Numerical Analysis

RICHARD L. BURDEN

DOUGLAS J. FAIRES

ANNETTE M. BURDEN

10E

Numerical Analysis

Numerical Analysis

TENTH EDITION

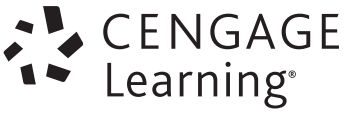
Richard L. Burden
Youngstown University

J. Douglas Faires
Youngstown University

Annette M. Burden
Youngstown University

CENGAGE Learning®

Australia • Brazil • Mexico • Singapore • United Kingdom • United States



Numerical Analysis,
Tenth Edition
Richard L. Burden, J. Douglas Faires,
Annette M. Burden

Product Director: *Terence Boyle*
Senior Product Team Manager: *Richard Stratton*
Associate Content Developer: *Spencer Arritt*
Product Assistant: *Kathryn Schrumpf*
Market Development Manager: *Julie Schuster*
Content Project Manager: *Jill Quinn*
Senior Art Director: *Linda May*
Manufacturing Planner: *Doug Bertke*
IP Analyst: *Christina Ciaramella*
IP Project Manager: *John Sarantakis*
Production Service: *Cenveo Publisher Services*
Compositor: *Cenveo Publisher Services*
Cover Image: © *agsandrew/Shutterstock.com*

Library of Congress Control Number: 2014949816

ISBN: 978-1-305-25366-7

Cengage Learning
20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at **www.cengage.com/global**.

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

To learn more about Cengage Learning Solutions, visit **www.cengage.com**. Purchase any of our products at your local college store or at our preferred online store **www.cengagebrain.com**.

Printed in the United States of America
Print Number: 01 Print Year: 2014

This edition is dedicated to the memory of
J. Douglas Faires
Doug was a friend, colleague, and coauthor for over 40 years.
He will be sadly missed.

Contents

Preface

About the Text

This text was written for a sequence of courses on the theory and application of numerical approximation techniques. It is designed primarily for junior-level mathematics, science, and engineering majors who have completed at least the first year of the standard college calculus sequence. Familiarity with the fundamentals of matrix algebra and differential equations is useful, but there is sufficient introductory material on these topics so that courses in these subjects are not needed as prerequisites.

Previous editions of *Numerical Analysis* have been used in a wide variety of situations. In some cases, the mathematical analysis underlying the development of approximation techniques was given more emphasis than the methods; in others, the emphasis was reversed. The book has been used as a core reference for beginning graduate-level courses in engineering, mathematics, computer science programs, and in first-year courses in introductory analysis offered at international universities. We have adapted the book to fit these diverse users without compromising our original purpose:

> *To introduce modern approximation techniques; to explain how, why, and when they can be expected to work; and to provide a foundation for further study of numerical analysis and scientific computing.*

The book contains sufficient material for at least a full year of study, but we expect many people will use the text only for a single-term course. In such a single-term course, students learn to identify the types of problems that require numerical techniques for their solution and see examples of the error propagation that can occur when numerical methods are applied. They accurately approximate the solution of problems that cannot be solved exactly and learn typical techniques for estimating error bounds for their approximations. The remainder of the text then serves as a reference for methods that are not considered in the course. Either the full-year or the single-course treatment is consistent with the philosophy of the text.

Virtually every concept in the text is illustrated by example, and this edition contains more than 2500 class-tested exercises ranging from elementary applications of methods and algorithms to generalizations and extensions of the theory. In addition, the exercise sets include numerous applied problems from diverse areas of engineering as well as from the physical, computer, biological, economic, and social sciences. The applications, chosen clearly and concisely, demonstrate how numerical techniques can and often must be applied in real-life situations.

A number of software packages, known as Computer Algebra Systems (CAS), have been developed to produce symbolic mathematical computations. Maple©, Mathematica©, and MATLAB© are predominant among these in the academic environment. Student versions of these software packages are available at reasonable prices for most common

computer systems. In addition, Sage, a free open source system, is now available. Information about this system can be found at the site

http://www.sagemath.org

Although there are differences among the packages, both in performance and in price, all can perform standard algebra and calculus operations.

The results in most of our examples and exercises have been generated using problems for which exact values *can* be determined because this better permits the performance of the approximation method to be monitored. In addition, for many numerical techniques, the error analysis requires bounding a higher ordinary or partial derivative of a function, which can be a tedious task and one that is not particularly instructive once the techniques of calculus have been mastered. So having a symbolic computation package available can be very useful in the study of approximation techniques because exact solutions can often be obtained easily using symbolic computation. Derivatives can be quickly obtained symbolically, and a little insight often permits a symbolic computation to aid in the bounding process as well.

Algorithms and Programs

In our first edition, we introduced a feature that at the time was innovative and somewhat controversial. Instead of presenting our approximation techniques in a specific programming language (FORTRAN was dominant at the time), we gave algorithms in a pseudocode that would lead to a well-structured program in a variety of languages. Beginning with the second edition, we listed programs in specific languages in the *Instructor's Manual* for the book, and the number of these languages increased in subsequent editions. We now have the programs coded and available online in most common programming languages and CAS worksheets. All of these are on the companion website for the book (see "Supplements").

For each algorithm, there is a program written in Fortran, Pascal, C, and Java. In addition, we have coded the programs using Maple, Mathematica, and MATLAB. This should ensure that a set of programs is available for most common computing systems.

Every program is illustrated with a sample problem that is closely correlated to the text. This permits the program to be run initially in the language of your choice to see the form of the input and output. The programs can then be modified for other problems by making minor changes. The form of the input and output are, as nearly as possible, the same in each of the programming systems. This permits an instructor using the programs to discuss them generically without regard to the particular programming system an individual student uses.

The programs are designed to run on a minimally configured computer and given in ASCII format to permit flexibility of use. This permits them to be altered using any editor or word processor that creates standard ASCII files. (These are also commonly called "text-only" files.) Extensive README files are included with the program files so that the peculiarities of the various programming systems can be individually addressed. The README files are presented both in ASCII format and as PDF files. As new software is developed, the programs will be updated and placed on the website for the book.

For most of the programming systems, the appropriate software is needed, such as a compiler for Pascal, Fortran, and C, or one of the computer algebra systems (Maple, Mathematica, and MATLAB). The Java implementations are an exception. You need the system to run the programs, but Java can be freely downloaded from various sites. The best way to obtain Java is to use a search engine to search on the name, choose a download site, and follow the instructions for that site.

New for This Edition

The first edition of this book was published more than 35 years ago, in the decade after major advances in numerical techniques were made to reflect the new widespread availability of computer equipment. In our revisions of the book, we have added new techniques in an attempt to keep our treatment current. To continue this trend, we have made a number of significant changes for this edition:

- Some of the examples in the book have been rewritten to better emphasize the problem being solved before the solution is given. Additional steps have been added to some of the examples to explicitly show the computations required for the first steps of iteration processes. This gives readers a way to test and debug programs they have written for problems similar to the examples.
- Chapter exercises have been split into computational, applied, and theoretical to give the instructor more flexibility in assigning homework. In almost all of the computational situations, the exercises have been paired in an odd-even manner. Since the odd problems are answered in the back of the text, if even problems were assigned as homework, students could work the odd problems and check their answers prior to doing the even problem.
- Many new applied exercises have been added to the text.
- Discussion questions have been added after each chapter section primarily for instructor use in online courses.
- The last section of each chapter has been renamed and split into four subsections: Numerical Software, Discussion Questions, Key Concepts, and Chapter Review. Many of these discussion questions point the student to modern areas of research in software development.
- Parts of the text were reorganized to facilitate online instruction.
- Additional PowerPoints have been added to supplement the reading material.
- The bibliographic material has been updated to reflect new editions of books that we reference. New sources have been added that were not previously available.

As always with our revisions, every sentence was examined to determine if it was phrased in a manner that best relates what we are trying to describe.

Supplements

The authors have created a companion website containing the supplementary materials listed below. The website located at

https://sites.google.com/site/numericalanalysis1burden/

is for students and instructors. Some material on the website is for instructor use only. Instructors can access protected materials by contacting the authors for the password.

Some of the supplements can also be obtained at

https://www.cengagebrain.com

by searching the ISBN.

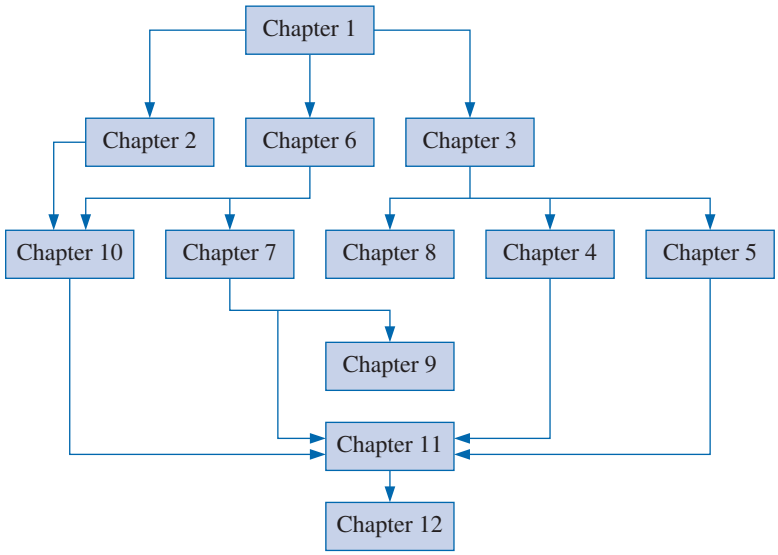
1. *Student Program Examples* that contain Maple, Matlab, and Excel code for student use in solving text problems. This is organized to parallel the text chapter by chapter. Commands in these systems are illustrated. The commands are presented in very short program segments to show how exercises may be solved without extensive programming.
2. *Student Lectures* that contain additional insight to the chapter content. These lectures were written primarily for the online learner but can be useful to students taking the course in a traditional setting.
3. *Student Study Guide* that contains worked-out solutions to many of the problems. The first two chapters of this guide are available on the website for the book in PDF format so that prospective users can tell if they find it sufficiently useful. The entire guide can be obtained only from the publisher by calling Cengage Learning Customer & Sales Support at 1-800-354-9706 or by ordering online at http://www.cengagebrain.com/.
4. *Algorithm Programs* that are complete programs written in Maple, Matlab, Mathematica, C, Pascal, Fortran, and Java for all the algorithms in the text. These programs are intended for students who are more experienced with programming languages.
5. *Instructor PowerPoints* in PDF format for instructor use in both traditional and online courses. Contact authors for password.
6. *Instructor's Manual* that provides answers and solutions to all the exercises in the book. Computation results in the *Instructor's Manual* were regenerated for this edition using the programs on the website to ensure compatibility among the various programming systems. Contact authors for password.
7. *Instructor Sample Tests* for instructor use. Contact authors for password.
8. *Errata*.

