numbers and arithmetic performed on them, known as **round-off** errors, also interest the numerical analyst. The basic techniques used in the study of experimental errors and round-off errors are the same. However, the uncertainty in the data can be much larger than round-off in general. Mathematical truncation errors arise from the approximations applied in the numerical solution of a problem—such as replacing infinite processes with finite ones or replacing noncomputable problems with computable ones. These are the principal sources of errors of interest to numerical analysts in almost all classes of problems (with the possible exception of numerical linear algebra, where rounding errors are the major sources of errors). We will discuss truncation errors arising in the various numerical methods in the following chapters.

Humans perform calculations using the decimal number system. In the decimal number system, a positive number a is represented by

$$a = \sum_k \alpha_k 10^k,$$

where α_k are called the digits of a ($\alpha_k = 0, 1, 2, \cdots, 9$), and 10 is the **base** of the number system used. For example, the number written as 3465.37 represents

$$3 \cdot 10^3 + 4 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0 + 3 \cdot 10^{-1} + 7 \cdot 10^{-2}.$$

Most computers use the base-2 (or **binary**) number system, or a simple variant of it such as 8 or 16. When the base-2 number system is used, the digits are 0 and 1. For example, the number $(1101.0011)_2$ (the subscript 2 is used to denote that base-2 number system

is being used) may be determined as

$$(1101.0011)_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4}$$
$$= 13.1875.$$

Similarly, the digits in the base-16 number system are $0, 1, \cdots, 9, A, \cdots, F$ (where $A, \cdots, F$ are used to denote $10, \cdots, 15$). For example, the number $(23A.D)_{16}$ may be determined as

$$(23A.D)_{16} = 2 \cdot 16^2 + 3 \cdot 16^1 + 10 \cdot 16^0 + 13 \cdot 16^{-1} = 570.8125.$$

The representation of a real number could be a nonterminating sequence of digits, depending on the base used (for example, consider the number $\frac{1}{3}$ in the decimal or binary number system). However, since performing calculations on the computer calls for a uniform representation for numbers of greatly varying magnitude, there is usually a limit to the number of digits that may be used to represent a number. For representing integers, such a limit will affect only the range of integers representable on the computer. On the other hand, for real numbers it will also affect the precision of the representation. For example, note that a real number such as 0.1, which is exactly represented in the decimal number system, does not have an exact representation using a finite number of digits in the binary number system. This inherent limitation due to the **finite precision** in the representation then affects the accuracy of calculations as well. We will refer to arithmetic done on the computer as "finite-precision arithmetic." In this section, we will use β to denote the base of the arithmetic, and assume $\beta = 10$ in the numerical examples.

In the base-β number system, a real number x is represented in its **floating-point** form as

$$x = \pm(0.d_1 d_2 \cdots d_n)_\beta \beta^e,$$

where $(0.d_1 d_2 \cdots d_n)_\beta$ is a fraction in the base-β number system, called the **mantissa,** e is an integer called the **exponent,** and n is the number of digits carried in the representation of x. This form for $x \neq 0$ is said to be **normalized** if $d_1 \neq 0$ (if x were zero, all d_i will be zero). For example, the IBM 3000 series of computers use

$$\beta = 16 \qquad n = 6 \qquad -64 \leq e \leq 63,$$

and the CDC 6000 machines use

$$\beta = 2 \qquad n = 48 \qquad -975 \leq e \leq 1071.$$

Numerical analysis is concerned with the analysis of errors in numerical calculations. The following definition shows two commonly used error measures.

DEFINITION **1.1**

If x_A is an approximation to x, then

$$\textbf{\textit{absolute error}} \text{ in } x_A = Abs(x_A) = |x - x_A|,$$
$$\textbf{\textit{relative error}} \text{ in } x_A = Rel(x_A) = Abs(x_A)/|x|.$$

EXAMPLE 1.11

Calculate the absolute and relative errors given the following:

a. $x = 0.200 \times 10^1$, $x_A = 0.210 \times 10^1$. The absolute error is 0.1 and the relative error is 0.5×10^{-1}.

b. $x = 0.200 \times 10^{-4}$, $x_A = 0.210 \times 10^{-4}$. The absolute error is 0.1×10^{-5} and the relative error is 0.5×10^{-1}.

c. $x = 0.200 \times 10^3$, $x_A = 0.210 \times 10^3$. The absolute error is 0.1×10^2 and the relative error is 0.5×10^{-1}. ▒

As indicated by the preceding example, the absolute error depends on the size of x and may be misleading. Therefore, it may be more meaningful to measure errors relative to x.

If the mantissa of a given number contains more than n digits in its exact representation, then it must be shortened to n digits in some way so that it can be represented on the machine, thus limiting the precision in the representation. There are two commonly used ways of shortening a number whose mantissa is longer than n digits, called **chopping** and **rounding.** The error that results from chopping or rounding a given number to the precision of the computer is commonly referred to as **round-off error.**

EXAMPLE 1.12

Let $x = 475.846$. If x were to be represented by an approximation x_A using only 5 digits in the mantissa in base-10 arithmetic, then

$$x_A = 475.84 \text{ is obtained by chopping,}$$

$$x_A = 475.85 \text{ is obtained by rounding.}$$ ▒

In the following, we formally describe the procedures used for chopping and rounding for the decimal number system. Suppose the exact decimal ($\beta = 10$) representation of x has a normalized floating-point form

$$x = \pm 0.d_1 d_2 \cdots d_n d_{n+1} d_{n+2} \cdots \times 10^e.$$

By **chopping** x to n digits we mean replacing x with

$$x_A = \pm 0.d_1 d_2 \cdots d_n \times 10^e,$$

which amounts to simply throwing away all the digits after d_n in the exact representation of x. By **rounding** x to n digits we mean replacing x with

$$x_A = \begin{cases} \pm 0.d_1 d_2 \cdots d_n \times 10^e, & \text{if } 0 \le d_{n+1} < 5; \\ \pm (0.d_1 d_2 \cdots d_n + 0.00 \cdots 01) \times 10^e, & \text{if } 5 \le d_{n+1} < 10. \end{cases}$$

As a result, whenever $d_{n+1} \ge 5$, we will add one to d_n and *round up;* otherwise, we will simply chop off all but the first n digits and *truncate.*

For a computer using n-digit base-β arithmetic, if we let $fl(x)$ denote the floating-point representation of x whose exact representation has a normalized floating-point form

$$x = \pm(0.d_1d_2\cdots d_nd_{n+1}d_{n+2}\cdots)_\beta \times \beta^e,$$

then

$$fl(x) = \pm(0.d_1d_2\cdots d_n)_\beta \times \beta^e,$$

when chopping is used, and

$$fl(x) = \begin{cases} \pm(0.d_1d_2\cdots d_n)_\beta \times \beta^e, & \text{if } 0 \le d_{n+1} < \frac{\beta}{2}; \\ \pm[(0.d_1d_2\cdots d_n)_\beta + (0.00\cdots 01)_\beta] \times \beta^e, & \text{if } \frac{\beta}{2} \le d_{n+1} < \beta \end{cases}$$

when rounding is used. In any case, whether we chop or round, it is clear that a certain error is committed when x is replaced by $fl(x)$. The difference between x and $fl(x)$ is called the **round-off error** in x. The round-off error depends on the size of x and is therefore best measured relative to x. Suppose we write

$$fl(x) = x(1+\delta) \tag{1.7}$$

where $\delta = \delta(x)$ is some number which depends on x. Then we can bound δ independently of x. The notation of (1.7) is attributed to Wilkinson [19].

For chopping,

$$\begin{aligned} |x - fl(x)| &= (0.0\cdots 0d_{n+1}d_{n+2}\cdots)_\beta \times \beta^e \\ &= (0.d_{n+1}d_{n+2}\cdots)_\beta \times \beta^{e-n} \\ &\le \beta^{e-n}. \end{aligned}$$

Therefore the relative error in $fl(x)$ is

$$\begin{aligned} \frac{|x - fl(x)|}{|x|} &\le \frac{\beta^{e-n}}{(0.d_1d_2\cdots)_\beta \times \beta^e} \\ &\le \frac{\beta^{-n}}{(0.100\cdots)_\beta} = \frac{\beta^{-n}}{\beta^{-1}} \end{aligned}$$

from which we obtain

$$|\delta| = \frac{|x - fl(x)|}{|x|} \le \beta^{1-n}. \tag{1.8}$$

Similarly, for rounding, we proceed by cases in order to obtain the error in $fl(x)$. For the case corresponding to $0 \le d_{n+1} < \frac{\beta}{2}$ we obtain

$$\begin{aligned} |x - fl(x)| &= (0.0\cdots 0d_{n+1}d_{n+2}\cdots)_\beta \times \beta^e \\ &= (0.d_{n+1}d_{n+2}\cdots)_\beta \times \beta^{e-n} \\ &\le \frac{1}{2}\beta^{e-n}. \end{aligned}$$

Therefore the relative error in $fl(x)$ is

$$\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\frac{\beta^{e-n}}{(0.d_1d_2\cdots)_\beta \times \beta^e}$$
$$\leq \frac{1}{2}\frac{\beta^{-n}}{(0.100\cdots)_\beta} = \frac{1}{2}\frac{\beta^{-n}}{\beta^{-1}}$$

from which we obtain

$$|\delta| = \frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\beta^{1-n}. \tag{1.9a}$$

For the case corresponding to $\frac{\beta}{2} \leq d_{n+1} < \beta$ we have

$$|x - fl(x)| = (0.0\cdots 0d_{n+1}d_{n+2}\cdots)_\beta \times \beta^e - (0.00\cdots 01)_\beta \times \beta^e$$
$$= \left|(0.d_{n+1}d_{n+2}\cdots)_\beta \times \beta^{e-n} - (0.10\cdots)_\beta \times \beta^{e-n+1}\right|$$
$$= \left[(1.00\cdots)_\beta - (0.d_{n+1}d_{n+2}\cdots)_\beta\right] \times \beta^{e-n}$$
$$\leq \frac{1}{2}\beta^{e-n}.$$

Therefore the relative error in $fl(x)$ is

$$\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\frac{\beta^{e-n}}{(0.d_1d_2\cdots)_\beta \times \beta^e}$$
$$\leq \frac{1}{2}\frac{\beta^{-n}}{(0.100\cdots)_\beta} = \frac{1}{2}\frac{\beta^{-n}}{\beta^{-1}}$$

from which we obtain

$$|\delta| = \frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\beta^{1-n}. \tag{1.9b}$$

From (1.8), (1.9a), and (1.9b) we conclude that the relative round-off δ satisfies $|\delta| \leq \varepsilon$, where

$$\varepsilon = \begin{cases} \beta^{1-n}, & \text{for chopping;} \\ \frac{1}{2}\beta^{1-n}, & \text{for rounding.} \end{cases}$$

In the decimal number system, since $\beta = 10$, the maximum relative error in the representation of a real number using n digits is then 10^{1-n} when chopping is used and $0.5 \times 10^{1-n}$ when rounding is used. Therefore, the worst-case round-off effects due to rounding will be roughly half the worst-case round-off effects arising due to chopping. However, because of the additional work involved in the rounding process, many computers use chopping. It is clear that the quantity ε depends on n, and hence the machine being used. Therefore, it is frequently referred to as the **machine epsilon.** Formally

defined, the machine epsilon for any given computer is the smallest positive floating-point number ε for which $fl(1+\varepsilon) > 1$. For the IBM 3000 series machines mentioned earlier, $\varepsilon = 16^{-5}$, and for the CDC 6000 machines it is 2^{-47}.

Computers that accept Fortran programs are expected to provide two kinds of floating-point numbers. The first, called *single precision* numbers, have mantissas roughly half as long as those for the second kind, called *double precision* numbers. In any case, the exponent e is limited to a range $m \le e \le M$ for certain integers m and M. Whenever $0 < |x| < \beta^{m-1}$, the machine indicates an **underflow** condition and sets x to zero in most systems; whenever $|x| > \beta^M$, an **overflow** condition is indicated and on most systems the computation is halted. For example, on the IBM 3000 series computers, for all numbers x satisfying $0 < |x| < 16^{-65}$ an underflow condition is indicated, while for $|x| > 16^{63}$ an overflow condition is indicated. In the discussions to follow, we will assume that the numbers we are dealing with do not cause an underflow or an overflow condition.

Finally, the concept of **significant digits** is often used in place of the relative error. An approximation x_A is said to have m significant digits with respect to the exact value x if m is the largest nonnegative integer for which

$$\frac{|x - x_A|}{|x|} \le \frac{1}{2}\beta^{1-m}.$$

In the decimal number system, we say a number has m significant digits if m is the largest nonnegative integer for which the relative error in the number is less than 5×10^{-m}.

EXAMPLE 1.13

Consider $x = 0.02136$ and $x_A = 0.02147$ ($\beta = 10$). Then

$$\frac{|x - x_A|}{|x|} = \frac{0.00011}{0.02136} \approx 0.00515 \le \frac{1}{2}10^{1-2} = 5 \times 10^{-2}.$$

Therefore, x_A has two significant digits with respect to x. ▩

EXAMPLE 1.14

Consider $x = 25.486$ and $x_A = 25.484$ ($\beta = 10$). Then

$$\frac{|x - x_A|}{|x|} = \frac{0.002}{25.486} \approx 7.8 \times 10^{-5} \le 5 \times 10^{-4}.$$

Therefore, x_A has four significant digits with respect to x. ▩

For more extensive treatment of computer arithmetic, see standard textbooks on computer architecture such as Mano [17] and Hwang [13].

While the finite-precision representation of floating-point numbers results in an inherent round-off error, performing arithmetic on the computer results in the propagation of the round-off. This is mainly because, in general, the result of an arithmetic operation performed on two floating-point numbers of the same length fails to be

a floating-point number of the same length. For example, if $x = 3 = (0.300) \times 10^1$, $y = (0.852) \times 10^{-6}$, and $z = 4 = (0.400) \times 10^1$, then

$$x + y = (0.3000000852) \times 10^1 \quad \text{and} \quad \frac{z}{x} = (0.133\cdots) \times 10^1.$$

Therefore, an error introduced at any step during a computation may be amplified or reduced in subsequent operations. This is called the **propagation** of errors.

Let x and y be the exact values of two numbers with corresponding machine representations x_A and y_A. Let $*$ denote one of the operations $+$, $-$, $\times$, or $/$. Then, $fl(x_A * y_A)$ will be the actual result obtained, whereas the exact value must be $x * y$, since the machine version of an arithmetic operation usually includes rounding or chopping. Hence the error in the computation is

$$x * y - fl(x_A * y_A) = (x * y - x_A * y_A) + (x_A * y_A - fl(x_A * y_A)). \tag{1.10}$$

The first term in parentheses on the right side of (1.10) is called the **propagated error,** and the second term is the round-off error. The round-off error is easily bounded by using (1.8) for chopping, or (1.9a) and (1.9b) for rounding.

Different approaches have been explored in order to estimate propagated errors in computations. The first computational method for estimating propagated errors is known as *interval arithmetic.* In interval arithmetic, each number is represented by a pair of machine numbers, a lower bound and an upper bound. The result of every basic operation is then realized as an interval. The following example will illustrate the use of interval arithmetic in obtaining propagated errors.

EXAMPLE 1.15

Suppose $x_A = 2.34$ and $y_A = 1.71$ approximate x and y, respectively, using 3-digit decimal rounding arithmetic. Then, for the product xy, we obtain the approximation $fl(x_A y_A) = fl(2.34 \times 1.71) = fl(4.0014) = 4.00$. Hence, the round-off error in the last step is 0.0014. To obtain the propagated error, we proceed as follows. We have

$$|x - 2.34| \leq 0.005 \quad \text{and} \quad |y - 1.71| \leq 0.005,$$

or, equivalently,

$$2.335 \leq x \leq 2.345 \quad \text{and} \quad 1.705 \leq y \leq 1.715.$$

Therefore,

$$3.981175 \leq x \times y \leq 4.021675.$$

For the propagated error we have

$$-0.020225 \leq xy - x_A y_A \leq 0.020275.$$

■

The main objections to using interval arithmetic are that it requires too much computational time, and that the error bounds obtained may be too exaggerated.

In order to determine the effect of an individual error on the final answer in a computation, we may proceed as follows: Suppose x_A is the approximate value used

in place of the exact number x, and the final answer depends on x as $f(x)$. If $f(x)$ is differentiable, the Mean-Value Theorem for Derivatives (Theorem 1.7) yields

$$f(x) = f(x_A) + f'(\xi)(x - x_A) \tag{1.11}$$

for some ξ between x and x_A. Since x and x_A are usually very close, from (1.11) we obtain

$$\text{Abs}(f(x_A)) = |f(x) - f(x_A)| = |f'(\xi)| \cdot |x - x_A| \approx |f'(x_A)| \cdot \text{Abs}(x_A). \tag{1.12}$$

For example, consider $f(x) = \sqrt{x}$. Then $f'(x) = 1/2\sqrt{x}$. Therefore, from (1.12) we obtain $\text{Abs}(\sqrt{x_A}) \approx \text{Abs}(x_A)/2\sqrt{x_A}$. For the relative error we have

$$\text{Rel}(\sqrt{x_A}) = \left|\frac{\text{Abs}(\sqrt{x_A})}{\sqrt{x_A}}\right| \approx \frac{\text{Abs}(x_A)}{2\sqrt{x_A}} \cdot \frac{1}{\sqrt{x_A}} = \frac{1}{2} \cdot \text{Rel}(x_A).$$

In other words, the relative error in $\sqrt{x_A}$ is about half the relative error in x_A irrespective of the size of x. Therefore, taking the square root may be considered a safe operation, since it reduces the relative error in the argument.

In a similar manner, the error propagation in a computation depending on two variables can be studied. For example, suppose x_A and y_A are the approximate values used in place of the exact numbers x and y, and the final answer depends on x and y as $f(x, y)$. If $f(x, y)$ is differentiable with respect to x and y, Theorem 1.9 may be used to obtain

$$f(x, y) \approx f(x_A, y_A) + f_x(x_A, y_A)(x - x_A) + f_y(x_A, y_A)(y - y_A)$$

where $f_x = \partial f/\partial x$ and $f_y = \partial f/\partial y$, respectively. Therefore,

$$\text{Abs}(f(x_A, y_A)) \approx |f_x(x_A, y_A)| \cdot \text{Abs}(x_A) + |f_y(x_A, y_A)| \cdot \text{Abs}(y_A). \tag{1.13}$$

Let's consider an example.

EXAMPLE 1.16

The formula for the net capacitance when two capacitors of values x and y are connected in series (see Fig. 1.7) is

$$z = \frac{xy}{x + y}.$$

Suppose the measured values of x and y are $x_A = 2.71\ \mu\text{F}^*$ and $y_A = 3.14\ \mu\text{F}$, rounded to three digits. Determine the propagated error in the calculation

$$z_A = \frac{x_A y_A}{x_A + y_A}.$$

* Capacitances are measured in farads, and μF stands for microfarads.

Figure 1.7

SOLUTION Let $z = f(x, y) = xy/(x + y)$. Then, differentiating partially with respect to x and y respectively yields

$$f_x(x, y) = \frac{y^2}{(x + y)^2},$$

$$f_y(x, y) = \frac{x^2}{(x + y)^2},$$

which result in $f_x(x_A, y_A) \approx 0.288103$ and $f_y(x_A, y_A) \approx 0.214599$. Therefore, by (1.13) it follows that

$$\text{Abs}(z_A) \approx (0.288103)\text{Abs}(x_A) + (0.214599)\text{Abs}(y_A).$$

With $\text{Abs}(x_A) = \text{Abs}(y_A) = 0.005$, we have $\text{Abs}(z_A) \approx 0.00251351$. ▲

EXERCISE SET 1.1.2

1. Convert the following numbers to their decimal equivalents.
   a. $(101101.101)_2$
   b. $(2AB.EF)_{16}$
   c. $(2057.34)_8$
   d. $(.1010101\cdots)_2$
   e. $(10101\cdots01)_2$ with the parentheses enclosing k digits.
2. Convert the following numbers to their decimal equivalents.
   a. $(.10101\cdots01)_2$ with the parentheses enclosing k digits
   b. $(.1001100110011\cdots)_2$
   c. $(.F0F0F\cdots)_{16}$
3. Calculate the absolute error and the relative error given that
   a. $x = \dfrac{1}{6}, \quad x_A = 0.1667$
   b. $x = \dfrac{100}{6}, \quad x_A = 16.67$
4. Calculate the absolute error and the relative error given that

a. $x = \dfrac{1}{7}, \quad x_A = 0.1429$

b. $x = \dfrac{100}{7}, \quad x_A = 14.29$

5. For the following numbers x and x_A, determine how many significant digits there are in x_A with respect to x.
   a. $x = 257.03, \quad x_A = 257.028$
   b. $x = 0.025703, \quad x_A = 0.025713$
   c. $x = 34.7186, \quad x_A = 34.7286$
6. For the following numbers x and x_A, determine how many significant digits there are in x_A with respect to x.
   a. $x = 457.0271, \quad x_A = 457.03$
   b. $x = 0.0457027, \quad x_A = 0.0457017$
   c. $x = 54.4126, \quad x_A = 54.4016$
7. Compute using three-digit decimal arithmetic with rounding.
   a. $16.3 + 0.0893$
   b. $(173. + 0.753) - (158. + 15.0)$
   c. $0.0182 \times 197.$
8. Compute using three-digit decimal arithmetic with rounding. Estimate the propagated errors in each case.
   a. $\dfrac{2}{5} + \dfrac{4}{3}$
   b. $\left(\dfrac{1}{3} - \dfrac{3}{11}\right) + \dfrac{3}{20}$
   c. $\dfrac{2}{5} \times \dfrac{4}{3}$
9. The numbers given below are correctly rounded to the number of digits shown. For each computation, determine the smallest interval in which the true result must lie.
   a. $1.0053 + 0.357$
   b. $45.78 - 11.673$
   c. $(2.717) \times (3.843)$
   d. $7.143/1.414$
10. The numbers given below are correctly rounded to the number of digits shown. For each computation, determine the smallest interval in which the true result must lie.
    a. $2.1057 + 0.0313$
    b. $4.572 - 11.673$
    c. $(2.609) \times (1.213)$
    d. $1.732/2.236$
11. Let $f(x) = b^x$ for some positive constant b. Estimate the propagated error $|b^x - b^{x_A}|$ using Equation (1.12).
12. Let $f(x) = \sin kx$ for some constant k. Estimate the propagated error $|\sin kx - \sin kx_A|$ using Equation (1.12).
13. Consider the following function evaluations in which the arguments are correctly rounded to the number of digits shown. Estimate the propagated error and the corresponding relative error in each computation.

a. $\sin(2.413)$
b. $\ln(2.4134)$

14. Consider the following function evaluations in which the arguments are correctly rounded to the number of digits shown. Estimate the propagated error and the corresponding relative error in each computation.
a. $\sqrt{0.1224}$
b. $e^{1.437}$

15. The ideal gas law is stated as $PV = nRT$, in which R is a constant for all gases. The value R is known with some uncertainty described by

$$R = 8.3135 + \varepsilon, \qquad |\varepsilon| \leq 0.002.$$

Assuming $P = V = n = 1$, determine the uncertainty in the value of T calculated by using $PV = nRT$, resulting from the uncertainty in R.

16. The sides of a right triangle are x and y, and the hypotenuse is z. If $x_A = 3.15$ and $y_A = 2.78$ are the approximate values of x and y rounded to three digits, determine the absolute and the relative errors in the length z_A of the hypotenuse calculated by using $z_A = \sqrt{x_A^2 + y_A^2}$.

1.2 Practice

In this section, we will discuss the practical issues of vital importance to all numerical computing. The notion of **convergence** is very important in analysis. Basic concepts of the derivative, integral, and continuity are defined in terms of convergent sequences, and elementary functions are defined by convergent series. It turns out that convergence is an essential concept in numerical analysis as well. For example, consider solving numerically the rootfinding problem $f(x) = 0$. Let the desired root be α. A numerical method usually produces a sequence $\alpha_1, \alpha_2, \cdots$ of numbers converging to α. The process of obtaining each member of this sequence is called **iteration,** and the members α_k themselves are called **iterates.** In general, any numerical method that produces a sequence of iterates $\alpha_1, \alpha_2, \cdots$ may be used to determine α to any desired accuracy merely by calculating α_n for a *large enough* n. However, it is often very difficult to estimate n, the number of iterations required to achieve a prescribed accuracy, since the desired solution α is usually unknown for real-life problems. The larger the n, the more accurate we expect the computed results to be. On the other hand, the computed results may be affected adversely by the propagation of errors as the number of iterations increases. An important concept that is often used to describe how errors propagate is called **stability.** In this section, we will discuss the formal notions of stability and convergence as related to numerical methods.

1.2.1 STABILITY AND CONVERGENCE

We begin by considering the basic arithmetic operations. Errors may be magnified due to a single arithmetic operation. Note that (1.13) may be used to determine the propagated

error in arithmetic operations by setting $f(x, y) = x \pm y$, xy, and x/y. According to (1.13),

$$\text{Abs}(f(x_A, y_A)) \approx |f_x(x_A, y_A)|\text{Abs}(x_A) + |f_y(x_A, y_A)|\text{Abs}(y_A).$$

Therefore, since $f_x(x, y) = f_y(x, y) \equiv 1$ for $f(x, y) = x + y$, we obtain

$$\text{Abs}(x_A + y_A) \approx \text{Abs}(x_A) + \text{Abs}(y_A).$$

In a similar manner, the other operations may be analyzed. We obtain

$$\text{Abs}(x_A \pm y_A) \approx \text{Abs}(x_A) + \text{Abs}(y_A), \tag{1.14a}$$

$$\text{Abs}(x_A y_A) \approx |y|\text{Abs}(x_A) + |x|\text{Abs}(y_A), \tag{1.14b}$$

$$\text{Abs}(x_A/y_A) \approx |1/y|\text{Abs}(x_A) + |x/y^2|\text{Abs}(y_A). \tag{1.14c}$$

For the relative errors we have

$$\text{Rel}(x_A \pm y_A) \approx \frac{\text{Abs}(x_A) + \text{Abs}(y_A)}{|x \pm y|}, \tag{1.15a}$$

$$\text{Rel}(x_A y_A) \approx \text{Rel}(x_A) + \text{Rel}(y_A), \tag{1.15b}$$

$$\text{Rel}(x_A/y_A) \approx \text{Rel}(x_A) + \text{Rel}(y_A). \tag{1.15c}$$

It is evident from (1.14b) that the absolute error in the product $x_A y_A$ could be much larger than the absolute error in x_A or y_A, especially when either x or y is fairly large in size. However, as seen from (1.15b), the relative error in the product is at worst equal to the sum of the relative errors in x_A and y_A.

Similarly, from (1.14c) it follows that the absolute error in the quotient x_A/y_A could be much larger than the absolute error in x_A or y_A, especially when y is fairly small in size, while, as seen from (1.15c), the relative error in the quotient is at worst equal to the sum of the relative errors in x_A and y_A.

Finally, it is evident from (1.14a) that the absolute error in $x_A \pm y_A$ is at most equal to the sum of the absolute errors in x_A and y_A, while, as seen from (1.15a), the relative error in $x_A \pm y_A$ may be much larger than the relative error in x_A or y_A. For example, suppose we wish to calculate the number $z = x - y$ and we have only approximations x_A and y_A to x and y which are good to, say, k digits. Then $z_A = x_A - y_A$ will also be an approximation to z which is good to k digits as long as x and y do not agree to one or more digits. When x and y agree to one or more digits, there will be cancellation of digits in the subtraction, resulting in fewer good digits in z_A. The following numerical example will illustrate this fact.

EXAMPLE 1.17

Let $x_A = (0.34523412) \times 10^2$ and $y_A = (0.34522301) \times 10^2$ be approximations to x and y correct to seven significant digits. Then, in eight-digit decimal floating-point arithmetic,

$$z_A = x_A - y_A = (0.1111000) \times 10^{-2}$$

is the exact difference between x_A and y_A. However, z_A is good only to three significant

digits as an approximation to z, since the fourth digit was obtained from the eighth digits of x and y, which were in error. In other words, even though the absolute error in z is at most the sum of the absolute errors in x and y, the relative error in z_A is about 10,000 times the relative error in x_A or y_A. This phenomenon is referred to as **loss of significance** or **catastrophic cancellation.** ▒

In summary, the preceding discussion indicates that for both multiplication and division, relative errors do not propagate rapidly; however, the absolute errors may propagate rapidly when we multiply by a very large number or divide by a very small number. On the other hand, for addition and subtraction, the absolute errors do not propagate rapidly but the relative errors may propagate, especially when we compute the difference between nearly equal quantities, resulting in the loss of several significant digits. Therefore, it is necessary to be watchful of such situations in large calculations since loss of significance can give rise to gross inaccuracy. In some situations, it is possible to avoid the loss of significant digits. We illustrate this by the following examples.

EXAMPLE 1.18

Consider evaluating $\dfrac{\sqrt{1+x}-1}{x}$ for $x = 0.0001$ using finite-precision decimal arithmetic with $n = 4, 5, 6, 7,$ or 8 digits in the mantissa and chopping. With four or five digits in the mantissa, we obtain a value zero for the given expression. With six digits and chopping, we obtain a value 0.4; with seven and eight digits, we obtain 0.49 and 0.499, respectively. As the number of digits carried in the arithmetic is increased, the computed result seems to approach 0.5. Then what is going wrong when fewer digits are carried in the arithmetic? Since $\sqrt{1+x}$ is very close to 1 for small values of x, the desired evaluation will involve differencing nearly equal quantities. Consequently, the results obtained will not be all that good when very few digits are carried in the computation.

However, it is possible to obtain reasonably accurate results if the expression to be evaluated is rewritten in a form that does not involve differencing nearly equal quantities. For example, rationalizing the numerator of the expression given, we obtain

$$\frac{\sqrt{1+x}-1}{x} = \frac{\sqrt{1+x}-1}{x} \cdot \frac{\sqrt{1+x}+1}{\sqrt{1+x}+1} = \frac{1}{1+\sqrt{1+x}}.$$

From this expression, we obtain the answer 0.5 using only four digits in the mantissa and chopping. With five, six, seven, and eight digits, we obtain the answers 0.5, 0.49999, 0.4999877, and 0.49998752, respectively. ▒

Thus we have avoided the loss of significant digits in the given expression by rewriting it in a form that does not involve differencing nearly equal quantities. Quite often, Taylor's series expansions are very useful in this regard. For example, consider evaluating the function $f(x) = 1 - \cos x$ for very small values of x. Multiplying and

dividing $f(x)$ by $1 + \cos x$ yields

$$f(x) = (1 - \cos x) \cdot \frac{1 + \cos x}{1 + \cos x} = \frac{1 - \cos^2 x}{1 + \cos x} = \frac{\sin^2 x}{1 + \cos x}.$$

The last expression does not involve differencing nearly equal quantities. We could have also avoided the differencing of nearly equal quantities in the original expression by using the Taylor's series expansion for $\cos x$ around 0. Then, $f(x)$ becomes

$$f(x) = 1 - \left(1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots\right)$$

$$= \frac{x^2}{2!} - \frac{x^4}{4!} + \cdots.$$

EXAMPLE 1.19

Evaluate $\dfrac{5^x - 3^x}{x}$ for $x = 10^{-4}$ using six-digit decimal arithmetic and chopping.

SOLUTION

The answer is 0.6 if we evaluate directly using the given expression. However, since 5^x and 3^x are both very close to 1 for small values of x, we will be differencing nearly equal quantities. We may rewrite 5^x and 3^x as $e^{(\ln 5)x}$ and $e^{(\ln 3)x}$, respectively, and use Taylor's series expansion for $e^{\alpha x}$ to rewrite the two exponentials. Using

$$e^{\alpha x} = 1 + \alpha x + \frac{\alpha^2 x^2}{2!} + \cdots$$

we obtain

$$5^x = e^{(\ln 5)x} = 1 + (\ln 5)x + \frac{(\ln 5)^2 x^2}{2!} + \cdots,$$

and

$$3^x = e^{(\ln 3)x} = 1 + (\ln 3)x + \frac{(\ln 3)^2 x^2}{2!} + \cdots.$$

Therefore,

$$5^x - 3^x = [(\ln 5) - (\ln 3)]x + \left[(\ln 5)^2 - (\ln 3)^2\right]\frac{x^2}{2!} + \cdots,$$

and

$$\frac{5^x - 3^x}{x} = \ln(5/3) + \left[(\ln 5)^2 - (\ln 3)^2\right]\frac{x}{2!} + \text{ higher order terms.}$$

Using only the first term on the right side of the last expression yields a value of $0.5108\cdots$ for the given expression. Alternately, we could increase the number of digits carried in the arithmetic and evaluate the expression directly. For example, using eight digits, we obtain a value of 0.511 for the given expression by direct evaluation. ▲

The familiar quadratic formula for the solution of the equation $ax^2 + bx + c = 0$ provides another example where loss of significant digits could occur. The roots of the quadratic equation $ax^2 + bx + c = 0$ are given by the formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Assume that $b^2 - 4ac > 0$ and $b > 0$, and we wish to obtain the root of smaller magnitude. That is, we wish to calculate

$$x^{(1)} = \frac{-b + \sqrt{b^2 - 4ac}}{2a}. \tag{1.16}$$

Note that b and $\sqrt{b^2 - 4ac}$ will agree to several places whenever $4ac$ is much smaller than b^2. Hence, if finite-precision arithmetic is used, the root $x^{(1)}$ will be obtained with fewer correct digits than were used in the calculation. Now, by rationalizing the numerator in (1.16) we obtain

$$\begin{aligned} x^{(1)} &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \cdot \frac{(-b - \sqrt{b^2 - 4ac})}{(-b - \sqrt{b^2 - 4ac})} \\ &= \frac{b^2 - (b^2 - 4ac)}{-2a(b + \sqrt{b^2 - 4ac})}. \end{aligned}$$

Further simplification yields

$$x^{(1)} = \frac{-2c}{b + \sqrt{b^2 - 4ac}}. \tag{1.17a}$$

Similarly, if $b < 0$, the root

$$x^{(2)} = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

will suffer from the loss of significant digits in its computation. Once again, rationalization yields

$$\begin{aligned} x^{(2)} &= \frac{-b - \sqrt{b^2 - 4ac}}{2a} \cdot \frac{(-b + \sqrt{b^2 - 4ac})}{(-b + \sqrt{b^2 - 4ac})} \\ &= \frac{b^2 - (b^2 - 4ac)}{-2a(b - \sqrt{b^2 - 4ac})}. \end{aligned}$$

Further simplification yields

$$x^{(2)} = \frac{-2c}{b - \sqrt{b^2 - 4ac}}. \tag{1.17b}$$

Note that there will be no differencing nearly equal quantities in the computations corresponding to (1.17a) and (1.17b), and thus the loss of significant digits will be avoided. Combining these general ideas for both the roots of the equation corresponding

to $b > 0$ and $b < 0$, we may compute the roots using the formulas

$$\frac{-2c}{b + \text{sign}(b)\sqrt{b^2 - 4ac}} \text{ and } \frac{-b - \text{sign}(b)\sqrt{b^2 - 4ac}}{2a},$$

where

$$\text{sign}(b) = \begin{cases} 1, & \text{if } b > 0; \\ -1, & \text{if } b < 0. \end{cases}$$

EXAMPLE 1.20

Consider solving $x^2 + 72x + 1 = 0$, whose exact roots are

$$x^{(1)} = \frac{-72 + \sqrt{72^2 - 4}}{2} \text{ and } x^{(2)} = \frac{-72 - \sqrt{72^2 - 4}}{2}.$$

On simplifying, we get

$$x^{(1)} = -36 + \sqrt{1295} \text{ and } x^{(2)} = -36 - \sqrt{1295}.$$

Using six-digit decimal arithmetic, $\sqrt{1295} \approx 35.9861$, so that

$$|\sqrt{1295} - 35.9861| \leq 0.00005.$$

Using $\sqrt{1295} = 35.9861$, we now obtain the roots

$$x_A^{(1)} = -0.0139 \text{ and } x_A^{(2)} = -71.9861.$$

Therefore,

$$|\text{Abs}(x_A^{(1)})| \leq 0.00005,$$

$$|\text{Abs}(x_A^{(2)})| \leq 0.00005,$$

$$|\text{Rel}(x_A^{(1)})| \leq \frac{0.00005}{0.0139} \approx 3.6 \times 10^{-3}, \text{ and}$$

$$|\text{Rel}(x_A^{(2)})| \leq \frac{0.00005}{71.9861} \approx 6.95 \times 10^{-7}.$$

In other words, even though the number used for $\sqrt{1295}$ is good to six digits, the relative error in $x_A^{(1)}$ is very high, indicating fewer correct digits. Note that $x_A^{(2)}$ involves only the addition of nearly equal quantities and hence does not give rise to any difficulties in its computation. We could improve the answer for $x_A^{(1)}$ by using (1.17a) and obtain

$$x_A^{(1)} = \frac{-2}{72 + \sqrt{72^2 - 4}} = \frac{-1}{36 + \sqrt{1295}} = -1/71.9861 = -0.0138916$$

rounded to six digits. For the error in the new $x_A^{(1)}$ we have

$$|\text{Abs}(x_A^{(1)})| \leq \left| x^{(1)} - \frac{-1}{71.9861} \right| + \left| \frac{-1}{71.9861} - (-0.0138916) \right|,$$

$$\leq \left| \text{Abs}\left(\frac{1}{71.9861} \right) \right| + 5 \times 10^{-8},$$

and

$$|\text{Rel}(x_A^{(1)})| \leq \left|\text{Rel}\left(\frac{1}{71.9861}\right)\right| + \frac{5 \times 10^{-8}}{0.0138916},$$
$$\leq 6.95 \times 10^{-7} + 3.60 \times 10^{-6},$$
$$\leq 4.3 \times 10^{-6},$$

which is of the order of the relative error in $x_A^{(2)}$. ▩

Another situation in which loss of significance errors occurs is the addition of many numbers of large magnitude and varying sign to obtain a result that is much smaller. An example illustrating this will be given in the next subsection.