Possible Course Suggestions

Numerical Analysis is designed to allow instructors flexibility in the choice of topics as well as in the level of theoretical rigor and in the emphasis on applications. In line with these aims, we provide detailed references for the results that are not demonstrated in the text and for the applications that are used to indicate the practical importance of the methods. The text references cited are those most likely to be available in college libraries and have been updated to reflect recent editions. We also include quotations from original research papers when we feel this material is accessible to our intended audience. All referenced material has been indexed to the appropriate locations in the text, and Library of Congress call information for reference material has been included to permit easy location if searching for library material.

The following flowchart indicates chapter prerequisites. Most of the possible sequences that can be generated from this chart have been taught by the authors at Youngstown State University.

The material in this edition should permit instructors to prepare an undergraduate course in numerical linear algebra for students who have not previously studied numerical analysis. This could be done by covering Chapters 1, 6, 7, and 9.



Acknowledgments

We have been fortunate to have had many of our students and colleagues give us their impressions of earlier editions of this book, and we take all of these comments very seriously. We have tried to include all the suggestions that complement the philosophy of the book and are extremely grateful to all those that have taken the time to contact us about ways to improve subsequent versions.

We would particularly like to thank the following, whose suggestions we have used in this and previous editions.

Douglas Carter,

John Carroll, Dublin University

Yavuz Duman, T.C. Istanbul Kultur Universitesi

Neil Goldman,

Christopher Harrison,

Teryn Jones, Youngstown State University

Aleksandar Samardzic, University of Belgrade

Mikhail M. Shvartsman, University of St. Thomas

Dennis C. Smolarski, Santa Clara University

Dale Smith, Comcast

We would like to thank Dr. Barbara T. Faires for her cooperation in providing us with materials that we needed to make this revision possible. Her graciousness during such a difficult time was greatly appreciated.

As has been our practice in past editions of the book, we have used student help at Youngstown State University in preparing the tenth edition. Our able assistant for this edition was Teryn Jones who worked on the Java applets. We would like to thank Edward R. Burden, an Electrical Engineering doctoral student at Ohio State University, who has been checking all the application problems and new material in the text. We also would like to express gratitude to our colleagues on the faculty and administration of Youngstown State University for providing us the opportunity and facilities to complete this project.

We would like to thank some people who have made significant contributions to the history of numerical methods. Herman H. Goldstine has written an excellent book titled *A History of Numerical Analysis from the 16th Through the 19th Century* [Golds]. Another source of excellent historical mathematical knowledge is the MacTutor History of Mathematics archive at the University of St. Andrews in Scotland. It has been created by John J. O'Connor and Edmund F. Robertson and has the Internet address

http://www-gap.dcs.st-and.ac.uk/~history/

An incredible amount of work has gone into creating the material on this site, and we have found the information to be unfailingly accurate. Finally, thanks to all the contributors to Wikipedia who have added their expertise to that site so that others can benefit from their knowledge.

In closing, thanks again to those who have spent the time and effort to contact us over the years. It has been wonderful to hear from so many students and faculty who used our book for their first exposure to the study of numerical methods. We hope this edition continues this exchange and adds to the enjoyment of students studying numerical analysis. If you have any suggestions for improving future editions of the book, we would, as always, be grateful for your comments. We can be contacted most easily by e-mail at the addresses listed below.

Richard L. Burden
rlburden@ysu.edu

Annette M. Burden
amburden@ysu.edu

CHAPTER

1

Mathematical Preliminaries and Error Analysis

Introduction

In beginning chemistry courses, we see the *ideal gas law*,

$$PV = NRT,$$

which relates the pressure P, volume V, temperature T, and number of moles N of an "ideal" gas. In this equation, R is a constant that depends on the measurement system.

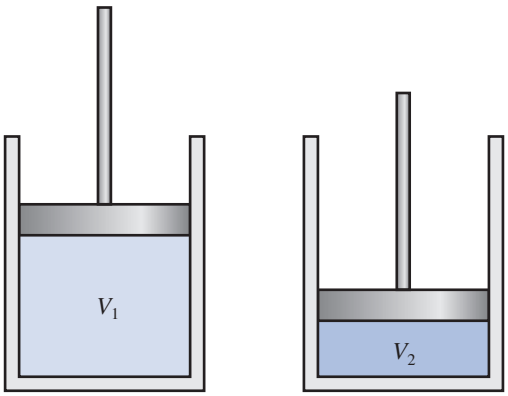
Suppose two experiments are conducted to test this law, using the same gas in each case. In the first experiment,

$$P = 1.00 \text{ atm}, \qquad V = 0.100 \text{ m}^3,$$
$$N = 0.00420 \text{ mol}, \qquad R = 0.08206.$$

The ideal gas law predicts the temperature of the gas to be

$$T = \frac{PV}{NR} = \frac{(1.00)(0.100)}{(0.00420)(0.08206)} = 290.15 \text{ K} = 17^\circ\text{C}.$$

When we measure the temperature of the gas, however, we find that the true temperature is 15°C.



We then repeat the experiment using the same values of R and N but increase the pressure by a factor of two and reduce the volume by the same factor. The product PV remains the same, so the predicted temperature is still 17°C. But now we find that the actual temperature of the gas is 19°C.

Clearly, the ideal gas law is suspect, but before concluding that the law is invalid in this situation, we should examine the data to see whether the error could be attributed to the experimental results. If so, we might be able to determine how much more accurate our experimental results would need to be to ensure that an error of this magnitude does not occur.

Analysis of the error involved in calculations is an important topic in numerical analysis and is introduced in Section 1.2. This particular application is considered in Exercise 26 of that section.

This chapter contains a short review of those topics from single-variable calculus that will be needed in later chapters. A solid knowledge of calculus is essential for an understanding of the analysis of numerical techniques, and more thorough review might be needed for those who have been away from this subject for a while. In addition there is an introduction to convergence, error analysis, the machine representation of numbers, and some techniques for categorizing and minimizing computational error.

1.1 Review of Calculus

Limits and Continuity

The concepts of *limit* and *continuity* of a function are fundamental to the study of calculus and form the basis for the analysis of numerical techniques.

Definition 1.1 A function f defined on a set X of real numbers has the **limit** L at x_0, written

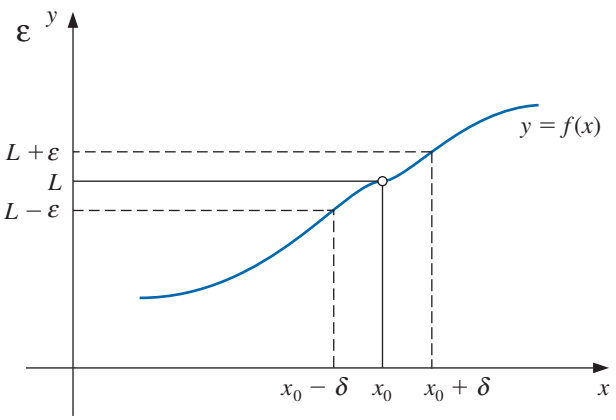
$$\lim_{x \to x_0} f(x) = L,$$

if, given any real number $\varepsilon > 0$, there exists a real number $\delta > 0$ such that

$$|f(x) - L| < \varepsilon, \quad \text{whenever} \quad x \in X \quad \text{and} \quad 0 < |x - x_0| < \delta.$$

(See Figure 1.1.) ■

Figure 1.1



Definition 1.2 Let f be a function defined on a set X of real numbers and $x_0 \in X$. Then f is **continuous** at x_0 if

$$\lim_{x \to x_0} f(x) = f(x_0).$$

The function f is **continuous on the set** X if it is continuous at each number in X. ■

The basic concepts of calculus and its applications were developed in the late 17th and early 18th centuries, but the mathematically precise concepts of limits and continuity were not described until the time of Augustin Louis Cauchy (1789–1857), Heinrich Eduard Heine (1821–1881), and Karl Weierstrass (1815 –1897) in the latter portion of the 19th century.

The set of all functions that are continuous on the set X is denoted $C(X)$. When X is an interval of the real line, the parentheses in this notation are omitted. For example, the set of all functions continuous on the closed interval $[a, b]$ is denoted $C[a, b]$. The symbol $\mathbb{R}$ denotes the set of all real numbers, which also has the interval notation $(-\infty, \infty)$. So the set of all functions that are continuous at every real number is denoted by $C(\mathbb{R})$ or by $C(-\infty, \infty)$.