All our examples indicate that accurate results can be obtained by carrying enough digits in the arithmetic. However, it is not easy to know in advance how many digits need to be carried in the arithmetic in order to obtain reasonably accurate results. It is generally hoped that the precision built into the computer system is good enough so that there is no need to worry about rounding errors. But, since such a hope is neither justified nor fulfilled by any computer system, it is necessary to perform some mathematical analysis of the computational scheme used. With such an analysis, it may become much easier to assert that, within some reasonable limits, the numerical results obtained are indeed the results that were originally sought.

Next, we consider the propagation of errors as the number of arithmetic operations increases. Errors propagate in different ways. Some errors may decay and may not affect the accuracy much. Other errors may grow to an unacceptable extent and invalidate the computations completely. Suppose we denote the growth of an initial error ε after n steps by $E_n(\varepsilon)$. If

$$|E_n(\varepsilon)| \approx Cn\varepsilon$$

for some constant C independent of n, the error growth is said to be **linear.** However, if $E_n(\varepsilon)$ behaves like

$$|E_n(\varepsilon)| \approx k^n\varepsilon$$

for some constant $k > 1$, we say that the error growth is **exponential.**

Linear error growth is acceptable in most situations, and is usually not dangerous. In contrast, exponential error growth is dangerous and should be avoided whenever possible. Accordingly, a numerical method that exhibits a linear error growth is said to be **stable,** and a method that exhibits an exponential error growth is said to be **unstable** (see Fig. 1.8).

EXAMPLE 1.21

Consider generating the sequence $p_n = (1/3)^n$, $n > 0$ recursively by either

$$p_n = \frac{1}{3}p_{n-1}, \qquad n \geq 1, \qquad p_0 = 1, \tag{1.18}$$

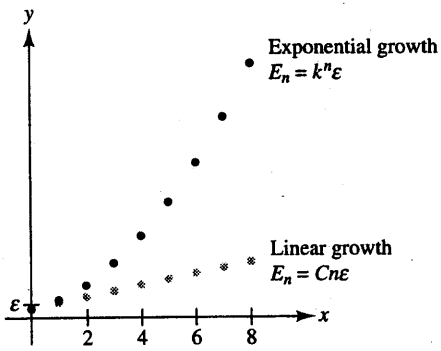


Figure 1.8 Linear and exponential error growth.

or

$$p_n = 4p_{n-1} - \frac{11}{9}p_{n-2}, \qquad n \geq 2, \qquad p_0 = 1,\ p_1 = \frac{1}{3}. \tag{1.19}$$

If six-digit floating-point decimal arithmetic with rounding is used, rounding error introduced by replacing 1/3 with 0.333333 will result in an error of only around $(0.333333)^n \times 10^{-6}$ in the n^{th} term of the sequence when (1.18) is used. Since this error decays as n increases, we may conclude that (1.18) is stable.

Next, let us consider the relation (1.19). A recursive relation of the form (1.19) is often called a **recurrence relation** or a **difference equation.** The difference equation may be used to obtain a general description of p_n in terms of n. This relationship is called the solution of the difference equation. For a general discussion on solving difference equations, refer to Liu [16]. For the present example, in order to see what p_n the formula (1.19) generates, we simply substitute $p_n = \alpha^n$ in (1.19) and obtain the equation

$$\alpha^2 - 4\alpha + \frac{11}{9} = \left(\alpha - \frac{1}{3}\right)\left(\alpha - \frac{11}{3}\right) = 0,$$

from which we conclude that the recurrence relation (1.19) generates $p_n = \alpha^n$ for $\alpha = 1/3$ or $\alpha = 11/3$. A general solution to (1.19) is given by

$$p_n = C_1\left(\frac{1}{3}\right)^n + C_2\left(\frac{11}{3}\right)^n,$$

where C_1 and C_2 are constants determined by the values of p_0 and p_1. For the case we are interested in, where $p_0 = 1$ and $p_1 = 1/3$, we must have $C_1 = 1$ and $C_2 = 0$. However, when six-digit floating-point decimal arithmetic with rounding is used, $p_0 = 1$ and $p_1 = 0.333333$. These conditions yield $C_1 = 0.100000 \times 10^1$ and $C_2 = -0.100000 \times 10^{-6}$. Since C_2 is not exactly zero, however small it may be, its contribution to p_n will result in an error of the form $C_2(11/3)^n$. Thus the error growth in the scheme (1.19) is exponential, indicating an unstable computation. This is confirmed by the computed

Table 1.1

n	exact	computed (1.19)
3	0.370370×10^{-1}	0.370340×10^{-1}
4	0.123457×10^{-1}	0.123350×10^{-1}
5	0.411523×10^{-2}	0.407630×10^{-2}
6	0.137174×10^{-2}	0.122910×10^{-2}
7	0.457247×10^{-3}	-0.657400×10^{-4}
8	0.152415×10^{-3}	-0.176519×10^{-2}

values shown in Table 1.1, which shows that the computed value of p_n becomes negative for $n \geq 7$ while the exact value is always positive. ▩

As we mentioned earlier, a numerical method often produces a sequence of iterates converging to the desired answer. When several methods are available for solving a given problem, we usually choose a method that converges the "fastest." The following definition is useful for comparing the convergence rates of sequences.

DEFINITION **1.2**

Suppose the sequence $\{\alpha_n\}_{n=1}^{\infty}$ *converges to a number* α. *We say that the* ***rate of convergence*** *is* $O(\beta_n)$ *(read "big oh of* β_n*"), or* at least of order β_n, *provided* $\beta_1, \beta_2, \cdots$ *is a sequence such that, for some constant* K *independent of* n,

$$\frac{|\alpha - \alpha_n|}{|\beta_n|} \leq K \quad \textit{for sufficiently large } n,$$

and we write $\alpha_n = \alpha + O(\beta_n)$.

EXAMPLE 1.22

Consider the sequences $\alpha_n = (n+2)/n^2$ and $\hat{\alpha}_n = (n+5)/n^3$. It is clear that $\lim_{n\to\infty} \alpha_n = 0$ and $\lim_{n\to\infty} \hat{\alpha}_n = 0$.

However, as seen in Table 1.2, the sequence $\hat{\alpha}_n$ converges to zero faster than the sequence α_n. For, if we let $\beta_n = 1/n$ and $\hat{\beta}_n = 1/n^2$, we have

$$\left|\frac{\alpha_n - 0}{\beta_n}\right| = \left|\frac{(n+2)/n^2 - 0}{1/n}\right| = \frac{n+2}{n} \leq 3,$$

$$\left|\frac{\hat{\alpha}_n - 0}{\hat{\beta}_n}\right| = \left|\frac{(n+5)/n^3 - 0}{1/n^2}\right| = \frac{n+5}{n} \leq 6,$$

indicating that $(n+2)/n^2 = 0 + O(1/n)$ and $(n+5)/n^3 = 0 + O(1/n^2)$. Since $1/n^2$ approaches zero faster than $1/n$, we may conclude that $\hat{\alpha}_n$ converges faster than α_n. ▩

Table 1.2

n	α_n	$\hat{\alpha}_n$
1	3.000000	6.000000
2	1.000000	0.875000
3	0.555556	0.296296
4	0.375000	0.140625
5	0.280000	0.080000
6	0.222222	0.050926
7	0.183673	0.034985
8	0.156250	0.025391

The concept of convergence rates for sequences may be generalized for functions in the following manner. This generalization is often convenient, since numerical methods may be studied in terms of an index n as $n \to \infty$, or a continuous parameter $h = 1/n$ as $h \to 0$.

DEFINITION **1.3**

Suppose $\lim_{x \to 0} F(x) = L$. *Then the* ***rate of convergence*** *of* $F(x)$ *to* L *is said to be* $O(G(x))$, *or* at least of order $G(x)$, *provided* $G(x)$ *is a function such that for some constant* $K > 0$ *independent of* x,

$$\frac{|F(x) - L|}{|G(x)|} \leq K \quad \textit{for sufficiently small } x > 0,$$

and we write $F(x) = L + O(G(x))$.

EXAMPLE 1.23

Determine the rate of convergence of

$$F(x) = \frac{\sin x - x + \frac{x^3}{6}}{x^5}$$

as $x \to 0$.

SOLUTION By repeated application of L'Hôpital's rule (Appendix A), we see that

$$\lim_{x \to 0} F(x) = \frac{1}{120}.$$

To determine the rate of convergence, we proceed as follows. Using Taylor's Theorem (Theorem 1.8) to expand $\sin x$ around $x = 0$, we get

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!}(-\cos(\xi(x))) \tag{1.20}$$

for some $\xi(x)$ between 0 and x. From (1.20) we obtain

$$\sin x - x + \frac{x^3}{6} = \frac{x^5}{120} - \frac{x^7}{7!}\cos(\xi(x)),$$

which gives

$$F(x) = \frac{\sin x - x + \frac{x^3}{6}}{x^5} = \frac{1}{120} - \frac{1}{7!}x^2\cos(\xi(x))$$

so that

$$\left|\frac{F(x) - \frac{1}{120}}{x^2}\right| = \frac{1}{7!}|\cos(\xi(x))| \leq \frac{1}{7!}.$$

Thus $F(x) = \frac{1}{120} + O(x^2)$, and we conclude that $F(x)$ approaches its limit as fast as x^2 approaches zero as $x \to 0$. ▲

EXERCISE SET 1.2.1

1. In order to determine the x-intercept of the line joining the two points (x_0, y_0) and (x_1, y_1) in the x-y plane, we may use

$$x = \frac{x_0y_1 - x_1y_0}{y_1 - y_0}$$

or

$$x = x_0 - \frac{(x_1 - x_0)y_0}{y_1 - y_0}.$$

Use each of these formulas and three-digit rounding arithmetic to calculate the x-intercept of the line joining the points $(x_0, y_0) = (2.78, 1.61)$ and $(x_1, y_1) = (5.91, 3.41)$. Explain which formula is better and why.

2. The two equations

$$12.85x + 9.47y = 4.190 \quad \textbf{(1.21a)}$$

$$7.20x + 5.23y = 2.500 \quad \textbf{(1.21b)}$$

have the unique solution $x = 1.8$, $y = -2.0$. The most commonly used method for solving such equations is to multiply equation (1.21a) by the coefficient of y in equation (1.21b), multiply equation (1.21b) by the coefficient of y in equation (1.21a), and difference the two resulting equations. Then we have

$$[(5.23)(12.85) - (9.47)(7.20)]x = (5.23)(4.190) - (9.47)(2.500).$$

a. Perform the above computations using four-digit decimal arithmetic and rounding, and obtain the values of x and y.

b. Perform the above computations using four-digit decimal arithmetic and chopping, and obtain the values of x and y.

c. Explain why the computed values of x and y are significantly different from the exact values.

3. Calculate $f(x) = \dfrac{e^x - 1}{x}$ for $x = 1.3 \times 10^{-4}$ using five-digit decimal arithmetic with rounding. Rewrite $f(x)$ in a form that avoids the loss of significant digits and evaluate $f(x)$ for $x = 1.3 \times 10^{-4}$ once again. Compare the two results obtained.

4. Calculate $f(x) = \dfrac{e^x - e^{-x}}{2x}$ for $x = 1.3 \times 10^{-4}$ using five-digit decimal arithmetic with rounding. Rewrite $f(x)$ in a form that avoids the loss of significant digits and evaluate $f(x)$ for $x = 1.3 \times 10^{-4}$ once again. Compare the two results obtained.

5. Rewrite each of the following expressions to avoid any possible loss of significance errors in their evaluation at the indicated values of x:

   a. $\dfrac{(\sin x/x)^2}{1 + \cos x}$ for x near π

   b. $\dfrac{e^x - 1 - x}{x^2}$ for x near 0

   c. $\dfrac{1}{1+x} - 1$ for x near 0

6. Rewrite each of the following expressions to avoid any possible loss of significance errors in their evaluation at the indicated values of x:

   a. $\sin 3x - \sin x$ for x near 0

   b. $(1 + x)^{1/3} - 1$ for x near 0

   c. $\sqrt{x^2 + 1} - x$ for very large x

7. Use five-digit arithmetic with rounding to determine the roots of $x^2 - 60x + 1 = 0$ correct to five digits. Use $\sqrt{899} = 29.983$.

8. Use five-digit arithmetic with rounding to determine the roots of $2x^2 - 205x + 3 = 0$ correct to five digits. Use $\sqrt{42001} = 204.94$.

9. Consider generating the sequence $\{p_n\}_{n=0}^{\infty}$, where $p_n = (1/3)^n$, using the recurrence relations

   i. $p_n = \dfrac{5}{6}p_{n-1} - \dfrac{1}{6}p_{n-2}, \quad p_0 = 1, \ p_1 = 1/3$, and

   ii. $p_n = \dfrac{5}{3}p_{n-1} - \dfrac{4}{9}p_{n-2}, \quad p_0 = 1, \ p_1 = 1/3.$

   Determine whether each of these procedures is stable.

10. Consider generating the sequence $\{p_n\}_{n=0}^{\infty}$, where $p_n = (2/3)^n$, using the recurrence relations

    i. $p_n = \dfrac{5}{6}p_{n-1} - \dfrac{1}{9}p_{n-2}, \quad p_0 = 1, \ p_1 = 2/3$, and

    ii. $p_n = \dfrac{14}{3}p_{n-1} - \dfrac{8}{3}p_{n-2}, \quad p_0 = 1, \ p_1 = 2/3.$

    Determine whether each of these procedures is stable.

11. Investigate the validity of the quotation from Mrs. La Touche given at the beginning of this chapter by evaluating the sum $\sum_{i=1}^{10}(1/i^2)$ first as $\frac{1}{1} + \frac{1}{8} + \cdots + \frac{1}{1000}$ and then as $\frac{1}{1000} + \frac{1}{729} + \cdots + \frac{1}{1}$, both using three-digit chopping arithmetic. Explain the results.

12. Investigate the validity of the quotation from Mrs. La Touche given at the beginning of this chapter by evaluating the sum $\sum_{i=1}^{10}(1/i^2)$ first as $\frac{1}{1}+\frac{1}{4}+\cdots+\frac{1}{100}$ and then as $\frac{1}{100}+\frac{1}{81}+\cdots+\frac{1}{1}$, both using three-digit chopping arithmetic. Explain the results.
13. We know that $\lim_{x\to 0}\dfrac{1-\cos x}{x}=0$. What is the rate of convergence?
14. We know that $\lim_{x\to 0}\left(\cos x+\frac{1}{2}x^2\right)=1$. What is the rate of convergence?

1.2.2 ALGORITHMS AND PROGRAMMING

Physical phenomena are frequently described by mathematical problems. The process of obtaining a mathematical problem, called the **mathematical model** corresponding to the physical phenomenon, is known as formulation. This is the first and foremost stage in problem solving.

Since mathematical models usually make simplifying assumptions about the physical situations they model, and different models tend to make different assumptions, a single physical phenomenon may be modeled by several mathematical models. There is usually no such thing as the best model for describing a situation. Once formulated, a mathematical model is often used to predict the behavior of the corresponding physical situation. Therefore, the validity of a model may be decided based on how closely it reproduces the characteristics of the underlying physical phenomenon, and based on the accuracy of its predictions. The analysis and validation of a mathematical model require the solution of the mathematical problem that the model yields. Most practical situations are described by mathematical problems that are not readily solvable analytically, and numerical computations have become an essential part of the solution process. Thus, the second stage in problem solving is the design and selection of an **algorithm** based on a numerical method for the mathematical problem described by the mathematical model. The term **algorithm** means a complete and unambiguous sequence of steps leading to the solution of a mathematical problem.

The next stage is the implementation of the algorithm as a *computer program* in a programming language. Finally, the actual execution of the computer program with various sets of input values constitutes the last stage.

Frequently, errors in the formulation are discovered through numerical experimentation. Hence a numerical analyst often plays a significant role in refining or improving a mathematical model. The design, selection, implementation, execution, and experimentation of an algorithm are the major tasks of a numerical analyst.

Nowadays, since a wide variety of mathematical software is readily available, the implementation stage might seem unnecessary. However, while it is very easy to use available software in principle, it is not so easy in practice. Therefore, the numerical analyst is challenged with another important but difficult task, namely, the "intelligent use" of a standard numerical technique or software for a specific application or situation. In order to accomplish these tasks, the numerical analyst must be aware of the difficulties that might arise due to the computing environment (such as the limited precision arithmetic available on the computer), or due to the limitations of an available piece of

software (such as its storage requirements). The selection of an algorithm requires an understanding of how errors may arise and propagate during its execution. On the other hand, the selection of suitable software requires an understanding of the limitations and efficiency of a particular implementation of the algorithm chosen.

The object of an algorithm in this textbook is to implement a numerical procedure to compute an approximate solution to a problem. In presenting a mathematical problem for computer solution, we should provide the proper input and specify what kind of output is expected. We will use the algorithm notation of Knuth [15]. Each algorithm presented in the text will be given an identifying number (e.g., 1 in the example to follow) and a name (**SUM** in the example) and the steps in the algorithm will be labeled by numbers within parentheses ((1), (2), etc.).

EXAMPLE 1.24

Let us consider the evaluation of the sum

$$\sum_{k=1}^{n} x_k = x_1 + x_2 + \cdots + x_n,$$

where n and the numbers $x_1, x_2 \cdots x_n$ are given. An algorithm for this computation is shown below. ▒

ALGORITHM 1

SUM (x, n, result) [To compute $\sum_{k=1}^{n} x_k$].
1. [Initialize] **result** $\leftarrow 0$.
2. [loop] **for** $k \leftarrow 1$ **to n do** through step 3.
3. [sum] **result** $\leftarrow$ **result** $+ x_k$.
4. [output] **output(result)**. ■

A list of input and output variable names follows the name of the algorithm (similar to the subprogram notation in a programming language). A bold face lower case letter will be used to denote a one-dimensional array, and an upper case letter will be used to denote a two-dimensional array. Each step begins with an explanatory comment for that step. Assignments are written using the $\leftarrow$ (read "gets") operator. The end of an algorithm is identified by a ■. The standard **for**-loop, **while**-loop, **if**···**then**···**else,** and **repeat**-loop constructs will be used as the basic control structures. **Goto** statements will be used occasionally, making use of the labels of various steps in the algorithm. Algorithm 1 illustrates the notation.

The algorithm **SUM** may be translated to a subprogram, which assumes that the input parameter **n** and the array **x** have been assigned values by a main program. Given below are a Fortran subprogram for **SUM** and a main program calling the subprogram.

```
************************************************
*
*     Subprogram for Algorithm 1  (SUM)
*
************************************************
*
      subroutine sum(x, n, result)
*
**** Declarations ******
*
      real result, x(10)
      integer k, n
*
**** Initialize *******
*
      result = 0.0
*
**** Accumulate *******
*
      do 10 k = 1, n
      result = result + x(k)
10    continue
*
**** Output *******
*
      write (*, 100) result
      return
100   format( ' The desired sum is ', f5.1)
      end
************************************************
************************************************
*
*     Main program to invoke sum
*
************************************************
*
      program add
*
**** Declarations **********
*
      real answer, values(10)
      integer index, count
*
**** Prompt and Obtain Input *******
*
      write (*, 100)
      read (*, *) count
      write (*, 200)
      read (*, *) (values(index), index = 1, count)
*
**** Call Subprogram sum ********
*
      call sum(values, count, answer)
*
**** Terminate Program ********
*
      stop
100   format( '  Please input number of terms (n) : ', $)
200   format( '  Please input the n values to be added : ', $)
      end
************************************************
```

With $n = 5$ and

$$(x_1, x_2, x_3, x_4, x_5) = (1, -2, 3, 0, 5),$$

this program prints a sum of 7. ■

For the remaining algorithms in this chapter, we will provide only the subprograms corresponding to the algorithms, and not the main programs. The translation of algorithms into subprograms will be left as an exercise in the other chapters. Let us consider a somewhat nontrivial example next.

EXAMPLE 1.25

We wish to develop an algorithm for evaluating ln(1.5) using the Taylor polynomial $P_N(x)$ (of degree N) for $\ln(1 + x)$ around $x_0 = 0$. We have

$$P_N(x) = \sum_{k=1}^{N} \frac{(-1)^{k+1}}{k} x^k,$$

$$|R_{N+1}(x)| = \left| \frac{(-1)^{N+2}}{N+1} \frac{1}{(1+\xi(x))^{N+1}} x^{N+1} \right|$$

$$\leq \frac{1}{N+1} x^{N+1}.$$

■

The algorithm for the computation of $\ln(1 + x)$ is given below.

ALGORITHM 2

SERIES (`x, tolerance, ln, nmax`) [To compute $\ln(1 + x)$ using the Maclaurin series].
1. [initialize] `n`← 1; `ln` ← 0; `term` ← `x`; `power` ← `x`; `sign` ← −1.
2. [loop] **while** `n` ≤ `nmax` **do** through step 7.
3. [accumulate] `sign`← −`sign`; `ln`←`sign` * `term`.
4. [next power] `power`←`power` * x.
5. [next term] `term`←`power`/(`n`+1).
6. [done?] **if** |`term`| < `tolerance` **then** { **output**(`ln, n`); **exit** }.
7. [advance] `n`←`n`+1.
8. [failed!] **output** ('Sorry, computations unsuccessful'). ■

In order to obtain ln(1.5), we should use $x = 0.5$ as input to Algorithm 2. Suppose that we wish to use a sufficiently large N for which

$$|\ln(1.5) - P_N(0.5)| < \texttt{tolerance}. \tag{1.22}$$

This would mean that the algorithm for solving this problem should test whether the

condition (1.22) is satisfied, and terminate when it is satisfied. Since the remainder term is a measure of the error in the approximation, it is sufficient to check whether the next term being added is smaller than `tolerance`. A Fortran translation of Algorithm 2 is given below.

```
*********************************************************
*
*         Subprogram for Algorithm 2   (SERIES)
*
*********************************************************
*
         subroutine series(x, tol, result, nmax)
*
******** Declarations *********
*
         real x, tol, result, term, power
         integer n, nmax, sign
*
******** Initialize *********
*
         result = 0.0
         n = 1
         term = x
         power = x
         sign = -1
*
******** Main Loop **********
*
7        continue
           sign = - sign
           result = result + sign * term
           power = power * x
           term = power/(n+1)
*
******** Result Acceptable ? ********
*
         if (abs(term).lt.tol) then
           write (*,100) n, result
           return
         else
           n = n + 1
         endif
*
******** Number of Iterations Exceeded? **********
*
         if (n. le. nmax) GOTO 7
         write (*,200)
         return
100      format('  log(1.5) using ',i3,' terms is ',f8.5)
200    format('  Sorry, computations unsuccessful')
         end
*
*********************************************************
```

Note that the smaller the tolerance, the larger the number of terms (N) needed. Thus N may be arbitrarily large. Therefore, it becomes necessary to define the maximum amount of computation we are willing to perform, based on cost considerations. For the present problem, this may be accomplished simply by providing an upper bound for

N, say, **nmax**. When the required value of N exceeds **nmax**, the algorithm may be designed to print a message saying that the computations did not terminate successfully. It is useful to incorporate a stopping technique in each algorithm so that infinite looping may be avoided. We will do this throughout the textbook.

It is assumed that the main program that calls the subprogram will supply the values of **x**, **tolerance**, and **nmax**. For example, corresponding to ln 1.5, we have **x** = 0.5. With **nmax** = 20, the program produces a result of 0.40553 using 9 terms with **tolerance** $= 10^{-4}$, 0.40546 using 12 terms with **tolerance** $= 10^{-5}$, and 0.40547 using 15 terms with **tolerance** $= 10^{-6}$.

The next two examples will illustrate loss of significance and error propagation. They will also illustrate the need for experimentation.

EXAMPLE 1.26

Consider the evaluation of e^{-10} using the Maclaurin expansion for e^x. The Maclaurin polynomial $P_N(x)$ (of degree N) and the corresponding remainder term (or truncation error) are given by

$$P_N(x) = \sum_{k=1}^{N} \frac{x^k}{k!},$$

$$|R_{N+1}(x)| = \left| \frac{x^{N+1}}{(N+1)!} e^{\xi(x)} \right|,$$

$$\leq \left| \frac{x^{N+1}}{(N+1)!} \right|, \quad \text{for } x < 0.$$

Thus in order to obtain an approximation for e^{-x}, we compute $P_N(-x)$, by choosing N large enough so that $|R_{N+1}(x)| <$ **tolerance**, where **tolerance** is a specified error tolerance. An algorithm for this computation follows.

ALGORITHM 3

EXPON (**x**, **tolerance**, **exp**, **NMAX**) [To compute exp(x) using the Maclaurin series].

1. [initialize] **N** ← 1; **exp** ← 1; **term** ← **x**.
2. [loop] **while** N ≤ **NMAX** **do** through step 5.
3. [accumulate] **exp** ← **exp** + **term**.
4. [next term] **N** ← **N** + 1; **term** ← **term** * **x**/**N**.
5. [done?] **if** |**term**| < **tolerance** **then** { **output**(**exp**, **N**); **exit**}.
6. [failed!] **output** ('Sorry, computations unsuccessful'). ■

For the present example, the algorithm will evaluate

$$P_N(-10) = 1 + \frac{(-10)}{1!} + \frac{(-10)^2}{2!} + \cdots + \frac{(-10)^N}{N!}, \tag{1.23}$$

for a large enough N to satisfy

$$|R_{N+1}(-10)| \leq |(-10)^{N+1}/(N+1)!| < \texttt{tolerance}.$$

Corresponding to a tolerance of 10^{-5} we obtain $N = 34$, since $|(-10)^{35}/(35!)| \approx 9.678 \times 10^{-6}$. On the other hand, the computed value of $P_{34}(-10)$ is -6.509318×10^{-5}, while the exact value of e^{-10} is around 4.539993×10^{-5}. Note that e^{-x} can never be negative for any value of x, and we have obtained a negative result! Clearly, this discrepancy does not arise due to a large truncation error, because the algorithm checks the truncation error against the prescribed tolerance.

Shown in Table 1.3 are the summands in (1.23), and the partial sum obtained after each summand is added to **exp** in the algorithm. Note that while the magnitude of the final result e^{-10} is relatively small, several of the summands that contribute to this result are very large. As a matter of fact, while they eventually cancel out, it is these large terms that determine the number of significant digits in the final result. This phenomenon, which Henrici [11] named **smearing,** arises whenever the magnitudes of the terms in a summation are considerably larger than the sum itself. This situation is frequently the case when a series with alternating or mixed signs is accumulated.

There are two possible remedies for this situation. First, note that $e^{-10} = 1/e^{10}$. Therefore, we could form the series for e^{10}, which does not involve cancellation of

Table 1.3

k	term	exp	k	term	exp
0	1.000000	1.000000	18	156.192100	54.499940
1	−10.000000	−9.000000	19	−82.206380	−27.706440
2	50.000000	41.000000	20	41.103190	13.396750
3	−166.666700	−125.666700	21	−19.572950	−6.176195
4	416.666700	291.000000	22	8.896794	2.720599
5	−833.333400	−542.333400	23	−3.868171	−1.147572
6	1388.889000	846.555500	24	1.611738	0.4641658
7	−1984.127000	−1137.572000	25	−0.6446952	−0.1805294
8	2480.159000	1342.587000	26	0.2479597	0.06743029
9	−2755.732000	−1413.145000	27	−0.09183693	−0.02440664
10	2755.732000	1342.587000	28	0.03279890	0.008392263
11	−2505.211000	−1162.624000	29	−0.01130997	−0.002917704
12	2087.676000	925.052200	30	0.003769989	0.0008522850
13	−1605.905000	−680.852400	31	−0.001216126	−0.0003638406
14	1147.075000	466.222300	32	0.0003800392	0.00001619864
15	−764.716500	−298.494200	33	−0.0001151634	−0.00009896477
16	477.947800	179.453600	34	0.00003387159	−0.00006509318
17	−281.145800	−101.692200	35	−0.000009677598	

terms with alternating signs. Using a tolerance of 10^{-5} we obtain $e^{10} \approx 22026.47$ with 35 terms, which yields $e^{-10} \approx 4.5399921 \times 10^{-5}$. Secondly, the algorithm **EXPON** may be used to calculate e^{-1}, and e^{-10} may be obtained using $e^{-10} = (e^{-1})^{10}$. Using a tolerance of 10^{-5}, we obtain $e^{-1} \approx 0.3678819$ with 9 terms, which yields $e^{-10} \approx 4.5402964 \times 10^{-5}$. The latter result is somewhat inferior because of the exponentiation operation involved in $(e^{-1})^{10}$. However, with a tolerance of 10^{-6}, we obtain $e^{-1} \approx 0.3678792$ with only 10 terms, and $e^{-10} \approx 4.5399932 \times 10^{-5}$. Note that such simple remedies may not be available for other series with terms of mixed signs. ■

EXAMPLE 1.27

Consider the evaluation of

$$I_n = \int_0^1 x^n e^{x-1}\, dx \tag{1.24}$$

for some $n > 1$. Note that $I_1 = 1/e \approx 0.3678794$. Using integration by parts for the right member in (1.24) yields

$$\int_0^1 x^n e^{x-1}\, dx = x^n e^{x-1}\Big|_{x=0}^{x=1} - n\int_0^1 x^{n-1} e^{x-1}\, dx.$$

In other words,

$$I_n = 1 - nI_{n-1}. \tag{1.25a}$$

Suppose we wish to obtain I_{12}. Starting with $I_1 = 1/e$, we may use (1.25a) to evaluate I_{12} recursively. We obtain the results shown in Table 1.4.

In particular, note that the computed value for I_{12} is ≈ -4.310974. This is impossible — the value of I_n could never become negative, because the integrand is nonnegative over the entire interval of integration for all n. Moreover,

$$\int_0^1 x^n e^{x-1}\, dx \le \int_0^1 x^n\, dx = \frac{1}{n+1}.$$

Therefore, we must have $I_{11} \le 1/12 \approx 0.083333$. The error in I_{11} is magnified (multiplied by 12) in the calculation of I_{12} since $I_{12} = 1 - 12I_{11}$, resulting in a negative value for I_{12}. This gets worse as n increases. In order to see this more clearly, let us consider

Table 1.4

n	I_n	n	I_n
1	0.3678795	7	0.1124296
2	0.2642411	8	0.1005630
3	0.2072767	9	0.0949326
4	0.1708932	10	0.0506744
5	0.1455340	11	0.4425812
6	0.1267958	12	−4.3109740

I_3 and I_4. We have

$$I_3 = 1 - 3I_2 = 1 - 3(1 - 2I_1) = -2 + (3!)I_1,$$

and

$$I_4 = 1 - 4I_3 = 1 - 4(-2 + (3!)I_1) = 9 - (4!)I_1.$$

By induction, it is easily seen that the round-off error in $I_1 = 1/e$ gets magnified by a factor of $n!$ in the calculation of I_n. To avoid this instability, let us rewrite (1.25a) as

$$I_{n-1} = \frac{1 - I_n}{n} \tag{1.25b}$$

and evaluate I_{12} backwards, starting from a large n. For example, since $I_{20} \leq 1/21 \approx 0.048$, we may begin by setting $I_{20} = 0$ in (1.25b), and compute $I_{19}, I_{18}, \ldots, I_{12}$. The initial error (a rather large error), gets divided by $20 \times 19 \times \cdots 13$ and thus gets reduced considerably. In fact, we obtain $I_{12} \approx 0.07177325$. This is consistent with the fact that $I_{12} \leq 1/13 \approx 0.0769$, and $I_{13} \leq 1/14 \approx 0.0714$. ▩

The two previous examples showed that it is useful to explore alternate methods of calculation and validate our computational results. In the process, it may be necessary to make use of the properties of the problem itself. Recall that we used the fact that $e^{-x} = 1/e^x = (e^{-1})^x$ in Example 1.26, while we rewrote the recurrence relation (1.25a) in the form (1.25b) in Example 1.27. Those examples illustrate the following basic ideas concerning scientific computing in general.

Implementing algorithms as computer programs is a very important part of scientific computing. A numerical analyst should be aware of several aspects of programming. (i) the programming language, (ii) the computer system being used, (iii) the process of debugging and verification of results, and (iv) organization and clear description of computations. Computer programs for numerical methods are usually written in a high-level programming language. Structured languages such as Fortran 77, Pascal, and C help create code that is easy to write, document, understand, debug, and modify if necessary. The accuracy of the numerical solution to a problem is generally *not* affected by the programming language used. Practical considerations such as the storage space required and execution time will become important when we wish to translate an algorithm into a computer program. The use of packaged mathematical software is becoming increasingly popular. We should consider such software items as available tools and understand the principles of the tools. Therefore, computational experience comes through writing code as well as experimenting with available mathematical software. Once an algorithm or software item has been selected, we must study the accuracy of the results, possible sources of error, and their effect on the final answer, and estimate the rounding and truncation errors so that the numerical results can be interpreted properly. It will also be helpful to carry out adequate accuracy checks in order to test the applicability of a specific algorithm to a specific problem.

Here are some specific suggestions for good programming practice:

- Always write out an algorithm for the computations desired. Check the algorithm by applying it to a typical yet simple problem for which the exact answer is known.

- When translating algorithms into computer programs in the language of your choice, developa program so that it can handle a general situation, as opposed to a specific instance of the problem. Remember that a program written for a particular set of numbers must be completely rewritten for another set.
- In order to assist in debugging and understanding how a program operates, you should output enough intermediate results. Make the output self-explanatory by labeling each quantity printed out.
- Echo-printing the input is a useful practice.
- Document your programs adequately so that they may be easily understood by anyone or yourself at a later time. However, avoid extensive commenting.
- If the algorithm for solving a specific problem turns out be large, construct the entire program by building subprograms that correspond to various steps in the algorithm. Divide the algorithm into steps which translate into subprograms that are reasonably small, say less than a page, so that the program becomes more easily readable. This will also facilitate easy debugging. Debug and test each subprogram separately, and then together.

These suggestions are by no means exhaustive. We have not discussed specific ways to program in one language or another in order to keep the treatment very general.

EXERCISE SET 1.2.2

COMPUTER EXERCISES

1. Use the following program segment to determine the machine epsilon for your computer system.

```
*****************************************************
*
*         Program Segment for Machine Epsilon
*
*****************************************************
*
          subroutine maceps(epslon)
*
******** Initialize **************
*
          epslon = 1.0
*
******** Main Loop **************
*
*         continue
            epslon = epslon/2.0
          if (epslon + 1.0 .gt. 1.0) goto 7
          epslon = 2.0 * epslon
          return
          end
*
*****************************************************
```

2. Write an algorithm in the notation of the text to compute the sum $S = \sum_{k=1}^{10} 2^k$. Translate your algorithm to a computer program in a language of your choice, and execute your program.
3. Write an algorithm in the notation of the text to compute the expression

$$z = \sum_{i=1}^{n} y_i^{-1} \prod_{j=1}^{i} x_j,$$

assuming that $x_1, x_2, \cdots, x_n$, and $y_1, y_2, \cdots, y_n$ are given. Translate your algorithm to a computer program in a language of your choice, and execute your program.
4. Construct an algorithm that takes as input an integer n $(n \geq 1)$, $(n+1)$ real numbers $a_0, a_1, \cdots, a_n$, and another real number x, and produces as output the sum

$$P = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n.$$

Translate your algorithm to a computer program in a language of your choice, and execute your program.
5. Consider the conversion of a positive decimal integer x to its binary equivalent. If

$$x = (a_n a_{n-1} \cdots a_1 a_0)_2,$$

then

$$x = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0.$$

Thus we obtain the following algorithm for the conversion.

ALGORITHM 4

DEC-TO-BIN-INT (x, a) [To convert positive decimal integer x into its binary equivalent].
1. [intialize] $k \leftarrow 0$.
2. [loop] **while** $\mathbf{x} \neq 0$ **do** through step 5.
3. [get next bit] $a_k \leftarrow \mathbf{x} \bmod 2$.
4. [modify x] $\mathbf{x} \leftarrow \mathbf{x} \div 2$.
5. [advance] $k \leftarrow k + 1$. ■

In this algorithm, the $\mathbf{x} \bmod 2$ operation corresponds to taking the remainder while the $\mathbf{x} \div 2$ operation corresponds to taking the quotient when $\mathbf{x}$ is divided into 2. Translate this algorithm into a computer program, and test it for (a) $\mathbf{x} = 51$, (b) $\mathbf{x} = 1023$, and (c) $\mathbf{x} = 513$.

6. Consider the conversion of a positive decimal fraction $x < 1$ to its binary equivalent. If

$$x = (.a_1a_2a_3\cdots)_2,$$

then

$$x = a_1 \cdot 2^{-1} + a_2 \cdot 2^{-2} + a_3 \cdot 2^{-3} + \cdots.$$

Thus we obtain the following algorithm for the conversion.

ALGORITHM 5

DEC-TO-BIN-FRAC (x, a) [To convert positive decimal fraction **x** into its binary equivalent].
1. [initialize] $k \leftarrow 1$.
2. [loop] **while** $\mathbf{x} \neq 0$ **do** through step 5.
3. [get next bit] $a_k \leftarrow$ int(2**x**).
4. [modify **x**] **x** $\leftarrow$ frac(2**x**).
5. [advance] $k \leftarrow k + 1$. ■

In this algorithm, the int(2**x**) operation corresponds to taking the integer part of 2**x** while the frac(2**x**) operation corresponds to taking the fractional part of 2**x**. Translate this algorithm into a computer program, and test it for (a) $\mathbf{x} = .625$, (b) $\mathbf{x} = .1$, and (c) $\mathbf{x} = .7$.

7. Translate the algorithm **EXPON** into a computer program in a language of your choice, and use it to repeat the computations of Example 1.26. Use the same program to compute (a) e^{-12}, (b) e^{12}.
8. Incorporate the modifications suggested at the end of the discussion in Example 1.26, and develop a computer program for the computation of e^{-x} for a large positive x. Use your program to compute e^{-12}. Compare with the result obtained in Exercise 7, and with the exact result.
9. Write a computer program to carry out the computation in (1.25b) for the evaluation of I_{12}. Experiment with the starting value I_{20}. Observe what happens when you start with (a) $I_{20} = 10$, (b) $I_{20} = 100$. Do not use subscripted variables.