The *limit of a sequence* of real or complex numbers is defined in a similar manner.

Definition 1.3 Let $\{x_n\}_{n=1}^{\infty}$ be an infinite sequence of real numbers. This sequence has the **limit** x (**converges to** x) if, for any $\varepsilon > 0$, there exists a positive integer $N(\varepsilon)$ such that $|x_n - x| < \varepsilon$ whenever $n > N(\varepsilon)$. The notation

$$\lim_{n \to \infty} x_n = x, \quad \text{or} \quad x_n \to x \quad \text{as} \quad n \to \infty,$$

means that the sequence $\{x_n\}_{n=1}^{\infty}$ converges to x. ■

Theorem 1.4 If f is a function defined on a set X of real numbers and $x_0 \in X$, then the following statements are equivalent:

a. f is continuous at x_0;

b. If $\{x_n\}_{n=1}^{\infty}$ is any sequence in X converging to x_0, then $\lim_{n \to \infty} f(x_n) = f(x_0)$.

■

The functions we will consider when discussing numerical methods will be assumed to be continuous because this is a minimal requirement for predictable behavior. Functions that are not continuous can skip over points of interest, which can cause difficulties in attempts to approximate a solution to a problem.

Differentiability

More sophisticated assumptions about a function generally lead to better approximation results. For example, a function with a smooth graph will normally behave more predictably than one with numerous jagged features. The smoothness condition relies on the concept of the derivative.

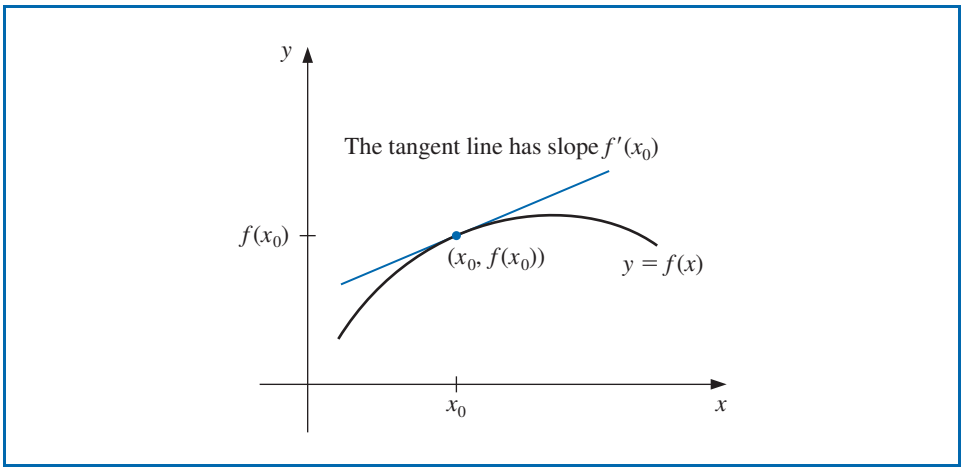
Definition 1.5 Let f be a function defined in an open interval containing x_0. The function f is **differentiable** at x_0 if

$$f'(x_0) = \lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}$$

exists. The number $f'(x_0)$ is called the **derivative** of f at x_0. A function that has a derivative at each number in a set X is **differentiable on** X. ■

The derivative of f at x_0 is the slope of the tangent line to the graph of f at $(x_0, f(x_0))$, as shown in Figure 1.2.

Figure 1.2



Theorem 1.6 If the function f is differentiable at x_0, then f is continuous at x_0. ■

The theorem attributed to Michel Rolle (1652–1719) appeared in 1691 in a little-known treatise titled *Méthode pour résoundre les égalites*. Rolle originally criticized the calculus that was developed by Isaac Newton and Gottfried Leibniz but later became one of its proponents.

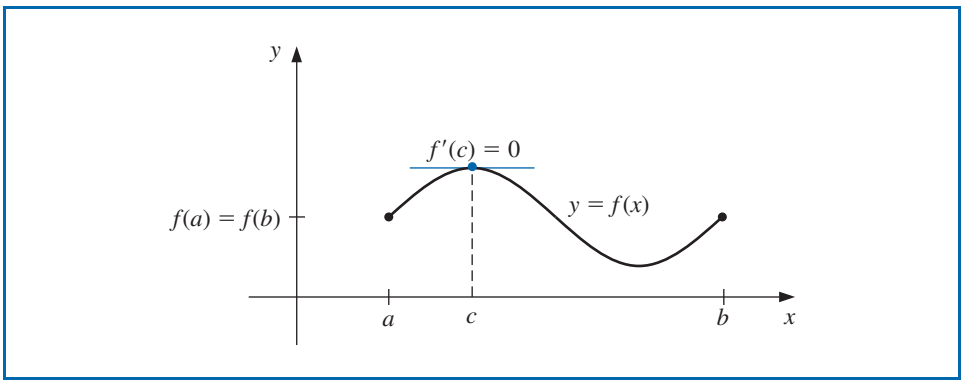
The next theorems are of fundamental importance in deriving methods for error estimation. The proofs of these theorems and the other unreferenced results in this section can be found in any standard calculus text.

The set of all functions that have n continuous derivatives on X is denoted $C^n(X)$, and the set of functions that have derivatives of all orders on X is denoted $C^\infty(X)$. Polynomial, rational, trigonometric, exponential, and logarithmic functions are in $C^\infty(X)$, where X consists of all numbers for which the functions are defined. When X is an interval of the real line, we will again omit the parentheses in this notation.

Theorem 1.7 **(Rolle's Theorem)**

Suppose $f \in C[a, b]$ and f is differentiable on (a, b). If $f(a) = f(b)$, then a number c in (a, b) exists with $f'(c) = 0$. (See Figure 1.3.) ■

Figure 1.3



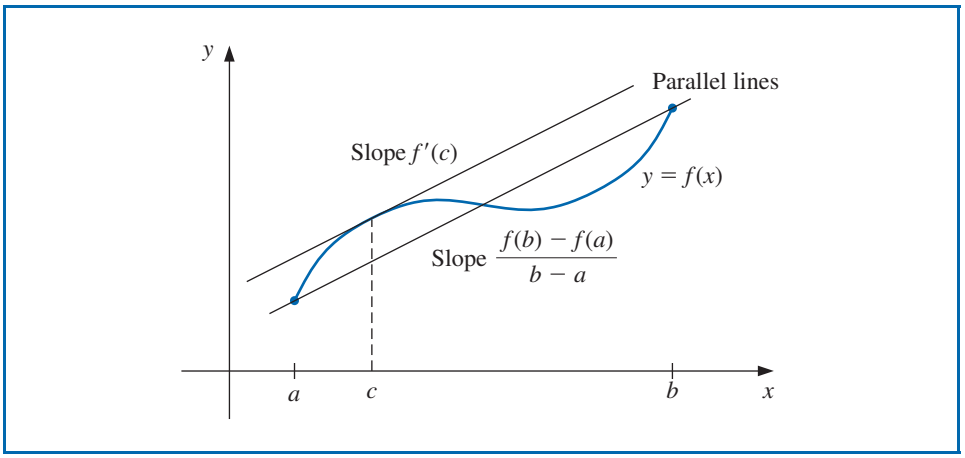
Theorem 1.8 **(Mean Value Theorem)**

If $f \in C[a, b]$ and f is differentiable on (a, b), then a number c in (a, b) exists with (See Figure 1.4.)

$$f'(c) = \frac{f(b) - f(a)}{b - a}.$$

■

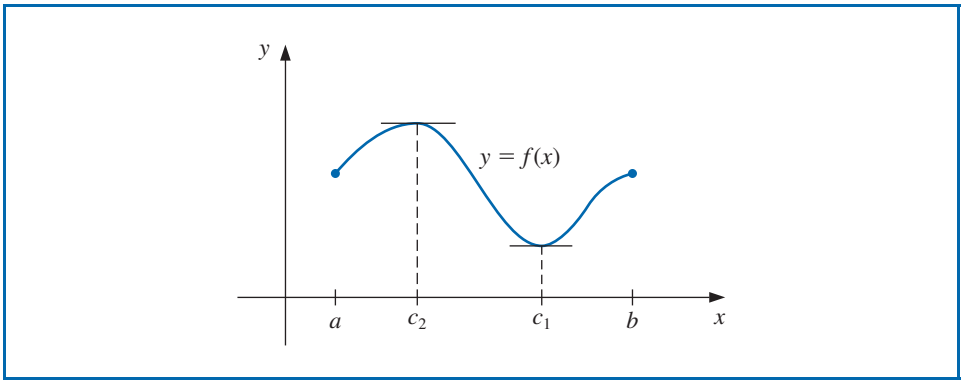
Figure 1.4



Theorem 1.9 **(Extreme Value Theorem)**

If $f \in C[a, b]$, then $c_1, c_2 \in [a, b]$ exist with $f(c_1) \leq f(x) \leq f(c_2)$, for all $x \in [a, b]$. In addition, if f is differentiable on (a, b), then the numbers c_1 and c_2 occur either at the endpoints of $[a, b]$ or where f' is zero. (See Figure 1.5.) ■

Figure 1.5



Example 1 Find the absolute minimum and absolute maximum values of

$$f(x) = 2 - e^x + 2x$$

on the intervals **(a)** $[0, 1]$, and **(b)** $[1, 2]$.