1.3 DISCUSSIONS

*1.3.1 LITERATURE SURVEY

Computer arithmetic is important to scientific programmers who want to produce portable software that will yield numerical results of reasonable accuracy. In this chapter, we have discussed only floating-point arithmetic, since it is the form of arithmetic used

in today's computers. Hwang [13] gives a complete introduction to floating-point arithmetic of computers. Various other forms of computer arithmetic have been explored. We have already discussed the use of interval arithmetic in the study of rounding errors. Interval arithmetic has grown rapidly in the last decade, and it has now become a subject in its own right, called *interval analysis.* An introduction to the methods and applications of interval analysis may be found in Moore [18] and Alefeld and Herzberger [2]. However, as indicated earlier, interval arithmetic is not widely used in practice because it requires a considerable amount of computational effort and may produce greatly exaggerated error bounds. Another form of arithmetic, known as *rational number arithmetic,* involves the use of a rational number system instead of the real number system. Specialized rounding procedures are available for representing a real number that is not rational. Techniques using rational-number arithmetic are given in Henrici [9] and Gregory and Krishnamoorthy [8]. The objections that apply to interval arithmetic also apply here. Finally, a third form of arithmetic, known as *range arithmetic,* has been studied. In this form of arithmetic, we keep track of the number of "good" significant digits at each stage of a computation. At the end of the computation, we will have an answer and an indication of how many digits in the answer are "good." This method has the same flavor as interval arithmetic and hence the same objections apply here as well. Range arithmetic is dealt with in Aberth [1].

Error propagation, especially with respect to rounding errors, is an important aspect of numerical computations. The current results and techniques are due to Wilkinson [19]. In the *statistical approach,* the rounding error is estimated based on the assumption that the local rounding errors are either uniformly or normally distributed between their extreme values. While this method provides an adequate mathematical theory of rounding errors, it involves a considerable amount of mathematical analysis and requires additional computer time. Further details on the statistical approach may be found in Henrici [10].

Other general references on numerical analysis include Isaacson and Keller [14], Conte and De Boor [6], Atkinson [3], Hildebrand [12], and Burden and Faires [5].

Bibliography

1. Aberth, O., "Precise scientific computation with a microprocessor," *IEEE Transactions on Computers* **C-33** (1984), 685–690.
2. Alefeld, G., and Herzberger, J., *Introduction to Interval Computation,* Translated by Jon Rokne, Academic Press, New York, 1983.
3. Atkinson, K. E., *An Introduction to Numerical Analysis,* 2nd ed., John Wiley & Sons, New York, 1989.
4. Berkey, D., *Calculus,* 3rd ed., Saunders College Publishing, Philadelphia, 1994.
5. Burden, R. L., and Faires, J. D., *Numerical Analysis,* 5th ed., PWS-KENT Publishing Company, Boston, 1993.
6. Conte, S. D., and De Boor, C., *Elementary Numerical Analysis—An Algorithmic Approach,* 3rd ed., McGraw-Hill, New York, 1980.
7. Fulks, W., *Advanced Calculus,* John Wiley & Sons, New York, 1978.
8. Gregory, R. T., and Krishnamoorthy, E. V., *Methods and Applications of Error-Free Computations,* Springer-Verlag, New York, 1984.
9. Henrici, P., "A subroutine for computation with rational numbers," *J. ACM* **3** (1956), 6–9.

10. Henrici, P., *Elements of Numerical Analysis,* John Wiley & Sons, New York, 1964.
11. Henrici, P., *Essentials of Numerical Analysis,* John Wiley & Sons, New York, 1982.
12. Hildebrand, F., *Introduction to Numerical Analysis,* McGraw-Hill, New York, 1966.
13. Hwang, K., *Computer Arithmetic,* John Wiley & Sons, New York, 1979.
14. Isaacson, E., and Keller, H., *Analysis of Numerical Methods,* John Wiley & Sons, New York, 1966.
15. Knuth, D. E., *The Art of Computer Programming,* Vol. 1, Fundamental Algorithms, 2nd ed., Addison Wesley, Reading, Mass., 1973.
16. Liu, C. L., *Introduction to Combinatorial Mathematics,* McGraw-Hill, New York, 1974.
17. Mano, *Computer System Architecture,* Addison Wesley, Reading, Mass., 1981.
18. Moore, R. E., *Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics,* SIAM Publications, Philadelphia, 1979.
19. Wilkinson, J. H., *Rounding Errors in Algebraic Processes,* Prentice-Hall, Englewood Cliffs, N. J., 1963.

1.3.2 SOFTWARE SURVEY

The development of mathematical software is becoming increasingly important, especially with the advancement of technology for the design of computers. Rice [23] and Cowell [20] are useful sources of information on mathematical software. In recent years, *microcomputers* and *supercomputers* have given much more significance to numerical computations and mathematical software development. Software items originally developed for mainframe computers are currently available for microcomputers as well. At the other end of the spectrum, design and analysis of numerical algorithms for supercomputers is a very active area of research today. Rodrigue [24] gives a recent account of numerical methods for scientific computing in a parallel computing environment. Parter [22] and Ortega and Voigt [21] give some idea about how a variety of physical problems lend themselves to supercomputing environments. In addition to the vast amount of mathematical software available for numerical methods, *computer algebra* systems have emerged in the recent years and are widely used by numerical analysts and applied mathematicians. MACSYMA, REDUCE, and muMATH are some of the computer algebra systems available today. One of the most recent and powerful mathematical software systems is Mathematica, which is available on a variety of machines. A description of Mathematica may be found in Wolfram [25].

20. Cowell, W. R., ed., *Sources and Development of Mathematical Software,* Prentice-Hall, Englewood Cliffs, N. J., 1984.
21. Ortega, J., and Voigt, R., "Solution of partial-differential equations on vector and parallel computers," *SIAM Rev.,* **27** (1985).
22. Parter, S., ed., *Large Scale Scientific Computation,* Academic Press, New York, 1984.

23. Rice, J. R., *Numerical Methods, Software, and Analysis,* McGraw-Hill, New York, 1983.
24. Rodrigue, G., *Parallel Processing for Scientific Computing,* SIAM Publications, Philadelphia, 1988.
25. Wolfram, S., *Mathematica: A System for Doing Mathematics by Computer,* Addison Wesley, Reading, Mass., 1988.

1.3.3 CHAPTER SUMMARY

In this chapter, we have reviewed some fundamentals of calculus and discussed the computer representation of numbers. The origin and propagation of rounding errors were studied. Concepts of stability and convergence were introduced. We also introduced the algorithm notation that will be used in this book.

I. The following theorems of calculus were stated without proofs:
   (1.1) Intermediate-Value Theorem.
   (1.2) Rolle's Theorem.
   (1.3) Generalized Rolle's Theorem.
   (1.4) Extreme-Value Theorem.
   (1.5) Weighted Mean-Value Theorem for Sums.
   (1.6) Integral Mean-Value Theorem.
   (1.7) Mean-Value Theorem for Derivatives.
   (1.8) Taylor's Theorem.
   Proofs of Theorems 1.5 through 1.8 are given in Appendix A.

II. Taylor's Theorem in Two Dimensions was stated and proved.

III. The terms *absolute error* and *relative error* were defined.
   The bounds

$$\frac{|x - fl(x)|}{x} \leq \beta^{1-n} \qquad \text{for chopping}$$

and

$$\frac{|x - fl(x)|}{x} \leq \frac{1}{2}\beta^{1-n} \qquad \text{for rounding}$$

were derived.

IV. The *propagated error* in function evaluations and arithmetic operations was studied.

V. It was pointed out that *loss of significance* or *catastrophic cancellation* may occur when we compute the difference of nearly equal quantities. We emphasized that it may be worthwhile to rewrite expressions for evaluation whenever we anticipate such differencing. Rationalization and Taylor's theorem were identified as useful tools in this regard.

VI. *Stable* and *unstable* computations were distinguished by studying the error growth $|E_n|$ after n steps. In particular, we considered
   a. $|E_n| \approx Cn\varepsilon$,
   b. $|E_n| \approx k^n\varepsilon$ for some $k > 1$,

where (1) corresponds to *linear growth* and (2) corresponds to *exponential growth.* A computation that exhibits linear growth of error is *stable,* and a computation that exhibits exponential error growth is *unstable.*

VII. We introduced convergence concepts for sequences and functions as related to numerical analysis.

a. We write $\alpha_n = \alpha + O(\beta_n)$ if $\dfrac{|\alpha - \alpha_n|}{|\beta_n|} \leq K$ for sufficiently large n and say that the rate of convergence of α_n to α is at least of β_n.

b. We write $F(x) = L + O(G(x))$ if $\dfrac{|F(x) - L|}{|G(x)|} \leq K$ for sufficiently small $x > 0$ and say that the rate of convergence of $F(x)$ to L as $x \to 0$ is at least of order $G(x)$.

VIII. We illustrated the algorithm notation of the text by means of several simple examples. We also offered programming suggestions.

REVIEW EXERCISES

1. Let $f(x)$ be a continuous function with $f(1) = -3$ and $f(2) = 10$. Does the graph of $f(x)$ intersect the x-axis at some number $c \in [1, 2]$? Why, or why not?
2. Let $f(x)$ be continuous and differentiable on $[2, 3]$, and let $f(2) = f(3)$. What does Rolle's Theorem permit us to conclude?
3. Does Rolle's Theorem apply to $f(x) = |x|$ on $[-1, 1]$? Why, or why not?
4. What properties of $f(x) = (x - 3)\sin(x - 5)\ln x$ permit us to conclude that there is a number $c \in [1, 6]$ for which $f''(c) = 0$?
5. Why doesn't the Integral Mean-Value Theorem (Theorem 1.6) apply to the case $f(x) = x^3$ and $w(x) = \sin \pi x$ on $[-1, 1]$?
6. Let $f(x)$ be a continuous function on $[a, b]$ with the property that $|f'(x)| \leq M$ for each $x \in [a, b]$. Use the Mean-Value Theorem for Derivatives (Theorem 1.7) to show that for any $x_1, x_2 \in [a, b]$

$$|f(x_1) - f(x_2)| \leq M|x_1 - x_2|.$$

7. Obtain the quadratic and cubic Taylor polynomials for $f(x) = x^4$ around $x_0 = 1$.
8. Obtain the Taylor polynomial of degree 3 for $f(x) = \sin x$ around $x_0 = 0$. Use your polynomial to find an approximation to $\sin 0.02$. Estimate the error in your approximation. Compare with the exact result.
9. Determine the linear and quadratic Taylor polynomials for each of the following:
   a. $f(x, y) = e^x \sin y, \qquad (x_0, y_0) = (0, 0)$
   b. $f(x, y) = e^x \cos y, \qquad (x_0, y_0) = (0, 0)$
10. Convert the following numbers into their decimal equivalents.
    a. $(110010.01)_2$
    b. $(A2D.BC)_{16}$
    c. $(102132.43)_8$

11. For the following numbers x and x_A, determine how many significant digits there are in x_A with respect to x.
    a. $x = 156.23, \quad x_A = 156.224$
    b. $x = 0.0027517, \quad x_A = 0.0027507$
12. Compute using four-digit arithmetic with chopping.
    a. $\frac{2}{7} + \frac{5}{9}$
    b. $\frac{2}{7} \times \frac{5}{9}$
13. Repeat Exercise 12 with rounding
14. Let $f(x) = x^{1/3}$. Estimate the propagated error $\left|x^{1/3} - x_A^{1/3}\right|$ using (1.12).
15. Evaluate $\dfrac{2^x - 1}{x}$ for $x = 0.0001$ using 3, 4, 5, 6, and 7-digit decimal arithmetic and chopping. Rewrite the expression so as to avoid loss of significance, and evaluate using 3, 4, 5, 6, and 7-digit decimal arithmetic and chopping once again.
16. Determine the roots of $x^2 + 80x + 1 = 0$ correct to 5 digits. Use $\sqrt{1599} = 39.987$.
17. Determine whether the following scheme for generating $p_n = (\frac{2}{7})^n$ is stable or not.

$$p_n = p_{n-1} - \frac{10}{49} p_{n-2}, \qquad p_0 = 1, \; p_1 = 2/7$$

18. We know that $\lim_{x \to 0} \ln(1 - x) + xe^{x/2} = 0$. What is the rate of convergence?
19. Write an algorithm in the notation of the text to evaluate $\sin x$ for a given x, using its Taylor series expansion around $x_0 = 0$.

Computer Exercises

20. We wish to evaluate the integral

$$I_n = \int_0^1 \frac{x^n}{x+5} \, dx$$

for some integer $n > 0$. Corresponding to $n = 0$ we have $I_0 = \ln 1.2 \approx 0.1823215$.
a. For $n > 0$, show that

$$I_n = \frac{1}{n} - 5I_{n-1}. \qquad \textbf{(1.26a)}$$

b. Write a computer program to evaluate I_{10} using (1.26a). Do not use subscripted variables. Start with $I_0 = 0.1823215$.
c. Note that (1.26a) may be rewritten as

$$I_{n-1} = \frac{\left(\frac{1}{n} - I_n\right)}{5}. \qquad \textbf{(1.26b)}$$

Write a computer program to evaluate I_{10} using (1.26b). Start with $I_{20} = 0$. Experiment with various starting values for I_{20}. For example, let (i) $I_{20} = 0.1$, and (ii) $I_{20} = 0.005$.

21. Consider using either of the two Maclaurin series

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{x^k}{k},$$

$$\ln(1-x) = -x - \frac{x^2}{2} - \frac{x^3}{3} - \cdots = -\sum_{k=1}^{\infty} \frac{x^k}{k},$$

for the computation of natural logarithms. The first series converges for $-1 < x \leq 1$, while the second one converges for $-1 \leq x < 1$. In order to compute $\ln 0.7$, for example, we may put $x = -0.3$ in the first series, or put $x = 0.3$ in the second one. Write computer programs to use either series for the computation of $\ln z$ for any $z \in (-1, 1)$. Your program should use a prescribed tolerance ε and a maximum number of terms. Output the desired logarithm, and the number of terms needed to achieve the desired accuracy.

Note that the two preceding series may be combined to yield

$$\ln\left(\frac{1+x}{1-x}\right) = 2\left(x + \frac{x^3}{3} + \cdots\right) = 2\sum_{k=1}^{\infty} \frac{x^{2k-1}}{2k-1}.$$

Natural logarithms may be obtained using this new series as well. For example, in order to compute $\ln 0.7$, we let $(1+x)/(1-x) = 0.7$ and obtain $x \approx -0.176470588$. Write a computer program to use this third series for the computation of $\ln z$ for any $z \in (-1, 1)$. Your program should use a prescribed tolerance ε and a maximum number of terms. Output the desired logarithm, and the number of terms needed to achieve the desired accuracy.

Use all three programs to compute $\ln 0.7$, $\ln 1.2$, and $\ln 2$. Use $\varepsilon = 10^{-5}$.

SUGGESTIONS FOR FURTHER STUDY

The following problem is concerned with the rate of convergence of a series, and hence the computing time required to sum the series. The series we wish to consider is given by

$$S = \sum_{n=1}^{\infty} \frac{1}{n^2+1}.$$

It is useful to know that computing the sum S directly on the computer is a waste of computer time. For example, determine how many terms will be necessary in order to compute S to within 10^{-10}. The purpose of this study is to show that some mathematical analysis before performing the actual computations is highly helpful.

1. First, we could make use of the known sum

$$T = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.$$

Instead of computing S directly, we may compute the series corresponding to $T - S$. Call this sum S_1. Then, clearly $S = T - S_1$. Determine how many terms will be necessary in order to compute S_1 to within 10^{-10}. Write a computer program to evaluate S_1, and then S.

2. The idea in 1 may be extended further to the calculation of S_1. For this purpose, we could make use of the known sum

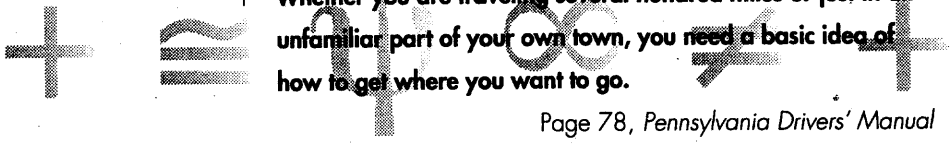
$$V = \sum_{n=1}^{\infty} \frac{1}{n^4} = \frac{\pi^4}{90}.$$

Instead of computing S_1 directly, we may compute the series corresponding to $V - S_1$. Call this sum S_2. Then, clearly $S_1 = V - S_2$. Determine how many terms will be necessary in order to compute S_2 to within 10^{-10}. Write a computer program to evaluate S_2, and then S_1 and S.

Thus number may be said to rule the whole world of quantity, and the four rules of arithmetic may be regarded as the complete equipment of the mathematician.

James Clerk Maxwell

CHAPTER 2 Rootfinding



Whether you are traveling several hundred miles or just in an unfamiliar part of your own town, you need a basic idea of how to get where you want to go.

Page 78, *Pennsylvania Drivers' Manual*

OUTLINE

Problems like the following frequently arise in day-to-day life. A 30-year home mortgage in the amount of \$40,000 is needed. The borrower can afford house payments of at most \$400 per month. What is the greatest interest rate the borrower can afford to pay?

If we let x be the interest rate per pay period, A the amount of mortgage, P the amount of each payment, and n the number of pay periods, then the unknown rate x

satisfies the *ordinary annuity equation* given by

$$A = \frac{P}{x}\left[1 - \frac{1}{(1+x)^n}\right]. \tag{2.1}$$

In terms of the unknown x, (2.1) may be written in the form $f(x) = 0$ for some function $f(x)$. Determining the value of x for which $f(x) = 0$ is equivalent to finding the point of intersection of the graph of $f(x)$ with the x-axis.

In this chapter, we will consider the numerical solution of the general class of problems given by a nonlinear equation of the form

$$f(x) = 0, \tag{2.2}$$

in which $f(x)$ is any continuously differentiable real valued function of a single real variable x. This is called the **rootfinding** problem. The values of x for which the equation $f(x) = 0$ is satisfied are called the **roots** of $f(x) = 0$ or the **zeros** of $f(x)$. The solution of (2.2) by analytical means is almost always impossible, except in rare cases such as those in which $f(x)$ is a factorable polynomial. Instead, the most commonly used methods for the solution of (2.2) are **iterative;** that is, they consist of computing successive values, each depending on one or more previous values, and (we hope) converging to a root. Almost all iterative methods for solving $f(x) = 0$ require one or more initial guesses for the desired solution. The methods considered in this chapter will be called **two-point methods** or **one-point methods,** depending on whether they require two initial guesses or one. We will also study methods for the case where $f(x)$ is a polynomial, since polynomial equations arise in many applications. This is called the **polynomial rootfinding** problem. In a later chapter, we will consider the numerical solution of systems of nonlinear equations.

2.1 Two-Point Methods

To begin with, we will consider methods that need two starting guesses for the solution. Among these, we will first consider those methods that start with an interval enclosing the desired solution, and proceed to obtain a sequence of enclosing intervals of decreasing width. Such methods are called **enclosure methods** or **bracketing methods.** If $f(x)$ is continuous on $[a, b]$ and $f(a)f(b) \leq 0$, then by Theorem 1.1 (Intermediate-Value Theorem) there exists at least one zero of $f(x)$ in $[a, b]$. (Note that the condition $f(a)f(b) \leq 0$ is equivalent to requiring $f(a)$ and $f(b)$ to be of opposite sign.) It is common practice to choose the interval $[a, b]$ so that it contains only one root α of $f(x) = 0$.