Solution We begin by differentiating $f(x)$ to obtain

$$f'(x) = -e^x + 2.$$

$f'(x) = 0$ when $-e^x + 2 = 0$ or, equivalently, when $e^x = 2$. Taking the natural logarithm of both sides of the equation gives

$$\ln(e^x) = \ln(2) \text{ or } x = \ln(2) \approx 0.69314718056$$

(a) When the interval is $[0, 1]$, the absolute extrema must occur at $f(0)$, $f(\ln(2))$, or $f(1)$. Evaluating, we have

$$f(0) = 2 - e^0 + 2(0) = 1$$

$$f(\ln(2)) = 2 - e^{\ln(2)} + 2\ln(2) = 2\ln(2) \approx 1.38629436112$$

$$f(1) = 2 - e + 2(1) = 4 - e \approx 1.28171817154.$$

Thus, the absolute minimum of $f(x)$ on $[0, 1]$ is $f(0) = 1$ and the absolute maximum is $f(\ln(2)) = 2\ln(2)$.

(b) When the interval is $[1, 2]$, we know that $f'(x) \neq 0$ so the absolute extrema occur at $f(1)$ and $f(2)$. Thus, $f(2) = 2 - e^2 + 2(2) = 6 - e^2 \approx -1.3890560983$. The absolute minimum on $[1, 2]$ is $6 - e^2$ and the absolute maximum is 1.

We note that

$$\max_{0 \le x \le 2} |f(x)| = |6 - e^2| \approx 1.3890560983.$$ ■

The following theorem is not generally presented in a basic calculus course but is derived by applying Rolle's Theorem successively to f, f', . . . , and, finally, to $f^{(n-1)}$. This result is considered in Exercise 26.

Theorem 1.10 (Generalized Rolle's Theorem)

Suppose $f \in C[a, b]$ is n times differentiable on (a, b). If $f(x) = 0$ at the $n + 1$ distinct numbers $a \le x_0 < x_1 < \ldots < x_n \le b$, then a number c in (x_0, x_n) and hence in (a, b) exists with $f^{(n)}(c) = 0$. ■

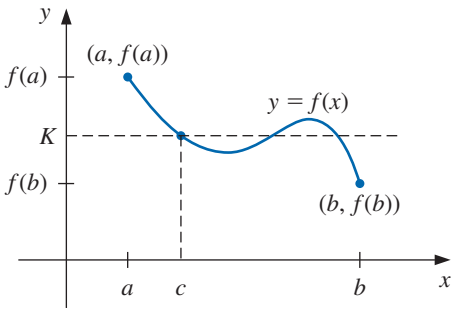
We will also make frequent use of the Intermediate Value Theorem. Although its statement seems reasonable, its proof is beyond the scope of the usual calculus course. It can, however, be found in most analysis texts (see, for example, [Fu], p. 67).

Theorem 1.11 (Intermediate Value Theorem)

If $f \in C[a, b]$ and K is any number between $f(a)$ and $f(b)$, then there exists a number c in (a, b) for which $f(c) = K$. ■

Figure 1.6 shows one choice for the number that is guaranteed by the Intermediate Value Theorem. In this example, there are two other possibilities.

Figure 1.6



Example 2 Show that $x^5 - 2x^3 + 3x^2 - 1 = 0$ has a solution in the interval $[0, 1]$.

Solution Consider the function defined by $f(x) = x^5 - 2x^3 + 3x^2 - 1$. The function f is continuous on $[0, 1]$. In addition,

$$f(0) = -1 < 0 \qquad \text{and} \qquad 0 < 1 = f(1).$$

Hence, the Intermediate Value Theorem implies that a number c exists, with $0 < c < 1$, for which $c^5 - 2c^3 + 3c^2 - 1 = 0$. ■

As seen in Example 2, the Intermediate Value Theorem is used to determine when solutions to certain problems exist. It does not, however, give an efficient means for finding these solutions. This topic is considered in Chapter 2.

Integration

The other basic concept of calculus that will be used extensively is the Riemann integral.

Definition 1.12 The **Riemann integral** of the function f on the interval $[a, b]$ is the following limit, provided it exists:

$$\int_a^b f(x)\,dx = \lim_{\max \Delta x_i \to 0} \sum_{i=1}^{n} f(z_i)\,\Delta x_i,$$

where the numbers $x_0, x_1, \ldots, x_n$ satisfy $a = x_0 \le x_1 \le \cdots \le x_n = b$, where $\Delta x_i = x_i - x_{i-1}$, for each $i = 1, 2, \ldots, n$, and z_i is arbitrarily chosen in the interval $[x_{i-1}, x_i]$. ■

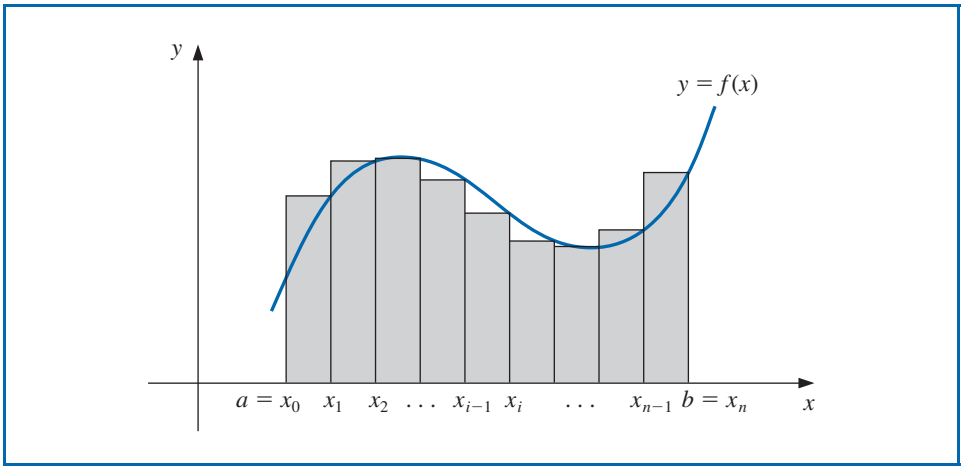
George Fredrich Berhard Riemann (1826–1866) made many of the important discoveries classifying the functions that have integrals. He also did fundamental work in geometry and complex function theory and is regarded as one of the profound mathematicians of the 19th century.

A function f that is continuous on an interval $[a, b]$ is also Riemann integrable on $[a, b]$. This permits us to choose, for computational convenience, the points x_i to be equally spaced in $[a, b]$ and, for each $i = 1, 2, \ldots, n$, to choose $z_i = x_i$. In this case,

$$\int_a^b f(x)\,dx = \lim_{n \to \infty} \frac{b-a}{n} \sum_{i=1}^{n} f(x_i),$$

where the numbers shown in Figure 1.7 as x_i are $x_i = a + i(b - a)/n$.

Figure 1.7



Two other results will be needed in our study of numerical analysis. The first is a generalization of the usual Mean Value Theorem for Integrals.

Theorem 1.13 **(Weighted Mean Value Theorem for Integrals)**

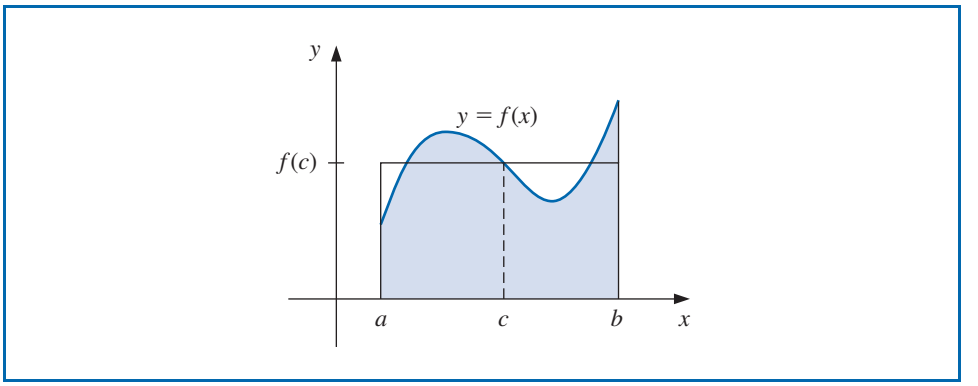
Suppose $f \in C[a, b]$, the Riemann integral of g exists on $[a, b]$, and $g(x)$ does not change sign on $[a, b]$. Then there exists a number c in (a, b) with

$$\int_a^b f(x)g(x)\,dx = f(c)\int_a^b g(x)\,dx.$$ ■

When $g(x) \equiv 1$, Theorem 1.13 is the usual Mean Value Theorem for Integrals. It gives the **average value** of the function f over the interval $[a, b]$ as (See Figure 1.8.)

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

Figure 1.8



The proof of Theorem 1.13 is not generally given in a basic calculus course but can be found in most analysis texts (see, for example, [Fu], p. 162).

Taylor Polynomials and Series

The final theorem in this review from calculus describes the Taylor polynomials. These polynomials are used extensively in numerical analysis.

Theorem 1.14 **(Taylor's Theorem)**

Suppose $f \in C^n[a, b]$, $f^{(n+1)}$ exists on $[a, b]$, and $x_0 \in [a, b]$. For every $x \in [a, b]$, there exists a number $\xi(x)$ between x_0 and x with

$$f(x) = P_n(x) + R_n(x),$$

where

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

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

Brook Taylor (1685–1731) described this series in 1715 in the paper *Methodus incrementorum directa et inversa*. Special cases of the result and likely the result itself had been previously known to Isaac Newton, James Gregory, and others.

and

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

■

Colin Maclaurin (1698–1746) is best known as the defender of the calculus of Newton when it came under bitter attack by Irish philosopher Bishop George Berkeley.