2.1.1 BISECTION METHOD

Consider solving the equation $x^3 - 2x - 5 = 0$. Let $f(x) = x^3 - 2x - 5$. Then $f(2) = -1$, $f(3) = 16$, and $f(2)f(3) < 0$. Hence, there must exist at least one number $\alpha \in$

[2, 3] such that $f(\alpha) = 0$. Further, since $f'(x) = 3x^2 - 2 > 0$ for all $x \in [2, 3]$, by Theorem 1.2 (Rolle's Theorem) there is at most one number α such that $f(\alpha) = 0$ in [2, 3]. Suppose now that we approximate α by the midpoint $x^{(1)}$ of the interval [2, 3]. Then we have

$$\alpha \approx x^{(1)} = 2.5 \text{ with absolute error } \leq 0.5.$$

Since $0 \neq f(2.5) = 5.625$ and $f(2)f(2.5) < 0$, it is clear that α must be between 2 and 2.5. If α is approximated once again by the midpoint $x^{(2)}$ of the interval [2, 2.5], we get

$$\alpha \approx x^{(2)} = 2.25 \text{ with absolute error } \leq 0.25.$$

Then

$$f(2) = -1 < 0 < 1.890625 = f(2.25),$$

from which it is clear that $2 < \alpha < 2.25$. This process may be continued until we are satisfied with the approximation obtained for the root. The method of finding the root of a given equation by *bisecting* an interval in each step is called the **bisection method.**

The bisections are normally carried out until the interval enclosing the root becomes "small." A tolerance limit $\varepsilon > 0$ may be specified in order to define "small" precisely. In addition, when programming for the bisection method or any other iterative method, it is helpful to include a bound on the number of iterations permitted. Suppose this number is called *maxit;* then the bisections may be stopped after *maxit* iterations, whether or not the approximations converged to a root to within the tolerance specified. This precaution prevents infinite looping in a computer program, which may be caused by the divergence of the method or by incorrect coding.

EXAMPLE 2.1

Solve $x^3 - 2x - 5 = 0$ for a root in the interval [2, 3] by the bisection method.

SOLUTION

The results corresponding to $\varepsilon = 10^{-6}$ are shown in Table 2.1. The solution of $x^3 - 2x - 5 = 0$ correct to 9 decimal places is 2.094551481. From the computations summarized in Table 2.1, we only know that after 20 iterations, the maximum absolute error is $\approx |2.0945530 - 2.0945521| = 0.0000009 < 10^{-6}$. In fact, the approximate root obtained at the 17^{th} iteration is much closer to the correct solution. However, there is no way we could have known this without knowing the exact solution α! ▲

Table 2.1

k	a	b	x^*	$f(x^*)$
1	2.0000000	3.0000000	2.5000000	5.6250000
2	2.0000000	2.5000000	2.2500000	1.8906250
3	2.0000000	2.2500000	2.1250000	0.3457031
4	2.0000000	2.1250000	2.0625000	−0.3513183
5	2.0625000	2.1250000	2.0937500	−0.0089416
6	2.0937500	2.1250000	2.1093750	0.1668358
7	2.0937500	2.1093750	2.1015625	0.0785622
8	2.0937500	2.1015625	2.0976563	0.0347149

Table 2.1 *(continued)*

k	a	b	x^*	$f(x^*)$
9	2.0937500	2.0976563	2.0957032	0.0128632
10	2.0937500	2.0957032	2.0947266	0.0019547
11	2.0937500	2.0947266	2.0942383	−0.0034949
12	2.0942383	2.0947266	2.0944682	−0.0009296
13	2.0944682	2.0947266	2.0946045	0.0005918
14	2.0944682	2.0946045	2.0945435	−0.0000890
15	2.0945435	2.0946045	2.0945740	0.0002513
16	2.0945435	2.0945740	2.0945587	0.0000805
17	2.0945435	2.0945587	2.0945511	−0.0000043
18	2.0945511	2.0945587	2.0945549	0.0000381
19	2.0945511	2.0945549	2.0945530	0.0000170
20	2.0945511	2.0945530	2.0945521	0.0000069

The following result shows exactly how the interval length decreases from one iteration to the next.

THEOREM **2.1**

Let $f(x)$ be a continuous function on $[a, b]$, and let $f(a)f(b) < 0$. Let α be the exact solution of $f(x) = 0$. Then the approximations $x^{(n)}$ produced by the bisection method satisfy

$$|\alpha - x^{(n)}| \leq \frac{b-a}{2^n}, \qquad n \geq 1. \tag{2.3}$$

□

PROOF Let a_n and b_n denote the end points of the enclosing interval after $n-1$ bisections (see Fig. 2.1). Since $x^{(n)} = (a_n + b_n)/2$, and since $\alpha \in (a_n, b_n)$, it follows that $|\alpha - x^{(n)}| \leq (b_n - a_n)/2$. Finally, since $b_n - a_n = (b-a)/2^{n-1}$, Equation (2.3) follows immediately. ▲

EXAMPLE 2.2

Use Theorem 2.1 to estimate the number of bisections N required to obtain an enclosing interval for a root of some $f(x) = 0$ such that the error is guaranteed not to exceed 10^{-6}. Assume that the starting interval $[a_1, b_1]$ has length 1. In other words, $b_1 - a_1 = 1$.

SOLUTION Let α be the desired root, and N the number of bisections required. Then, $x^{(N)}$ satisfies

$$|\alpha - x^{(N)}| \leq \frac{(b_1 - a_1)}{2^N}.$$

Thus we require

$$2^{-N}(b_1 - a_1) = 2^{-N} < 10^{-6}.$$

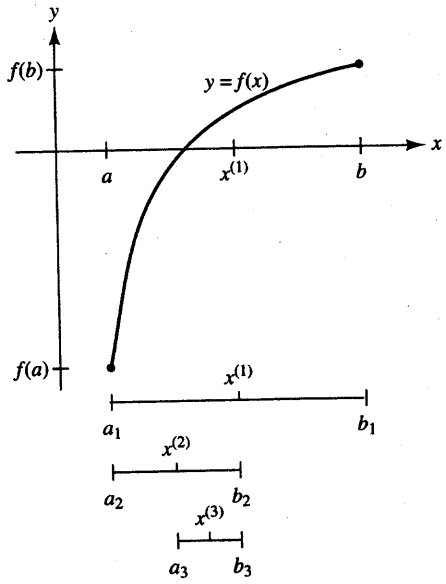


Figure 2.1 Bisection method.

Therefore,

$$-N \log_{10} 2 < -6, \qquad \text{or} \qquad N > \frac{6}{\log_{10} 2} \approx 19.9.$$

Hence, at least 20 iterations are required. Note that this is exactly what happened in Example 2.1. ▲

The inequality (2.3) implies that the sequence $\{x^{(n)}\}$ converges to α at the same **rate** as $\{2^{-n}\}$ converges to zero, or simply $x^{(n)} = \alpha + O(2^{-n})$. However, note that (2.3) gives only a **bound** on the error in the approximation. In general, the errors could be smaller than the bounds. Therefore, the number of iterations estimated as in Example 2.2 could be much higher than the actual number of iterations required.

We conclude this section with an algorithm for the bisection method.

ALGORITHM 6

BISECT$(a, b, x^*, \varepsilon, maxit)$ [To solve $f(x) = 0$ using the bisection method, where $f(x)$ is continuous on $[a, b]$ and $f(a)f(b) \leq 0$]

1. [initialize] `iter` $\leftarrow 0$.
2. [loop] **repeat** through step (5) **until** `iter` $= maxit$.
3. [bisect] $x^* \leftarrow a + (b - a)/2$; `iter` $\leftarrow$ `iter` $+ 1$.
4. [done?] **if** $(b - a)/2 < \varepsilon$ **then** {**output** (x^*); **stop**}.
5. [reduce interval] **if** $f(a)f(x^*) > 0$ **then** $a \leftarrow x^*$ **else** $b \leftarrow x^*$. ■

Even though the bisection method is conceptually simple and easy to implement, it suffers from severe drawbacks. For complicated functions or for larger starting intervals, the bisection method may converge very slowly. Further, a good intermediate approximation may go unnoticed. However, since this method *does* always converge to a solution, it is commonly used as a "starter" for more efficient methods in order to get a reasonably accurate solution much faster.

EXERCISE SET 2.1.1

1. It is useful to sketch the graph of $f(x)$ first in order to solve $f(x) = 0$. Sketch the graph of $f(x) = x^3 - 9x^2 + 24x - 15$ and determine intervals of unit length enclosing the solutions of $f(x) = 0$. [*Hint:* Start by first finding the intervals on which $f(x)$ is monotonically increasing. Identify the critical points, and determine where the graph is concave up and concave down.]
2. Graph $f(x) = x^4 + 2x^3 - 1$ and determine intervals of unit length enclosing the solutions of $f(x) = 0$.
3. Solve the equation $e^x - 3x = 0$ for a root in the interval $[1, 2]$ by using the bisection method. Iterate until the width of the enclosing interval is less than 10^{-3}.
4. Solve the equation $x^3 + x^2 - 3x - 3 = 0$ for a root in the interval $[1, 2]$ by using the bisection method. Iterate until the width of the enclosing interval is reduced to 10^{-2}.
5. Determine how many bisections are needed in order to reduce the enclosing interval from $[0, 1]$ to an interval of width less than 10^{-5} when solving the equation $3x + \sin x - e^x = 0$ for a root in $[0, 1]$.
6. Determine how many bisections are needed in order to obtain an interval of width less than 10^{-5} enclosing a root of the equation $\tan x = x$, starting from the interval $[4, 4.5]$.
7. Using the bisection method, determine the roots of the equations in Exercises 5 and 6 to within 10^{-3}. [*Hint:* Note that the value of x must be in radians.]
8. Solve $x^3 - 2x^2 - 5 = 0$ for a root in the interval $[2, 3]$ by using the bisection method. Iterate until the enclosing interval has width less than 10^{-3}.

COMPUTER EXERCISES

9. Write a computer program for the algorithm **BISECT.** Test your program on Exercises 3, 4, 5, 6, and 8. Use $\varepsilon = 10^{-6}$.
10. Use the bisection method to determine $7^{1/3}$ correct to within 10^{-6}. [*Hint:* Let $f(x) = x^3 - 7$.]
11. Determine $\sqrt{17}$ using the bisection method. Use $\varepsilon = 10^{-5}$.
12. Each of the following functions $f(x)$ is such that $f(0)f(1) < 0$. Use **BISECT** and iterate until a root of $f(x) = 0$ is obtained to within $\varepsilon = 10^{-5}$. Which point does the method yield as the root?
    a. $f(x) = \cos 10x$

b. $f(x) = \dfrac{1}{2x-1}$

c. $f(x) = \begin{cases} 1, & \text{for } x \le 0.25 \\ -1, & \text{for } x > 0.25 \end{cases}$

13. Solve the ordinary annuity equation (2.1) for the situation described at the beginning of Section 2.1 by using the bisection method. Use $\varepsilon = 10^{-6}$.
14. Determine by the bisection method where the cubic $y = x^3 - x + 1$ and the parabola $y = 2x^2$ intersect. Use $\varepsilon = 10^{-5}$.
15. For each of the following equations, determine a root in the indicated intervals by using the bisection method. Iterate until the enclosing interval has width less than 10^{-5}.
    a. $x^3 - x - 1 = 0$ in $[1, 2]$
    b. $x - 2^{-x} = 0$ in $[0, 1]$
    c. $x^3 + 4x^2 - 10 = 0$ in $[1, 2]$
    d. $x^3 - 2x^2 - 5 = 0$ in $[2, 3]$
16. Consider solving the equation $x + 2\cos \pi x + 0.5 = 0$ by using **BISECT.** [*Hint:* x is in radians.]
    a. Determine a root of the equation in $[0.5, 1.5]$, using $\varepsilon = 10^{-5}$.
    b. Suppose that step (5) of the algorithm **BISECT** is changed to

    if $f(b)f(x^*) > 0$ **then** $b \leftarrow x^*$ **else** $a \leftarrow x^*$.

    Use this new version of **BISECT** to solve the equation for a root in the interval $[0.5, 1.5]$. Implement this version as a computer program.
    c. Explain the difference between the answers obtained in (a) and (b).

2.1.2 REGULA-FALSI METHOD

It is evident that the bisection method makes no particular use of the value of $f(x)$ at any point of interest; it uses only the sign of $f(x)$ in the selection of an appropriate interval containing the root. It may be helpful to take into account the actual value of $f(x)$ at any point under inspection. For instance, in Example 2.1, since $f(2) = -1$ and $f(3) = 16$, it is reasonable to expect the root to be closer to 2 than to 3. Hence, instead of considering the mid-point of $[2, 3]$ (*i.e.*, the "average" of the end points 2 and 3), it may be useful to consider a "weighted average" of 2 and 3. The **regula-falsi** or "false position" method sets

$$x^{(1)} = \frac{3 \cdot |f(2)| + 2 \cdot |f(3)|}{|f(2)| + |f(3)|}. \tag{2.4}$$

Since $f(2)$ and $f(3)$ are of opposite sign, (2.4) takes the simple form

$$x^{(1)} = \frac{3 \cdot f(2) - 2 \cdot f(3)}{f(2) - f(3)},$$

which yields

$$x^{(1)} = \frac{3(-1) - 2(16)}{(-1) - (16)} = \frac{35}{17} \approx 2.0588.$$

Then $f(2.0588) \approx -0.3911$ indicates that the root lies in the interval [2.0588, 3]. Repeating the process one more time for the interval [2.0588, 3] yields

$$x^{(2)} = \frac{3(-0.3911) - (2.0588)(16)}{(-0.3911) - (16)} = \frac{34.114}{16.391} \approx 2.0813.$$

Now, since $f(2.0813) \approx -0.1468$, we may conclude that the root lies in the interval [2.0813, 3]. Hence, if we denote by $[a_k, b_k]$ the interval enclosing the root after $k-1$ iterations, then the regula-falsi method obtains the next approximation $x^{(k)}$ for the root by setting

$$x^{(k)} = a_k - \frac{(b_k - a_k) f(a_k)}{f(b_k) - f(a_k)}. \tag{2.5}$$

The sign of $f(x^{(k)})$ may be used to determine whether the root lies in the interval $[a_k, x^{(k)}]$ or in $[x^{(k)}, b_k]$, after which (2.5) may be applied once again and $x^{(k+1)}$ determined. This may be repeated until we are satisfied with the approximate root obtained. It is easily verified that $x^{(k)}$ as given by (2.5) is the point of intersection of the straight line joining the points $(a_k, f(a_k))$ and $(b_k, f(b_k))$. Hence, we may conclude that during each iteration, the regula-falsi method approximates the root of $f(x) = 0$ by replacing $f(x)$ with the *secant* line joining the two points $(a, f(a))$ and $(b, f(b))$, where $[a, b]$ is an interval bracketing the root (see Fig. 2.2). As a result, unlike the bisection method, the regula-falsi method does not produce an interval of "small" width enclosing the root. Therefore, it becomes necessary to formulate general stopping criteria.

Let $\{x^{(n)}\}$ denote the sequence of approximations produced by a numerical method for the solution of $f(x) = 0$. Then we may test whether

$$|x^{(n)} - x^{(n-1)}| < \varepsilon \quad \text{or} \tag{2.6a}$$

$$|f(x^{(n)})| < \varepsilon, \tag{2.6b}$$

for a prescribed tolerance $\varepsilon > 0$. However, both of these tests may sometimes lead

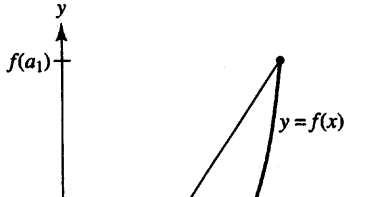


Figure 2.2 Regula-falsi method.

to erroneous conclusions! For example, if $x^{(n)} - x^{(n-1)} = 1/n$, the test (2.6a) will be satisfied for $n \geq 1/\varepsilon$. However, the sequence $\{x^{(n)}\}$ actually diverges. Next, suppose $f(x) = (2-x)^7$, $\alpha = 2$, $x^{(n)} = 2 - \frac{1}{n}$. It is easy to see that $|f(x^{(n)})| < 10^{-2}$ for all $n > 1$, whereas $|\alpha - x^{(n)}| < 10^{-2}$ requires $n > 100$. Hence, the test (2.6b) may indicate convergence long before the approximations actually approach the desired limit! In the absence of any additional information on $f(x)$ and α, the following "relative error" criterion may be more appropriate than (2.6a) or (2.6b).

$$\frac{|x^{(n)} - x^{(n-1)}|}{|x^{(n)}|} < \varepsilon, \qquad x^{(n)} \neq 0, \tag{2.7}$$

where $\varepsilon > 0$ is the prescribed error tolerance. Frequently, (2.7) is used in the form

$$|x^{(n)} - x^{(n-1)}| < \varepsilon |x^{(n)}|. \tag{2.8}$$

In the algorithms to follow, we will use the termination criterion (2.8).

ALGORITHM 7

REG-FALSI(a, b, x^*, ε, *maxit*) [To solve $f(x) = 0$ using the regula-falsi method]

1. [initialize] `iter` $\leftarrow 0$.
2. [loop] **repeat** through step 6 **until** `iter` $=$ *maxit*.
3. [iterate] $x^* \leftarrow a - \dfrac{f(a)(b-a)}{f(b) - f(a)}$; `iter` $\leftarrow$ `iter` $+1$.
4. [done?] **if** `iter` > 1 **and** $|x^* - $ `xold`$| < \varepsilon |x^*|$ **then** [**output** (x^*); **stop**].
5. [save x^*] `xold` $\leftarrow x^*$.
6. [reduce interval] **if** $f(a)f(x^*) > 0$ **then** $a \leftarrow x^*$ **else** $b \leftarrow x^*$. ■

EXAMPLE 2.3

Solve $x^3 - 2x - 5 = 0$ for a root in the interval [2, 3] by the regula-falsi method.

SOLUTION

Table 2.2 shows the results obtained corresponding to $\varepsilon = 10^{-6}$. ▲

Note that the approximation $x^{(k)}$ always lies to the left of the root α in this example. This is because $f(x)$ is increasing and concave upward on the interval $[a_1, b_1] = [2, 3]$, and the secant line is always above the graph of $f(x)$. Similarly, when $f(x)$ is decreasing and concave downward, x^* would always lie to the right of α, and the secant line below the graph of $f(x)$. If the graph of $f(x)$ has significant curvature between a_1 and b_1, the method will be very slow! (See Fig. 2.3.)