Maclaurin did not discover the series that bears his name; it was known to century mathematicians before he was born. However, he did devise a method for solving a system of linear equations that is known as Cramer's rule, which Cramer did not publish until 1750.

Here $P_n(x)$ is called the ***n*th Taylor polynomial** for f about x_0, and $R_n(x)$ is called the **remainder term** (or **truncation error**) associated with $P_n(x)$. Since the number $\xi(x)$ in the truncation error $R_n(x)$ depends on the value of x at which the polynomial $P_n(x)$ is being evaluated, it is a function of the variable x. However, we should not expect to be able to explicitly determine the function $\xi(x)$. Taylor's Theorem simply ensures that such a function exists and that its value lies between x and x_0. In fact, one of the common problems in numerical methods is to try to determine a realistic bound for the value of $f^{(n+1)}(\xi(x))$ when x is in some specified interval.

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. In the case $x_0 = 0$, the Taylor polynomial is often called a **Maclaurin polynomial**, and the Taylor series is often called a **Maclaurin series**.

The term **truncation error** in the Taylor polynomial refers to the error involved in using a truncated, or finite, summation to approximate the sum of an infinite series.

Example 3 Let $f(x) = \cos x$ and $x_0 = 0$. Determine

(a) the second Taylor polynomial for f about x_0; and

(b) the third Taylor polynomial for f about x_0.

Solution Since $f \in C^\infty(\mathbb{R})$, Taylor's Theorem can be applied for any $n \geq 0$. Also,

$$f'(x) = -\sin x, \quad f''(x) = -\cos x, \quad f'''(x) = \sin x, \quad \text{and} \quad f^{(4)}(x) = \cos x,$$

so

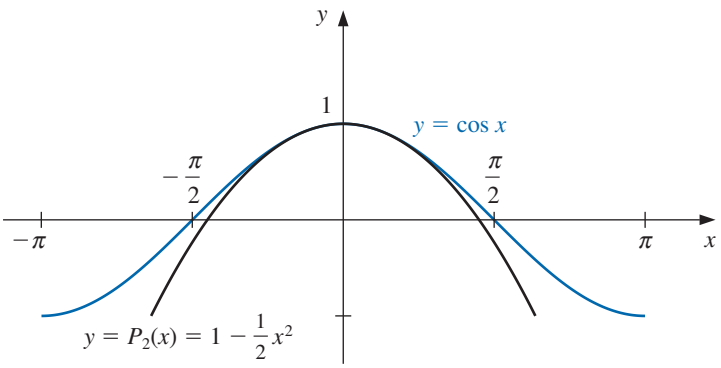
$$f(0) = 1, \quad f'(0) = 0, \quad f''(0) = -1, \quad \text{and} \quad f'''(0) = 0.$$

(a) For $n = 2$ and $x_0 = 0$, we have

$$\cos x = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f'''(\xi(x))}{3!}x^3$$
$$= 1 - \frac{1}{2}x^2 + \frac{1}{6}x^3 \sin \xi(x),$$

where $\xi(x)$ is some (generally unknown) number between 0 and x. (See Figure 1.9.)

Figure 1.9



When $x = 0.01$, this becomes

$$\cos 0.01 = 1 - \frac{1}{2}(0.01)^2 + \frac{1}{6}(0.01)^3 \sin \xi(0.01) = 0.99995 + \frac{10^{-6}}{6} \sin \xi(0.01).$$

The approximation to $\cos 0.01$ given by the Taylor polynomial is therefore 0.99995. The truncation error, or remainder term, associated with this approximation is

$$\frac{10^{-6}}{6} \sin \xi(0.01) = 0.1\overline{6} \times 10^{-6} \sin \xi(0.01),$$

where the bar over the 6 in $0.1\overline{6}$ is used to indicate that this digit repeats indefinitely. Although we have no way of determining $\sin \xi(0.01)$, we know that all values of the sine lie in the interval $[-1, 1]$, so the error occurring if we use the approximation 0.99995 for the value of $\cos 0.01$ is bounded by

$$|\cos(0.01) - 0.99995| = 0.1\overline{6} \times 10^{-6} |\sin \xi(0.01)| \leq 0.1\overline{6} \times 10^{-6}.$$

Hence, the approximation 0.99995 matches at least the first five digits of $\cos 0.01$, and

$$\begin{aligned} 0.9999483 < 0.99995 - 1.\overline{6} \times 10^{-6} &\leq \cos 0.01 \\ &\leq 0.99995 + 1.\overline{6} \times 10^{-6} < 0.9999517. \end{aligned}$$

The error bound is much larger than the actual error. This is due in part to the poor bound we used for $|\sin \xi(x)|$. It is shown in Exercise 27 that for all values of x, we have $|\sin x| \leq |x|$. Since $0 \leq \xi < 0.01$, we could have used the fact that $|\sin \xi(x)| \leq 0.01$ in the error formula, producing the bound $0.1\overline{6} \times 10^{-8}$.

(b) Since $f'''(0) = 0$, the third Taylor polynomial with remainder term about $x_0 = 0$ is

$$\cos x = 1 - \frac{1}{2}x^2 + \frac{1}{24}x^4 \cos \tilde{\xi}(x),$$

where $0 < \tilde{\xi}(x) < 0.01$. The approximating polynomial remains the same, and the approximation is still 0.99995, but we now have much better accuracy assurance. Since $|\cos \tilde{\xi}(x)| \leq 1$ for all x, we have

$$\left| \frac{1}{24}x^4 \cos \tilde{\xi}(x) \right| \leq \frac{1}{24}(0.01)^4(1) \approx 4.2 \times 10^{-10}.$$

So,

$$|\cos 0.01 - 0.99995| \leq 4.2 \times 10^{-10},$$

and

$$\begin{aligned} 0.99994999958 &= 0.99995 - 4.2 \times 10^{-10} \\ &\leq \cos 0.01 \leq 0.99995 + 4.2 \times 10^{-10} = 0.99995000042. \end{aligned}$$

■

Example 3 illustrates the two objectives of numerical analysis:

(i) Find an approximation to the solution of a given problem.

(ii) Determine a bound for the accuracy of the approximation.

The Taylor polynomials in both parts provide the same answer to (i), but the third Taylor polynomial gave a much better answer to (ii) than the second Taylor polynomial.

We can also use the Taylor polynomials to give us approximations to integrals.

Illustration We can use the third Taylor polynomial and its remainder term found in Example 3 to approximate $\int_0^{0.1} \cos x \, dx$. We have

$$\begin{aligned}\int_0^{0.1} \cos x \, dx &= \int_0^{0.1} \left(1 - \frac{1}{2}x^2\right) dx + \frac{1}{24}\int_0^{0.1} x^4 \cos\tilde{\xi}(x) \, dx \\ &= \left[x - \frac{1}{6}x^3\right]_0^{0.1} + \frac{1}{24}\int_0^{0.1} x^4 \cos\tilde{\xi}(x) \, dx \\ &= 0.1 - \frac{1}{6}(0.1)^3 + \frac{1}{24}\int_0^{0.1} x^4 \cos\tilde{\xi}(x) \, dx.\end{aligned}$$

Therefore,

$$\int_0^{0.1} \cos x \, dx \approx 0.1 - \frac{1}{6}(0.1)^3 = 0.0998\overline{3}.$$

A bound for the error in this approximation is determined from the integral of the Taylor remainder term and the fact that $|\cos\tilde{\xi}(x)| \le 1$ for all x:

$$\begin{aligned}\frac{1}{24}\left|\int_0^{0.1} x^4 \cos\tilde{\xi}(x) \, dx\right| &\le \frac{1}{24}\int_0^{0.1} x^4 |\cos\tilde{\xi}(x)| \, dx \\ &\le \frac{1}{24}\int_0^{0.1} x^4 \, dx = \frac{(0.1)^5}{120} = 8.\overline{3} \times 10^{-8}.\end{aligned}$$

The true value of this integral is

$$\int_0^{0.1} \cos x \, dx = \sin x\Big]_0^{0.1} = \sin 0.1 \approx 0.099833416647,$$

so the actual error for this approximation is 8.3314×10^{-8}, which is within the error bound. ■