Table 2.2

k	a	b	x^*	$f(x^*)$
1	2.0000000	3.0000000	2.0588235	−0.3907999
2	2.0588235	3.0000000	2.0812637	−0.1472041
3	2.0812637	3.0000000	2.0896392	−0.0546765
4	2.0896392	3.0000000	2.0927396	−0.0202029
5	2.0927396	3.0000000	2.0938837	−0.0074505
6	2.0938837	3.0000000	2.0943055	−0.0027457
7	2.0943055	3.0000000	2.0944608	−0.0010116
8	2.0944608	3.0000000	2.0945181	−0.0003727
9	2.0945181	3.0000000	2.0945392	−0.0001373
10	2.0945392	3.0000000	2.0945470	−0.0000506
11	2.0945470	3.0000000	2.0945498	−0.0000186
12	2.0945498	3.0000000	2.0945509	−0.0000069

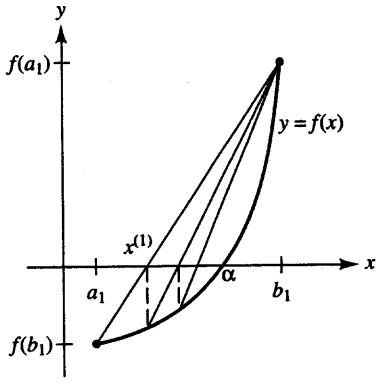


Figure 2.3 Regula-falsi method may be slow.

Modified Regula-Falsi Method. A quick remedy to the situation just described is to replace the secant lines with lines of smaller slope until x^* falls to the opposite side of the root. This is done by replacing the value of $f(x)$ at the stagnant point with $f(x)/2$. (See Fig. 2.4). This is called the **modified regula-falsi method.**

EXAMPLE 2.4

Solve $x^3 - 2x - 5 = 0$ for a root in the interval [2, 3] by the modified regula-falsi method.

SOLUTION Table 2.3 shows the results obtained corresponding to $\varepsilon = 10^{-6}$. ▲

Table 2.3

k	a	b	x*	f(x*)
1	2.0000000	3.0000000	2.0588235	−0.3907999
2	2.0588235	3.0000000	2.1026586	0.0909011
3	2.0588235	2.1026586	2.0943866	−0.0018404
4	2.0943866	2.1026586	2.0945507	−0.0000084
5	2.0945507	2.1026586	2.0945522	0.0000083
6	2.0945507	2.0945522	2.0945515	0.0000000

We conclude this section with an algorithm for the modified regula-falsi method.

ALGORITHM 8

MOD-REG-FAL($a, b, x^*, \varepsilon, maxit$) [To solve $f(x) = 0$ using the modified regula-falsi method]

1. [initialize] `iter` $\leftarrow 0$; `fprev` $\leftarrow f(a)$; $F \leftarrow$ `fprev`; $G \leftarrow f(b)$.
2. [loop] **repeat** through step 10 **until** `iter` $= maxit$.
3. [iterate] $x^* \leftarrow a - \dfrac{F(b-a)}{G-F}$; `iter` $\leftarrow$ `iter` $+ 1$.
4. [done?] **if** `iter` > 1 **and** $|x^* -$ `xold`$| < \varepsilon|x^*|$ **then** {**output**(x^*); **stop**}.
5. [reduce interval] `fstar` $\leftarrow f(x^*)$; **if** $f(a)$(`fstar`) > 0 **then Goto** step 8.
6. [root in $[a, x^*]$] $b \leftarrow x^*$. $G \leftarrow$ `fstar`.
7. [same side?] **if** (`fprev`)(`fstar`) > 0 **then** $F \leftarrow F/2$. **Goto** step 10.
8. [root in $[x^*, b]$] $a \leftarrow x^*$. $F \leftarrow$ `fstar`.
9. [same side?] **if** (`fprev`)(`fstar`) > 0 **then** $G \leftarrow G/2$.
10. [prepare for next iteration] `fprev` $\leftarrow$ `fstar`; `xold` $\leftarrow x^*$. ■

EXERCISE SET 2.1.2

1. Solve $\cos x - x = 0$ for a root in $[0, \pi/2]$ by using the regula-falsi method. Use the termination criterion (2.8) with $\varepsilon = 10^{-3}$. [*Note:* x is in radians.]
2. Solve the equation $x^6 - x - 1 = 0$ for a root in [1, 2] by using the regula-falsi method. Use the termination criterion (2.8) with $\varepsilon = 10^{-2}$.
3. Solve the equation $e^x - 3x = 0$ for a root in [1, 2] by using the regula-falsi method. Use (2.8) with $\varepsilon = 10^{-3}$.

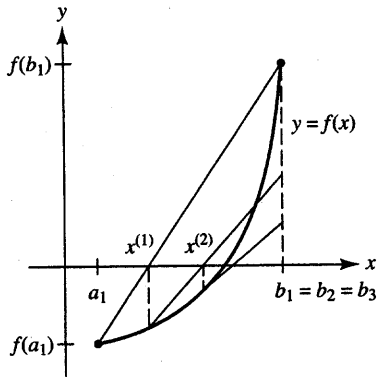


Figure 2.4 Modified regula-falsi method.

4. Solve the equation $x^3 - x - 1 = 0$ for a root in [1, 2] by using the regula-falsi method. Use (2.8) with $\varepsilon = 10^{-3}$.
5. Solve the equation $x + 2\cos \pi x + 0.5 = 0$ for a root in [0.5, 1.5] by using the regula-falsi method. Use (2.8) with $\varepsilon = 10^{-3}$. [*Note:* x is in radians.]

COMPUTER EXERCISES

6. Write a computer program for the algorithm **REG-FALSI.** Use your program to solve the equations of Exercises 1 to 5. Use $\varepsilon = 10^{-6}$.
7. Solve the equation $x^3 + x^2 - 3x - 3 = 0$ for a root in [1, 2] by using the regula-falsi method. Use (2.8) with $\varepsilon = 10^{-6}$.
8. Use the regula-falsi method to determine $7^{1/3}$ correct to within 10^{-6}. [*Hint:* Let $f(x) = x^3 - 7$.]
9. Solve the equation $e^x \cos x = 1$ for a root in the interval [1.2, 1.5] by using the regula-falsi method. Use (2.8) with $\varepsilon = 10^{-4}$.
10. Solve the equation $x^4 - 2x^3 - 4x^2 + 4x + 4 = 0$ for a root in the interval (a) [−2, −1], (b) [0, 2], (c) [2, 3], and (d) [−1, 0] by using the regula-falsi method. Use $\varepsilon = 10^{-3}$.
11. Write a computer program for the algorithm **MOD-REG-FAL.** Use your program to solve the equations of Exercises 1 to 5 and 7 to 10, with $\varepsilon = 10^{-6}$.
12. Solve the ordinary annuity equation (2.1) for the situation described at the beginning of Section 2.1 by using the regula-falsi and modified regula-falsi methods. Use (2.8) with $\varepsilon = 10^{-6}$.
13. Solve each of the following equations by the modified regula-falsi method. Note that your solutions must be approximations to the value of π. Use your own initial guesses, and $\varepsilon = 10^{-6}$.

a. $\sin \frac{x}{2} = 1$

b. $\cos \frac{x}{3} = \frac{1}{2}$

c. $\sin \frac{x}{4} = \cos \frac{x}{4}$

d. $\tan \frac{x}{4} = 1$

2.1.3 SECANT METHOD

The **secant method** is another natural modification of the regula-falsi method that replaces $f(x)$ with the secant and does not necessarily bracket the root during every iteration. Thus it begins with two initial guesses (not necessarily enclosing the root) and produces successive approximations from them. Let the two initial guesses be $x^{(0)}$ and $x^{(1)}$. Then the equation of the secant line joining $(x^{(0)}, f(x^{(0)}))$ and $(x^{(1)}, f(x^{(1)}))$ is

$$y - f(x^{(1)}) = \frac{f(x^{(1)}) - f(x^{(0)})}{x^{(1)} - x^{(0)}}(x - x^{(1)}).$$

The secant line intersects the x-axis at

$$x^* = x^{(1)} - \frac{f(x^{(1)})(x^{(1)} - x^{(0)})}{f(x^{(1)}) - f(x^{(0)})}. \tag{2.9}$$

For the next iteration, we may simply set $x^{(0)} = x^{(1)}$ and $x^{(1)} = x^*$ and calculate the next approximation x^*. (See Fig. 2.5.)

We are now ready to present an algorithm for the secant method.

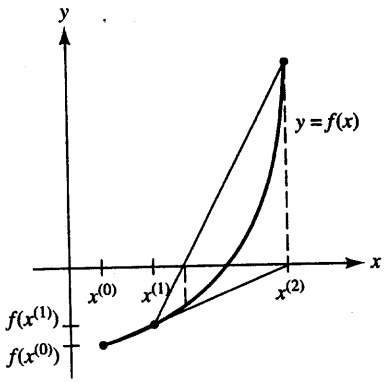


Figure 2.5 Secant method.

ALGORITHM 9

SECANT($x^{(0)}$, $x^{(1)}$, x^*, ε, *maxit*) [To solve $f(x) = 0$ using the secant method, starting with two initial guesses $x^{(0)}$ and $x^{(1)}$]

1. [initialize] `iter` $\leftarrow 0$.
2. [loop] **repeat** through step 5 **until** `iter` $=$ *maxit*.
3. [iterate] $x^* \leftarrow x^{(1)} - \dfrac{f(x^{(1)})(x^{(1)} - x^{(0)})}{f(x^{(1)}) - f(x^{(0)})}$; `iter` $\leftarrow$ `iter` $+ 1$.
4. [done?] **if** $|x^* - x^{(1)}| < \varepsilon|x^*|$ **then** {**output**(x^*); **stop**}.
5. [prepare for next iteration] $x^{(0)} \leftarrow x^{(1)}$; $x^{(1)} \leftarrow x^*$. ■

EXAMPLE 2.5

Solve $x^3 - 2x - 5 = 0$ by the secant method starting with $x^{(0)} = 2$ and $x^{(1)} = 3$.

SOLUTION

Table 2.4 shows the results obtained corresponding to $\varepsilon = 10^{-6}$. It is obvious from the table that the secant method yields the root much faster than the bisection, the regula-falsi, and the modified regula-falsi methods. ▲

Table 2.4

k	$x^{(0)}$	$x^{(1)}$	x^*	$f(x^*)$
1	2.0000000	3.0000000	2.0588235	−0.3907999
2	3.0000000	2.0588235	2.0812637	−0.1472041
3	2.0588235	2.0812637	2.0948241	0.0030438
4	2.0812637	2.0948241	2.0945494	−0.0000229
5	2.0948241	2.0945494	2.0945515	0.0000001

EXERCISE SET 2.1.3

1. Solve each of the following equations for a root in the indicated interval by using the secant method. In each case, use the end points of the interval as starting values. Use the termination criterion (2.8) with $\varepsilon = 10^{-3}$.
   - **a.** $\cos x - x = 0$ in $[0, \pi/2]$ [*Note:* x is in radians]
   - **b.** $x^6 - x - 1 = 0$ in $[1, 2]$
   - **c.** $e^x - 3x = 0$ in $[1, 2]$
2. Solve each of the following equations for a root in the indicated interval by using the secant method. In each case, use the left end point and the mid-point of the interval as starting values. Use the termination criterion (2.8) with $\varepsilon = 10^{-3}$.
   - **a.** $x^3 + x^2 - 3x - 3 = 0$ in $[1, 2]$
   - **b.** $e^x \cos x = 1$ in $[1.2, 1.5]$
   - **c.** $x + 2\cos \pi x + 0.5 = 0$ in $[0.5, 1.5]$ [*Note:* x is in radians]

COMPUTER EXERCISES

3. Write a computer program for the algorithm **SECANT.** Use your program to solve the equations of Exercises 1 and 2. Use $\varepsilon = 10^{-6}$.
4. Use the secant method to determine $7^{1/3}$, with $\varepsilon = 10^{-6}$. Use your own initial guesses. [*Hint:* Let $f(x) = x^3 - 7$.]
5. Determine $\sqrt{17}$ using the secant method. Let $x^{(0)} = 4$, $x^{(1)} = 3$. Use $\varepsilon = 10^{-6}$.
6. Solve the equation $x^4 - 2x^3 - 4x^2 + 4x + 4 = 0$ for a root in the interval (a) $[-2, -1]$, (b) $[0, 2]$, (c) $[2, 3]$, and (d) $[-1, 0]$ by using the secant method. Use the right end point and the mid-point of each interval as the starting values. Use $\varepsilon = 10^{-6}$.
7. Solve each of the following equations by using the secant method. Note that your solutions must be approximations to the value of π. Use your own initial guesses, and $\varepsilon = 10^{-5}$.
   a. $\sin \frac{x}{2} = 1$
   b. $\cos \frac{x}{3} = \frac{1}{2}$
   c. $\sin \frac{x}{4} = \cos \frac{x}{4}$
   d. $\tan \frac{x}{4} = 1$

2.2 One-Point Methods

The methods discussed in this section, Newton's method and fixed-point iteration, will require only one initial guess (as opposed two required by the bisection, regula-falsi, and secant methods) for the solution of $f(x) = 0$. Also presented in this section is a general framework for the analysis of one-point iteration methods. The section will conclude with a discussion of some special situations arising in the solution of $f(x) = 0$ by these methods.

2.2.1 NEWTON'S METHOD

Instead of working with the secant line joining two points on the graph of $y = f(x)$, Newton's method uses the tangent at *one* point on $y = f(x)$. Therefore, it requires only one initial guess instead of two. Let the initial guess be $x^{(0)}$. Then the equation of the tangent at $(x^{(0)}, f(x^{(0)}))$ is

$$y - f(x^{(0)}) = f'(x^{(0)})(x - x^{(0)}).$$

This tangent line intersects the x-axis at

$$x^* = x^{(0)} - \frac{f(x^{(0)})}{f'(x^{(0)})}. \tag{2.10}$$

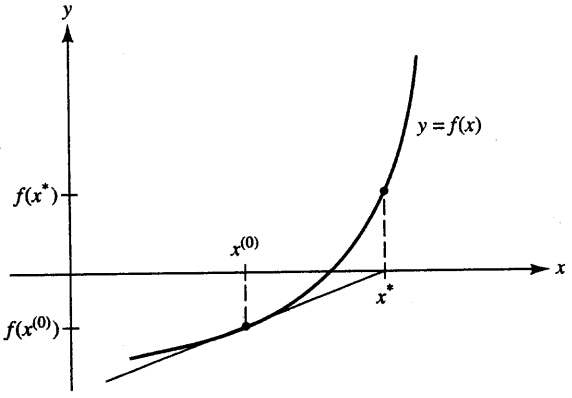


Figure 2.6 Newton's method.

For the next iteration, we may simply set $x^{(0)} = x^*$ and obtain the next x^* by using (2.10) once again. (See Fig. 2.6.)

EXAMPLE 2.6

Solve $x^3 - 2x - 5 = 0$ by Newton's method starting with $x^{(0)} = 2$.

SOLUTION

Newton's method yields the formula

$$x^* = x^{(0)} - \frac{(x^{(0)})^3 - 2x^{(0)} - 5}{3(x^{(0)})^2 - 2}$$

for the equation $f(x) = x^3 - 2x - 5 = 0$. Table 2.5 shows the results obtained corresponding to an error tolerance of $\varepsilon = 10^{-6}$. It is obvious from the table that Newton's method is even faster than the secant method. ▲

Table 2.5

k	$x^{(0)}$	$f(x^{(0)})$	$f'(x^{(0)})$	x^*
1	2.0000000	−1.0000000	10.0000000	2.1000000
2	2.1000000	0.0610000	11.2300000	2.0945681
3	2.0945681	0.0001857	11.1616468	2.0945515
4	2.0945515	0.0000000	11.1614377	2.0945515

ALGORITHM 10

NEWTON(x^*, ε, *maxit*) [To solve $f(x) = 0$ using Newton's method, starting with an initial guess x^*]

1. [initialize] `iter` $\leftarrow 0$.
2. [loop] **repeat** through step 4 **until** `iter` $=$ *maxit*.
3. [iterate] `xold` $\leftarrow x^*$; $x^* \leftarrow x^* - f(x^*)/f'(x^*)$; `iter` $\leftarrow$ `iter` $+ 1$.
4. [done?] **if** $|$`xold` $- x^*| < \varepsilon|x^*|$ **then** {**output**(x^*); **stop**}. ■

Newton's method happens to be a particular instance of a more general method known as *fixed-point iteration,* which will be discussed in the next subsection.

EXERCISE SET 2.2.1

1. Solve each of the following equations by using Newton's method, starting with the indicated value of $x^{(0)}$. Use $\varepsilon = 10^{-3}$.
   - **a.** $x^3 - 2x^2 - 5 = 0, x^{(0)} = 2$
   - **b.** $x^6 - x - 1 = 0, x^{(0)} = 1$
   - **c.** $e^x - 3x = 0, x^{(0)} = 2$
2. Solve each of the following equations by using Newton's method, starting with the indicated value of $x^{(0)}$. Use $\varepsilon = 10^{-3}$.
   - **a.** $4e^{-x}\cos x = 1, x^{(0)} = 1$
   - **b.** $x^3 + 4x^2 - 10 = 0, x^{(0)} = 2$
   - **c.** $x + 2\cos \pi x + 0.5 = 0, x^{(0)} = 1$
3. Use Newton's method to determine $7^{1/3}$ correct to within 10^{-4}. [*Hint:* Let $f(x) = x^3 - 7$.]
4. Determine $\sqrt{17}$ by using Newton's method. Let $x^{(0)} = 4$. Use $\varepsilon = 10^{-4}$.

COMPUTER EXERCISES

5. Write a computer program for the algorithm **NEWTON.** Use your program to solve Exercises 1 to 4. Use $\varepsilon = 10^{-6}$.
6. The *diode equation*

$$i = I_s(e^{v/\theta} - 1)$$

determines current i (in amperes) through a diode, where I_s is the saturation current (in amperes), θ is the diode variable, and v is the voltage across the diode. A junction diode for which $I_s = 20\ \mu$A and $\theta = 0.052$ must also satisfy the equation $v + 10^4 i = 4$. Find the current i (in amperes) through the diode by using Newton's method. Use $i = I_s v/\theta$ to obtain an initial approximation. [Note: μA $= 10^{-6}$ ampere]

7. The *van der Waals equation*

$$\left(P + \frac{a}{V^2}\right)(V - b) = nRT$$

generalizes the ideal gas law $PV = nRT$, where R = gas constant = 0.08205 l·atm/mole·°K and n is the number of moles of gas. For isobutane, $a = 12.87$ atm/l^2 and $b = 0.1142$ l. Using Newton's method, determine the volume of 1 mole of isobutane at a temperature $T = 313$°K and a pressure of 2 atm. Use the ideal gas law to obtain an initial guess.

8. In the analysis of the antisymmetric buckling of beams, a factor ϕ known as the *stability factor* must be determined. It is known that ϕ satisfies $0 < \phi \leq \pi/2$, and

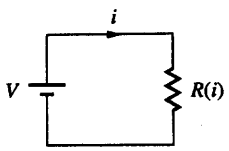
$$6\cos\phi = \gamma\phi\sin\phi,$$

where γ depends on the geometry and the critical stress on the beams. Determine ϕ given $\gamma = 0.1, 0.5, 1$, and 5 by using Newton's method. Use your own initial guesses, and $\varepsilon = 10^{-3}$.

9. The resistance $R(i)$ in the following electrical network varies with the current i through it according to

$$R(i) = A + Bi^{3/2}.$$

By Ohm's law, $V = R(i) \cdot i$. Determine i, given $V = 5$, $A = 100$, and $B = 10$.