EXERCISE SET 1.1

1. Show that the following equations have at least one solution in the given intervals.

a. $x \cos x - 2x^2 + 3x - 1 = 0$, $[0.2, 0.3]$ and $[1.2, 1.3]$

b. $(x-2)^2 - \ln x = 0$, $[1, 2]$ and $[e, 4]$

c. $2x\cos(2x) - (x-2)^2 = 0$, $[2, 3]$ and $[3, 4]$

d. $x - (\ln x)^x = 0$, $[4, 5]$

2. Show that the following equations have at least one solution in the given intervals.

a. $\sqrt{x} - \cos x = 0$, $[0, 1]$

b. $e^x - x^2 + 3x - 2 = 0$, $[0, 1]$

c. $-3\tan(2x) + x = 0$, $[0, 1]$

d. $\ln x - x^2 + \frac{5}{2}x - 1 = 0$, $[\frac{1}{2}, 1]$

3. Find intervals containing solutions to the following equations.

a. $x - 2^{-x} = 0$

b. $2x\cos(2x) - (x+1)^2 = 0$

c. $3x - e^x = 0$

d. $x + 1 - 2\sin(\pi x) = 0$

4. Find intervals containing solutions to the following equations.

a. $x - 3^{-x} = 0$

b. $4x^2 - e^x = 0$

c. $x^3 - 2x^2 - 4x + 2 = 0$

d. $x^3 + 4.001x^2 + 4.002x + 1.101 = 0$

5. Find $\max_{a \le x \le b} |f(x)|$ for the following functions and intervals.

a. $f(x) = (2 - e^x + 2x)/3, \quad [0, 1]$

b. $f(x) = (4x - 3)/(x^2 - 2x), \quad [0.5, 1]$

c. $f(x) = 2x \cos(2x) - (x - 2)^2, \quad [2, 4]$

d. $f(x) = 1 + e^{-\cos(x-1)}, \quad [1, 2]$

6. Find $\max_{a \le x \le b} |f(x)|$ for the following functions and intervals.

a. $f(x) = 2x/(x^2 + 1), \quad [0, 2]$

b. $f(x) = x^2\sqrt{(4 - x)}, \quad [0, 4]$

c. $f(x) = x^3 - 4x + 2, \quad [1, 2]$

d. $f(x) = x\sqrt{(3 - x^2)}, \quad [0, 1]$

7. Show that $f'(x)$ is 0 at least once in the given intervals.

a. $f(x) = 1 - e^x + (e - 1)\sin((\pi/2)x), \quad [0, 1]$

b. $f(x) = (x - 1)\tan x + x \sin \pi x, \quad [0, 1]$

c. $f(x) = x \sin \pi x - (x - 2)\ln x, \quad [1, 2]$

d. $f(x) = (x - 2)\sin x \ln(x + 2), \quad [-1, 3]$

8. Suppose $f \in C[a, b]$ and $f'(x)$ exists on (a, b). Show that if $f'(x) \neq 0$ for all x in (a, b), then there can exist at most one number p in $[a, b]$ with $f(p) = 0$.

9. Let $f(x) = x^3$.

a. Find the second Taylor polynomial $P_2(x)$ about $x_0 = 0$.

b. Find $R_2(0.5)$ and the actual error in using $P_2(0.5)$ to approximate $f(0.5)$.

c. Repeat part (a) using $x_0 = 1$.

d. Repeat part (b) using the polynomial from part (c).

10. Find the third Taylor polynomial $P_3(x)$ for the function $f(x) = \sqrt{x + 1}$ about $x_0 = 0$. Approximate $\sqrt{0.5}$, $\sqrt{0.75}$, $\sqrt{1.25}$, and $\sqrt{1.5}$ using $P_3(x)$ and find the actual errors.

11. Find the second Taylor polynomial $P_2(x)$ for the function $f(x) = e^x \cos x$ about $x_0 = 0$.

a. Use $P_2(0.5)$ to approximate $f(0.5)$. Find an upper bound for error $|f(0.5) - P_2(0.5)|$ using the error formula and compare it to the actual error.

b. Find a bound for the error $|f(x) - P_2(x)|$ in using $P_2(x)$ to approximate $f(x)$ on the interval $[0, 1]$.

c. Approximate $\int_0^1 f(x)\,dx$ using $\int_0^1 P_2(x)\,dx$.

d. Find an upper bound for the error in (c) using $\int_0^1 |R_2(x)\,dx|$ and compare the bound to the actual error.

12. Repeat Exercise 11 using $x_0 = \pi/6$.

13. Find the third Taylor polynomial $P_3(x)$ for the function $f(x) = (x - 1)\ln x$ about $x_0 = 1$.

a. Use $P_3(0.5)$ to approximate $f(0.5)$. Find an upper bound for error $|f(0.5) - P_3(0.5)|$ using the error formula and compare it to the actual error.

b. Find a bound for the error $|f(x) - P_3(x)|$ in using $P_3(x)$ to approximate $f(x)$ on the interval $[0.5, 1.5]$.

c. Approximate $\int_{0.5}^{1.5} f(x)\,dx$ using $\int_{0.5}^{1.5} P_3(x)\,dx$.

d. Find an upper bound for the error in (c) using $\int_{0.5}^{1.5} |R_3(x)\,dx|$ and compare the bound to the actual error.

14. Let $f(x) = 2x \cos(2x) - (x - 2)^2$ and $x_0 = 0$.

a. Find the third Taylor polynomial $P_3(x)$ and use it to approximate $f(0.4)$.

b. Use the error formula in Taylor's Theorem to find an upper bound for the error $|f(0.4) - P_3(0.4)|$. Compute the actual error.

c. Find the fourth Taylor polynomial $P_4(x)$ and use it to approximate $f(0.4)$.

d. Use the error formula in Taylor's Theorem to find an upper bound for the error $|f(0.4) - P_4(0.4)|$. Compute the actual error.

15. Find the fourth Taylor polynomial $P_4(x)$ for the function $f(x) = xe^{x^2}$ about $x_0 = 0$.

a. Find an upper bound for $|f(x) - P_4(x)|$, for $0 \le x \le 0.4$.

b. Approximate $\int_0^{0.4} f(x)\,dx$ using $\int_0^{0.4} P_4(x)\,dx$.

c. Find an upper bound for the error in (b) using $\int_0^{0.4} P_4(x)\,dx$.

d. Approximate $f'(0.2)$ using $P_4'(0.2)$ and find the error.

16. Use the error term of a Taylor polynomial to estimate the error involved in using $\sin x \approx x$ to approximate $\sin 1^\circ$.

17. Use a Taylor polynomial about $\pi/4$ to approximate $\cos 42^\circ$ to an accuracy of 10^{-6}.

18. Let $f(x) = (1 - x)^{-1}$ and $x_0 = 0$. Find the nth Taylor polynomial $P_n(x)$ for $f(x)$ about x_0. Find a value of n necessary for $P_n(x)$ to approximate $f(x)$ to within 10^{-6} on $[0, 0.5]$.

19. Let $f(x) = e^x$ and $x_0 = 0$. Find the nth Taylor polynomial $P_n(x)$ for $f(x)$ about x_0. Find a value of n necessary for $P_n(x)$ to approximate $f(x)$ to within 10^{-6} on $[0, 0.5]$.

20. Find the nth Maclaurin polynomial $P_n(x)$ for $f(x) = \arctan x$.

21. The polynomial $P_2(x) = 1 - \frac{1}{2}x^2$ is to be used to approximate $f(x) = \cos x$ in $[-\frac{1}{2}, \frac{1}{2}]$. Find a bound for the maximum error.

22. Use the Intermediate Value Theorem 1.11 and Rolle's Theorem 1.7 to show that the graph of $f(x) = x^3 + 2x + k$ crosses the x-axis exactly once, regardless of the value of the constant k.

23. A Maclaurin polynomial for e^x is used to give the approximation 2.5 to e. The error bound in this approximation is established to be $E = \frac{1}{6}$. Find a bound for the error in E.

24. The *error function* defined by

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,dt$$

gives the probability that any one of a series of trials will lie within x units of the mean, assuming that the trials have a normal distribution with mean 0 and standard deviation $\sqrt{2}/2$. This integral cannot be evaluated in terms of elementary functions, so an approximating technique must be used.

a. Integrate the Maclaurin series for e^{-x^2} to show that

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)k!}.$$

b. The error function can also be expressed in the form

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} e^{-x^2} \sum_{k=0}^{\infty} \frac{2^k x^{2k+1}}{1 \cdot 3 \cdot 5 \cdots (2k+1)}.$$

Verify that the two series agree for $k = 1, 2, 3$, and 4. [*Hint:* Use the Maclaurin series for e^{-x^2}.]

c. Use the series in part (a) to approximate $\operatorname{erf}(1)$ to within 10^{-7}.

d. Use the same number of terms as in part (c) to approximate $\operatorname{erf}(1)$ with the series in part (b).

e. Explain why difficulties occur using the series in part (b) to approximate $\operatorname{erf}(x)$.

THEORETICAL EXERCISES

25. The nth Taylor polynomial for a function f at x_0 is sometimes referred to as the polynomial of degree at most n that "best" approximates f near x_0.

a. Explain why this description is accurate.

b. Find the quadratic polynomial that best approximates a function f near $x_0 = 1$ if the tangent line at $x_0 = 1$ has equation $y = 4x - 1$ and if $f''(1) = 6$.

26. Prove the Generalized Rolle's Theorem, Theorem 1.10, by verifying the following.

a. Use Rolle's Theorem to show that $f'(z_i) = 0$ for $n - 1$ numbers in $[a, b]$ with $a < z_1 < z_2 < \cdots < z_{n-1} < b$.

b. Use Rolle's Theorem to show that $f''(w_i) = 0$ for $n - 2$ numbers in $[a, b]$ with $z_1 < w_1 < z_2 < w_2 \cdots w_{n-2} < z_{n-1} < b$.

c. Continue the arguments in parts (a) and (b) to show that for each $j = 1, 2, \ldots, n - 1$, there are $n - j$ distinct numbers in $[a, b]$, where $f^{(j)}$ is 0.

d. Show that part (c) implies the conclusion of the theorem.

27. Example 3 stated that for all x we have $|\sin x| \leq |x|$. Use the following to verify this statement.

a. Show that for all $x \geq 0$, $f(x) = x - \sin x$ is nondecreasing, which implies that $\sin x \leq x$ with equality only when $x = 0$.

b. Use the fact that the sine function is odd to reach the conclusion.

28. A function $f : [a, b] \to \mathbb{R}$ is said to satisfy a *Lipschitz condition* with Lipschitz constant L on $[a, b]$ if, for every $x, y \in [a, b]$, we have $|f(x) - f(y)| \leq L|x - y|$.

a. Show that if f satisfies a Lipschitz condition with Lipschitz constant L on an interval $[a, b]$, then $f \in C[a, b]$.

b. Show that if f has a derivative that is bounded on $[a, b]$ by L, then f satisfies a Lipschitz condition with Lipschitz constant L on $[a, b]$.

c. Give an example of a function that is continuous on a closed interval but does not satisfy a Lipschitz condition on the interval.