10. According to *Archimedes' law,* when a solid of density σ is placed in a liquid of density ρ, it will sink to a depth h that displaces an amount of liquid whose weight equals the weight of the solid. For a sphere of radius r, Archimedes' law translates to

$$\frac{1}{3}\pi(3rh^2 - h^3)\rho = \frac{4}{3}\pi r^3\sigma.$$

Given $r = 5$, $\rho = 1$, and $\sigma = 0.6$, determine h.

2.2.2 FIXED-POINT ITERATION

The method discussed in this subsection is suitable for equations expressed in the form

$$x = g(x)$$

for some function $g(x)$.

DEFINITION **2.1**

*A number α is said to be a **fixed-point** of a function $g(x)$ if $\alpha = g(\alpha)$.*

In other words, a solution to the equation $x = g(x)$ is called a fixed-point of $g(x)$.

EXAMPLE 2.7

The function $g(x) = x^2$ has fixed points at 0 and 1, since $g(0) = 0$ and $g(1) = 1$. Therefore, $x = 0$ and $x = 1$ are the solutions of the equation $x = g(x) = x^2$. ▒

Let us first establish the relationship between the process of obtaining the fixed-point of a function $g(x)$ and the rootfinding problem $f(x) = 0$. For this purpose, we note that the roots of $f(x) = 0$ correspond to the solutions of the equation $x = g(x)$ when $g(x) = x - f(x)$. For example, the equation $f(x) = x - x^2 = 0$ may be solved by obtaining the fixed-points of $g(x) = x^2$. Therefore, if a fixed-point for any given function $g(x)$ could be determined, then every rootfinding problem $f(x) = 0$ could be solved as well, by simply setting $g(x) = x - f(x)$. We will first consider some results regarding the existence and uniqueness of fixed-points in general.

THEOREM **2.2**

Suppose $g(x)$ is a continuous function on $[a, b]$ satisfying

$$a \le g(x) \le b \quad \textit{for all} \quad a \le x \le b. \tag{2.11}$$

Then $g(x)$ has a fixed-point in $[a, b]$. (See Fig. 2.7.) □

PROOF When $g(a) = a$ or $g(b) = b$, the existence of a fixed-point is obvious. Therefore it suffices to consider the case corresponding to $g(a) \neq a$ and $g(b) \neq b$. In this case, since g maps $[a, b]$ into itself, we must have $g(a) > a$ and $g(b) < b$. Defining $h(x) = g(x) - x$,

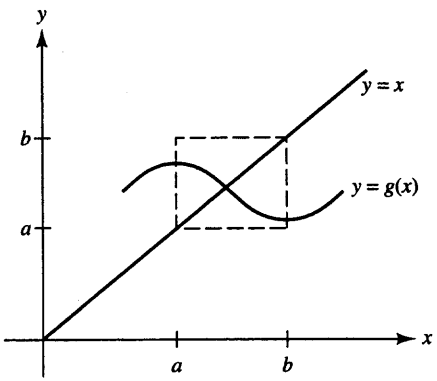


Figure 2.7 Existence of a fixed-point.

we see that $h(a) > 0$ and $h(b) < 0$. By Theorem 1.1 (Intermediate-Value Theorem), there is an $\alpha \in (a, b)$ for which $h(\alpha) = 0$, so $\alpha = g(\alpha)$. ▲

Remark The condition (2.11) is sometimes denoted by $g([a, b]) \subseteq [a, b]$. The notation $g([a, b])$ is used to denote the set $\{g(x) \mid x \in [a, b]\}$. The condition (2.11) is only sufficient but not necessary for a fixed-point to exist in the interval $[a, b]$. ▒

EXAMPLE 2.8

Does Theorem 2.2 guarantee the existence of a fixed-point for $g(x) = x^3$ in the interval (a) $[-\frac{1}{2}, \frac{1}{2}]$, (b) $[0, 1]$, and (c) $[0, 2]$?

SOLUTION The fixed-points of $g(x) = x^3$ are $x = 0$, $x = -1$, and $x = 1$. We consider (a), (b), and (c) separately.

a. We have

$$g([-\tfrac{1}{2}, \tfrac{1}{2}]) = [-\tfrac{1}{8}, \tfrac{1}{8}] \subseteq [-\tfrac{1}{2}, \tfrac{1}{2}].$$

Therefore, Theorem 2.2 guarantees the existence of a fixed-point for $g(x) = x^3$ in the interval $[-\frac{1}{2}, \frac{1}{2}]$. The fixed-point $x = 0$ is the only fixed-point in $[-\frac{1}{2}, \frac{1}{2}]$.

b. We have

$$g([0, 1]) = [0, 1] \subseteq [0, 1].$$

Therefore, Theorem 2.2 guarantees the existence of a fixed-point for $g(x) = x^3$ in the interval $[0, 1]$. In this case, the fixed-points $x = 0$ and $x = 1$ are both in $[0, 1]$.

c. We have

$$g([0, 2]) = [0, 8] \not\subset [0, 2].$$

Therefore, Theorem 2.2 does not guarantee the existence of a fixed-point for $g(x) = x^3$ in the interval $[0, 2]$. However, $g(x)$ has two of its fixed-points, $x = 0$ and $x = 1$, in the interval $[0, 2]$. ▲

Remark Example 2.8 shows that while (2.11) may be used to detect the existence of a fixed-point, it is not useful for determining whether there is only one or more fixed-points in a given interval $[a, b]$. Moreover, it shows that (2.11) is only sufficient but not necessary for the existence of a fixed-point. The following result shows the conditions for the existence of a unique fixed-point. ▒

THEOREM 2.3

Let $g(x)$ be continuous on $[a, b]$. Suppose $g(x)$ satisfies (2.11), and assume that there exists a constant $\kappa > 0$ such that

$$|g'(x)| \leq \kappa < 1 \qquad \textit{for all } x \in [a, b]. \tag{2.12}$$

Then $g(x)$ has a unique fixed-point $\alpha \in [a, b]$. □

PROOF Since $g([a, b]) \subseteq [a, b]$, it follows from Theorem 2.2 that there is a number $\alpha \in [a, b]$ such that $\alpha = g(\alpha)$. Suppose now that there are two numbers, say $\alpha, \beta \in [a, b]$, with

$$\alpha = g(\alpha) \qquad \text{and} \qquad \beta = g(\beta).$$

Then, by the Mean-Value Theorem for Derivatives (Theorem 1.7),

$$|\alpha - \beta| = |g(\alpha) - g(\beta)| = |g'(\xi)| \cdot |(\alpha - \beta)|,$$

for some ξ between α and β. Since $|g'(x)| \leq \kappa < 1$ for all $x \in [a, b]$, (2.12) is equivalent to

$$|\alpha - \beta| \leq \kappa |\alpha - \beta| < |\alpha - \beta|,$$

a contradiction. Therefore, $\alpha = \beta$. ▲

As the following examples will illustrate, the condition (2.12) is sufficient but not necessary for a unique fixed-point to exist.

EXAMPLE 2.9

Does Theorem 2.3 guarantee the existence of a unique fixed-point for $g(x) = x^3$ in the interval (a) $[-\frac{1}{2}, \frac{1}{2}]$, (b) $[0, 1]$, and (c) $[-1, 1]$?

SOLUTION The fixed-points of $g(x) = x^3$ are $x = 0$, $x = -1$, and $x = 1$. We consider the three cases separately.

a. We have

$$g([-\tfrac{1}{2}, \tfrac{1}{2}]) = [-\tfrac{1}{8}, \tfrac{1}{8}] \subseteq [-\tfrac{1}{2}, \tfrac{1}{2}].$$

Also, since $g'(x) = 3x^2$, we have $|g'(x)| \leq \frac{3}{4} < 1$. Hence, Theorem 2.3 guarantees the existence of a unique fixed-point for $g(x) = x^3$ in the interval $[-\frac{1}{2}, \frac{1}{2}]$. The fixed-point $x = 0$ is the only fixed-point in $[-\frac{1}{2}, \frac{1}{2}]$.

b. We have

$$g([0, 1]) = [0, 1] \subseteq [0, 1].$$

Also, since $g'(x) = 3x^2$, it is clear that $|g'(x)| \geq 1$ for $|x| \geq 1/\sqrt{3} \approx 0.577$. In other words, for $x > 1/\sqrt{3}$, we have $|g'(x)| \not< 1$. Therefore, Theorem 2.3 does not guarantee the existence of a unique fixed-point for $g(x) = x^3$ in the interval $[0, 1]$. However, the fixed-points $x = 0$ and $x = 1$ are both in $[0, 1]$.

c. We have

$$g([-1, 1]) = [-1, 1] \subset [-1, 1].$$

As before, since $g'(x) = 3x^2$, we see that $|g'(x)| \geq 1$ for $|x| \geq 1/\sqrt{3} \approx 0.577$. Therefore, $|g'(x)| \not< 1$ for some $x \in [-1, 1]$. We conclude that Theorem 2.3 does not guarantee the existence of a unique fixed-point for $g(x) = x^3$ in the interval $[-1, 1]$. However, all three fixed-points, $x = 0$, $x = -1$, and $x = 1$, of $g(x)$, are in the interval $[-1, 1]$. ▲

EXAMPLE 2.10

Determine whether $g(x) = 3 - \dfrac{2}{x}$ has a unique fixed-point in the interval [1.5, 3.0].

SOLUTION Since $g(x) = 3 - \dfrac{2}{x}$ monotonically increases with x, it is easy to see that

$$g([1.5, 3.0]) = [g(1.5), g(3.0)] = \left[\tfrac{5}{3}, \tfrac{7}{3}\right] \subset [1.5, 3.0].$$

Hence (2.11) is satisfied. Then, since $g'(x) = 2/x^2 > 0$ decreases monotonically with increasing x, we have $|g'(x)| \leq |g'(1.5)| = \frac{8}{9} < 1$, thus satisfying (2.12). By Theorem 2.3, we may conclude that $g(x)$ has a unique fixed-point in [1.5, 3.0]. ▲

EXAMPLE 2.11

Does Theorem 2.3 guarantee the existence of a unique fixed-point for $g(x) = 4^{-x}$ in the interval [0, 1]?

SOLUTION Since $g(x) = 4^{-x}$ monotonically decreases with increasing x, it is clear that $g([0, 1]) = [g(1), g(0)] = [0.25, 1] \subset [0, 1]$. Therefore, by Theorem 2.2, there is at least one fixed-point for $g(x)$ in [0, 1]. Now, $g'(x) = -4^{-x} \ln 4$ yields $g'(0) = -1.386\cdots$. In other words, $|g'(x)| \not< 1$ for some $x \in [0, 1]$. Therefore, Theorem 2.3 does not guarantee a unique fixed-point in [0, 1]. However, since $g(x)$ is monotonic, the fixed-point in [0, 1] is indeed unique. (See Fig. 2.8.) ▲

For the functions $g(x) = x^2$ and $g(x) = x^3$ considered in the preceding examples, it is easy to determine the fixed-points analytically by solving the algebraic equations corresponding to $x = g(x)$. For general functions $g(x)$, it may be possible only to

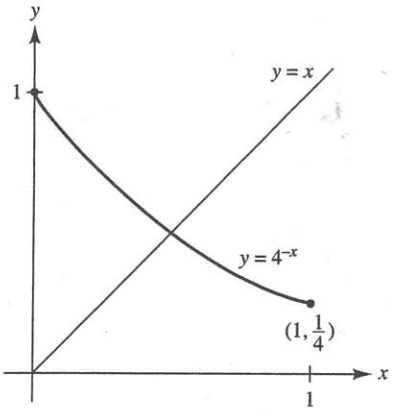


Figure 2.8 Unique fixed-point for $g(x) = 4^{-x}$ in [0, 1].

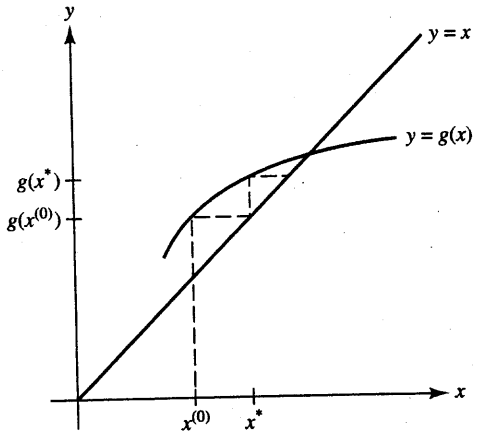


Figure 2.9 Fixed-point iteration.

obtain an approximation to the fixed-point. In order to obtain such an approximation, an iteration scheme described by

$$x^{(n+1)} = g(x^{(n)}), \qquad n = 0, 1, \cdots \tag{2.13}$$

may be used, starting with an initial guess $x^{(0)}$. If the sequence $\{x^{(n)}\}_{n=0}^{\infty}$ approaches some α, and $g(x)$ is continuous, then by Theorem A.1 (see Appendix A),

$$\alpha = \lim_{n\to\infty} x^{(n+1)} = \lim_{n\to\infty} g(x^{(n)}) = g\left(\lim_{n\to\infty} x^{(n)}\right) = g(\alpha),$$

or simply $\alpha = g(\alpha)$. In other words, if the iterates produced by (2.13) converge to a number α, then α is a fixed-point of $g(x)$. The scheme corresponding to (2.13) is called **fixed-point iteration** (see Fig. 2.9).

We are now ready to present the fixed-point iteration method as an algorithm.

ALGORITHM 11

FIX-PT(x^*, ε, *maxit*) [To solve $f(x) = 0$, starting with an initial guess x^* and using the fixed-point iteration method. It is assumed that the equation $f(x) = 0$ is first rewritten in the form $x = g(x)$ and the description of $g(x)$ is available.]

1. [initialize] `iter` $\leftarrow 0$.
2. [loop] **repeat** through step 4 **until** `iter` $=$ *maxit*.
3. [iterate] `xold` $\leftarrow x^*$; $x^* \leftarrow g(x^*)$; `iter` $\leftarrow$ `iter` $+ 1$.
4. [done?] **if** $|$`xold` $- x^*| < \varepsilon |x^*|$ **then** {**output** (x^*); **stop**}. ■

As mentioned earlier, we may rewrite the equation $f(x) = 0$ in the form $x = g(x)$ for some $g(x)$ so that a fixed-point of $g(x)$ is a root of $f(x) = 0$. There may be many ways in which this can be accomplished. In other words, there may be several different possibilities for the function $g(x)$ that facilitate the use of fixed-point iteration. In any case, Algorithm 11 may be used to solve the rootfinding problem for $f(x) = 0$ once an appropriate $g(x)$ has been chosen. Thus it becomes necessary to decide what constitutes a proper selection of $g(x)$. Let us begin with a simple example.

EXAMPLE 2.12

Consider solving $x^2 - 5 = 0$. Suppose we rewrite this equation as $x = 5/x$, so we have $g(x) = 5/x$. Then, with $x^{(0)} = 2$, the scheme $x^{(n+1)} = g(x^{(n)})$ will generate a nonconverging sequence of iterates (alternating between 2 and 2.5). In other words, $g(x) = 5/x$ is not a very good choice. ▒

EXAMPLE 2.13

Consider solving the equation $f(x) = x^3 - 2x - 5 = 0$ for a root in [2, 3]. We first rewrite $f(x) = 0$ in the form $x = g(x)$ in two different ways as follows.

$$x = g_1(x) = \frac{x^3 - 5}{2}, \tag{2.14a}$$

$$x = g_2(x) = (2x + 5)^{1/3}. \tag{2.14b}$$

Then we iterate using $x^{(n+1)} = g_1(x^{(n)})$ and $x^{(n+1)} = g_2(x^{(n)})$ for $n = 0, 1, \cdots$, starting with $x^{(0)} = 2$. The computed iterates are shown in Table 2.6. Further calculations show that (2.14a) produces a diverging sequence of negative numbers, and (2.14b) produces a converging sequence. In eight iterations, (2.14b) produces numbers approaching 2.0945515, which is a very good approximation to a root of the equation $x^3 - 2x - 5 = 0$. In other words, $g_2(x)$ is an appropriate choice for the convergence of fixed-point iteration, while $g_1(x)$ is not. ▒

Table 2.6

$n+1$	$g_1(x^{(n)})$	$g_2(x^{(n)})$
1	1.5000000	2.0800838
2	−0.8125000	2.0923507
3	−2.7681885	2.0942170
4	−13.1061307	2.0945007
5	−1128.1243667	2.0945438

The following result may be used to determine whether a particular choice for $g(x)$ is appropriate for fixed-point iteration.

THEOREM **2.4**

Let $g(x)$ be continuous on $[a, b]$. Suppose $g([a, b]) \subseteq [a, b]$, and $|g'(x)| \leq \kappa < 1$ for all $x \in [a, b]$. Then the iterative scheme (2.13) will converge to the unique fixed-point $\alpha \in [a, b]$ for any choice of initial approximation $x^{(0)} \in [a, b]$, and

$$|\alpha - x^{(n)}| \leq \frac{\kappa^n}{1-\kappa}|x^{(1)} - x^{(0)}|. \tag{2.15}$$

□

PROOF Since $g([a, b]) \subseteq [a, b]$, $x^{(0)} \in [a, b] \Rightarrow x^{(n)} \in [a, b]$ for all n. Further,

$$|\alpha - x^{(n)}| = |g(\alpha) - g(x^{(n-1)})| \leq \kappa|\alpha - x^{(n-1)}|.$$

By induction, it follows that

$$|\alpha - x^{(n)}| \leq \kappa^n|\alpha - x^{(0)}|, \quad n \geq 1. \tag{2.16a}$$

As $n \to \infty$, $\kappa^n \to 0$, thereby showing $x^{(n)} \to \alpha$. For the bound (2.15), we have

$$|\alpha - x^{(0)}| = |\alpha - x^{(1)}| + |x^{(1)} - x^{(0)}| \leq \kappa|\alpha - x^{(0)}| + |x^{(1)} - x^{(0)}|,$$

which gives

$$|\alpha - x^{(0)}| \leq \frac{1}{1-\kappa}|x^{(1)} - x^{(0)}|. \tag{2.16b}$$

Then, (2.16a) and (2.16b) together yield (2.15). (See Fig. 2.10.) ▲

Remark We call a function $g(x)$ that satisfies the hypotheses of Theorems 2.2 through 2.4 on an interval $[a, b]$ a ***contraction mapping*** on $[a, b]$. Therefore, Theorem 2.4 essentially

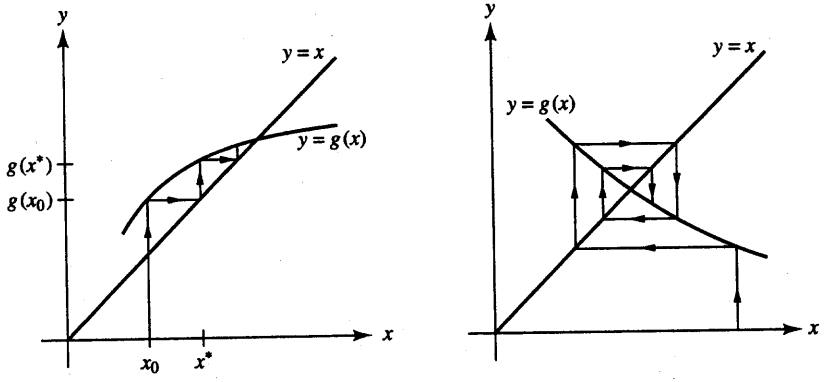


Figure 2.10 Convergence of fixed-point iteration.