29. Suppose $f \in C[a, b]$ and x_1 and x_2 are in $[a, b]$.

a. Show that a number ξ exists between x_1 and x_2 with

$$f(\xi) = \frac{f(x_1) + f(x_2)}{2} = \frac{1}{2}f(x_1) + \frac{1}{2}f(x_2).$$

b. Suppose c_1 and c_2 are positive constants. Show that a number ξ exists between x_1 and x_2 with

$$f(\xi) = \frac{c_1 f(x_1) + c_2 f(x_2)}{c_1 + c_2}.$$

c. Give an example to show that the result in part (b) does not necessarily hold when c_1 and c_2 have opposite signs with $c_1 \neq -c_2$.

30. Let $f \in C[a, b]$, and let p be in the open interval (a, b).

a. Suppose $f(p) \neq 0$. Show that a $\delta > 0$ exists with $f(x) \neq 0$, for all x in $[p - \delta, p + \delta]$, with $[p - \delta, p + \delta]$ a subset of $[a, b]$.

b. Suppose $f(p) = 0$ and $k > 0$ is given. Show that a $\delta > 0$ exists with $|f(x)| \leq k$, for all x in $[p - \delta, p + \delta]$, with $[p - \delta, p + \delta]$ a subset of $[a, b]$.

DISCUSSION QUESTION

1. In your own words, describe the Lipschitz condition. Give several examples of functions that satisfy this condition or give several examples of functions that do not satisfy this condition.

1.2 Round-off Errors and Computer Arithmetic

The arithmetic performed by a calculator or computer is different from the arithmetic in algebra and calculus courses. You would likely expect that we always have as true statements things such as $2+2 = 4, 4 \cdot 8 = 32$, and $(\sqrt{3})^2 = 3$. However, with *computer* arithmetic, we

expect exact results for $2+2=4$ and $4\cdot 8=32$, but we will not have precisely $(\sqrt{3})^2=3$. To understand why this is true, we must explore the world of finite-digit arithmetic.

In our traditional mathematical world, we permit numbers with an infinite number of digits. The arithmetic we use in this world *defines* $\sqrt{3}$ as that unique positive number that when multiplied by itself produces the integer 3. In the computational world, however, each representable number has only a fixed and finite number of digits. This means, for example, that only rational numbers—and not even all of these—can be represented exactly. Since $\sqrt{3}$ is not rational, it is given an approximate representation, one whose square will not be precisely 3, although it will likely be sufficiently close to 3 to be acceptable in most situations. In most cases, then, this machine arithmetic is satisfactory and passes without notice or concern, but at times problems arise because of this discrepancy.

Error due to rounding should be expected whenever computations are performed using numbers that are not powers of 2. Keeping this error under control is extremely important when the number of calculations is large.

The error that is produced when a calculator or computer is used to perform real-number calculations is called **round-off error**. It occurs because the arithmetic performed in a machine involves numbers with only a finite number of digits, with the result that calculations are performed with only approximate representations of the actual numbers. In a computer, only a relatively small subset of the real number system is used for the representation of all the real numbers. This subset contains only rational numbers, both positive and negative, and stores the fractional part, together with an exponential part.

Binary Machine Numbers

In 1985, the IEEE (Institute for Electrical and Electronic Engineers) published a report called *Binary Floating Point Arithmetic Standard 754–1985*. An updated version was published in 2008 as *IEEE 754-2008*. This provides standards for binary and decimal floating point numbers, formats for data interchange, algorithms for rounding arithmetic operations, and the handling of exceptions. Formats are specified for single, double, and extended precisions, and these standards are generally followed by all microcomputer manufacturers using floating-point hardware.

A 64-bit (binary digit) representation is used for a real number. The first bit is a sign indicator, denoted s. This is followed by an 11-bit exponent, c, called the **characteristic**, and a 52-bit binary fraction, f, called the **mantissa**. The base for the exponent is 2.

Since 52 binary digits correspond to between 16 and 17 decimal digits, we can assume that a number represented in this system has at least 16 decimal digits of precision. The exponent of 11 binary digits gives a range of 0 to $2^{11}-1=2047$. However, using only positive integers for the exponent would not permit an adequate representation of numbers with small magnitude. To ensure that numbers with small magnitude are equally representable, 1023 is subtracted from the characteristic, so the range of the exponent is actually from -1023 to 1024.

To save storage and provide a unique representation for each floating-point number, a normalization is imposed. Using this system gives a floating-point number of the form

$$(-1)^s 2^{c-1023}(1+f).$$

Illustration Consider the machine number

$$0\ 10000000011\ 1011100100010000000000000000000000000000000000000000.$$

The leftmost bit is $s=0$, which indicates that the number is positive. The next 11 bits, 10000000011, give the characteristic and are equivalent to the decimal number

$$c=1\cdot 2^{10}+0\cdot 2^9+\cdots+0\cdot 2^2+1\cdot 2^1+1\cdot 2^0=1024+2+1=1027.$$

The exponential part of the number is, therefore, $2^{1027-1023} = 2^4$. The final 52 bits specify that the mantissa is

$$f = 1 \cdot \left(\frac{1}{2}\right)^1 + 1 \cdot \left(\frac{1}{2}\right)^3 + 1 \cdot \left(\frac{1}{2}\right)^4 + 1 \cdot \left(\frac{1}{2}\right)^5 + 1 \cdot \left(\frac{1}{2}\right)^8 + 1 \cdot \left(\frac{1}{2}\right)^{12}.$$

As a consequence, this machine number precisely represents the decimal number

$$\begin{aligned}(-1)^s 2^{c-1023}(1+f) &= (-1)^0 \cdot 2^{1027-1023}\left(1 + \left(\frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{4096}\right)\right) \\ &= 27.56640625.\end{aligned}$$

However, the next smallest machine number is

0 10000000011 1011100100001111111111111111111111111111111111111111,

and the next largest machine number is

0 10000000011 1011100100010000000000000000000000000000000000000001.

This means that our original machine number represents not only 27.56640625 but also half of the real numbers that are between 27.56640625 and the next smallest machine number as well as half the numbers between 27.56640625 and the next largest machine number. To be precise, it represents any real number in the interval

$$\begin{aligned}&[27.566406249999982236431605997495353221893310546875,\\ &\ 27.566406250000001776356839400250464677810668945312 5).\end{aligned}$$ ■

The smallest normalized positive number that can be represented has $s = 0$, $c = 1$, and $f = 0$ and is equivalent to

$$2^{-1022} \cdot (1 + 0) \approx 0.22251 \times 10^{-307},$$

and the largest has $s = 0$, $c = 2046$, and $f = 1 - 2^{-52}$ and is equivalent to

$$2^{1023} \cdot (2 - 2^{-52}) \approx 0.17977 \times 10^{309}.$$

Numbers occurring in calculations that have a magnitude less than

$$2^{-1022} \cdot (1 + 0)$$

result in **underflow** and are generally set to zero. Numbers greater than

$$2^{1023} \cdot (2 - 2^{-52})$$

result in **overflow** and typically cause the computations to stop (unless the program has been designed to detect this occurrence). Note that there are two representations for the number zero: a positive 0 when $s = 0$, $c = 0$, and $f = 0$ and a negative 0 when $s = 1$, $c = 0$, and $f = 0$.

Decimal Machine Numbers

The use of binary digits tends to conceal the computational difficulties that occur when a finite collection of machine numbers is used to represent all the real numbers. To examine these problems, we will use more familiar decimal numbers instead of binary representation. Specifically, we assume that machine numbers are represented in the normalized *decimal* floating-point form

$$\pm 0.d_1d_2 \dots d_k \times 10^n, \quad 1 \le d_1 \le 9, \quad \text{and} \quad 0 \le d_i \le 9,$$

for each $i = 2, \dots, k$. Numbers of this form are called k-digit *decimal machine numbers*.

Any positive real number within the numerical range of the machine can be normalized to the form

$$y = 0.d_1d_2 \dots d_kd_{k+1}d_{k+2} \dots \times 10^n.$$

The error that results from replacing a number with its floating-point form is called **round-off error** regardless of whether the rounding or the chopping method is used.

The floating-point form of y, denoted $fl(y)$, is obtained by terminating the mantissa of y at k decimal digits. There are two common ways of performing this termination. One method, called **chopping**, is to simply chop off the digits $d_{k+1}d_{k+2} \dots$. This produces the floating-point form

$$fl(y) = 0.d_1d_2 \dots d_k \times 10^n.$$

The other method, called **rounding**, adds $5 \times 10^{n-(k+1)}$ to y and then chops the result to obtain a number of the form

$$fl(y) = 0.\delta_1\delta_2 \dots \delta_k \times 10^n.$$

For rounding, when $d_{k+1} \ge 5$, we add 1 to d_k to obtain $fl(y)$; that is, we *round up*. When $d_{k+1} < 5$, we simply chop off all but the first k digits; that is, *round down*. If we round down, then $\delta_i = d_i$, for each $i = 1, 2, \dots, k$. However, if we round up, the digits (and even the exponent) might change.

Example 1 Determine the five-digit (a) chopping and (b) rounding values of the irrational number π.

Solution The number π has an infinite decimal expansion of the form $\pi = 3.14159265 \dots$. Written in normalized decimal form, we have

$$\pi = 0.314159265 \dots \times 10^1.$$

The relative error is generally a better measure of accuracy than the absolute error because it takes into consideration the size of the number being approximated.

(a) The floating-point form of π using five-digit chopping is

$$fl(\pi) = 0.31415 \times 10^1 = 3.1415.$$

(b) The sixth digit of the decimal expansion of π is a 9, so the floating-point form of π using five-digit rounding is

$$fl(\pi) = (0.31415 + 0.00001) \times 10^1 = 3.1416.$$ ■

The following definition describes three methods for measuring approximation errors.

Definition 1.15 Suppose that p^* is an approximation to p. The **actual error** is $p - p^*$, the **absolute error** is $|p - p^*|$, and the **relative error** is $\dfrac{|p - p^*|}{|p|}$, provided that $p \ne 0$. ■

Consider the actual, absolute, and relative errors in representing p by p^* in the following example.

Example 2 Determine the actual, absolute, and relative errors when approximating p by p^* when

(a) $p = 0.3000 \times 10^1$ and $p^* = 0.3100 \times 10^1$;

(b) $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$;

(c) $p = 0.3000 \times 10^4$ and $p^* = 0.3100 \times 10^4$.

Solution

(a) For $p = 0.3000 \times 10^1$ and $p^* = 0.3100 \times 10^1$, the actual error is -0.1, the absolute error is 0.1, and the relative error is $0.333\overline{3} \times 10^{-1}$.

(b) For $p = 0.3000 \times 10^{-3}$ and $p^* = 0.3100 \times 10^{-3}$, the actual error is -0.1×10^{-4}, the absolute error is 0.1×10^{-4}, and the relative error is $0.333\overline{3} \times 10^{-1}$.

(c) For $p = 0.3000 \times 10^4$ and $p^* = 0.3100 \times 10^4$, the actual error is -0.1×10^3, the absolute error is 0.1×10^3, and the relative error is again $0.333\overline{3} \times 10^{-1}$.

We often cannot find an accurate value for the true error in an approximation. Instead, we find a bound for the error, which gives us a "worst-case" error.

This example shows that the same relative error, $0.333\overline{3} \times 10^{-1}$, occurs for widely varying absolute errors. As a measure of accuracy, the absolute error can be misleading and the relative error more meaningful because the relative error takes into consideration the size of the value. ■

An error bound is a nonnegative number larger than the absolute error. It is sometimes obtained by the methods of calculus for finding the maximum absolute value of a function. We hope to find the smallest possible upper bound for the error to obtain an estimate of the actual error that is as accurate as possible.

The following definition uses relative error to give a measure of significant digits of accuracy for an approximation.

Definition 1.16 The number p^* is said to approximate p to t **significant digits** (or figures) if t is the largest nonnegative integer for which

$$\frac{|p - p^*|}{|p|} \leq 5 \times 10^{-t}. \quad ■$$

The term *significant digits* is often used to loosely describe the number of decimal digits that appear to be accurate. The definition is more precise, and provides a continuous concept.

Table 1.1 illustrates the continuous nature of significant digits by listing, for the various values of p, the least upper bound of $|p - p^*|$, denoted $\max |p - p^*|$, when p^* agrees with p to four significant digits.

Table 1.1

p	0.1	0.5	100	1000	5000	9990	10000
$\max \lvert p - p^* \rvert$	0.00005	0.00025	0.05	0.5	2.5	4.995	5.

Returning to the machine representation of numbers, we see that the floating-point representation $fl(y)$ for the number y has the relative error

$$\left| \frac{y - fl(y)}{y} \right|.$$

If k decimal digits and chopping are used for the machine representation of

$$y = 0.d_1 d_2 \ldots d_k d_{k+1} \ldots \times 10^n,$$

then

$$\left|\frac{y - fl(y)}{y}\right| = \left|\frac{0.d_1d_2 \ldots d_kd_{k+1} \ldots \times 10^n - 0.d_1d_2 \ldots d_k \times 10^n}{0.d_1d_2 \ldots \times 10^n}\right|$$
$$= \left|\frac{0.d_{k+1}d_{k+2} \ldots \times 10^{n-k}}{0.d_1d_2 \ldots \times 10^n}\right| = \left|\frac{0.d_{k+1}d_{k+2} \ldots}{0.d_1d_2 \ldots}\right| \times 10^{-k}.$$

Since $d_1 \neq 0$, the minimal value of the denominator is 0.1. The numerator is bounded above by 1. As a consequence,

$$\left|\frac{y - fl(y)}{y}\right| \leq \frac{1}{0.1} \times 10^{-k} = 10^{-k+1}.$$

In a similar manner, a bound for the relative error when using k-digit rounding arithmetic is $0.5 \times 10^{-k+1}$. (See Exercise 28.)

Note that the bounds for the relative error using k-digit arithmetic are independent of the number being represented. This result is due to the manner in which the machine numbers are distributed along the real line. Because of the exponential form of the characteristic, the same number of decimal machine numbers is used to represent each of the intervals $[0.1, 1]$, $[1, 10]$, and $[10, 100]$. In fact, within the limits of the machine, the number of decimal machine numbers in $[10^n, 10^{n+1}]$ is constant for all integers n.

Finite-Digit Arithmetic

In addition to inaccurate representation of numbers, the arithmetic performed in a computer is not exact. The arithmetic involves manipulating binary digits by various shifting, or logical, operations. Since the actual mechanics of these operations are not pertinent to this presentation, we shall devise our own approximation to computer arithmetic. Although our arithmetic will not give the exact picture, it suffices to explain the problems that occur. (For an explanation of the manipulations actually involved, the reader is urged to consult more technically oriented computer science texts, such as [Ma], *Computer System Architecture*.)

Assume that the floating-point representations $fl(x)$ and $fl(y)$ are given for the real numbers x and y and that the symbols $\oplus$, $\ominus$, $\otimes$, and $\oslash$ represent machine addition, subtraction, multiplication, and division operations, respectively. We will assume a finite-digit arithmetic given by

$$x \oplus y = fl(fl(x) + fl(y)), \quad x \otimes y = fl(fl(x) \times fl(y)),$$
$$x \ominus y = fl(fl(x) - fl(y)), \quad x \oslash y = fl(fl(x) \div fl(y)).$$

This arithmetic corresponds to performing exact arithmetic on the floating-point representations of x and y and then converting the exact result to its finite-digit floating-point representation.

Example 3 Suppose that $x = \frac{5}{7}$ and $y = \frac{1}{3}$. Use five-digit chopping for calculating $x + y$, $x - y$, $x \times y$, and $x \div y$.

Solution Note that

$$x = \frac{5}{7} = 0.\overline{714285} \quad \text{and} \quad y = \frac{1}{3} = 0.\overline{3}$$

implies that the five-digit chopping values of x and y are

$$fl(x) = 0.71428 \times 10^0 \quad \text{and} \quad fl(y) = 0.33333 \times 10^0.$$

Thus,

$$x \oplus y = fl(fl(x) + fl(y)) = fl\left(0.71428 \times 10^0 + 0.33333 \times 10^0\right)$$
$$= fl\left(1.04761 \times 10^0\right) = 0.10476 \times 10^1.$$

The true value is $x + y = \frac{5}{7} + \frac{1}{3} = \frac{22}{21}$, so we have

$$\text{Absolute Error} = \left|\frac{22}{21} - 0.10476 \times 10^1\right| = 0.190 \times 10^{-4}$$

and

$$\text{Relative Error} = \left|\frac{0.190 \times 10^{-4}}{22/21}\right| = 0.182 \times 10^{-4}.$$

Table 1.2 lists the values of this and the other calculations. ■

Table 1.2

Operation	Result	Actual value	Absolute error	Relative error
$x \oplus y$	0.10476×10^1	22/21	0.190×10^{-4}	0.182×10^{-4}
$x \ominus y$	0.38095×10^0	8/21	0.238×10^{-5}	0.625×10^{-5}
$x \otimes y$	0.23809×10^0	5/21	0.524×10^{-5}	0.220×10^{-4}
$x \oslash y$	0.21428×10^1	15/7	0.571×10^{-4}	0.267×10^{-4}

The maximum relative error for the operations in Example 3 is 0.267×10^{-4}, so the arithmetic produces satisfactory five-digit results. This is not the case in the following example.

Example 4 Suppose that in addition to $x = \frac{5}{7}$ and $y = \frac{1}{3}$ we have

$$u = 0.714251, \quad v = 98765.9, \quad \text{and} \quad w = 0.111111 \times 10^{-4},$$

so that

$$fl(u) = 0.71425 \times 10^0, \quad fl(v) = 0.98765 \times 10^5, \quad \text{and} \quad fl(w) = 0.11111 \times 10^{-4}.$$

Determine the five-digit chopping values of $x \ominus u$, $(x \ominus u) \oplus w$, $(x \ominus u) \otimes v$, and $u \oplus v$.

Solution These numbers were chosen to illustrate some problems that can arise with finite-digit arithmetic. Because x and u are nearly the same, their difference is small. The absolute error for $x \ominus u$ is

$$|(x - u) - (x \ominus u)| = |(x - u) - (fl(fl(x) - fl(u)))|$$
$$= \left|\left(\frac{5}{7} - 0.714251\right) - \left(fl\left(0.71428 \times 10^0 - 0.71425 \times 10^0\right)\right)\right|$$
$$= \left|0.347143 \times 10^{-4} - fl\left(0.00003 \times 10^0\right)\right| = 0.47143 \times 10^{-5}.$$

This approximation has a small absolute error but a large relative error

$$\left|\frac{0.47143 \times 10^{-5}}{0.347143 \times 10^{-4}}\right| \leq 0.136.$$

The subsequent division by the small number w or multiplication by the large number v magnifies the absolute error without modifying the relative error. The addition of the large and small numbers u and v produces large absolute error but not large relative error. These calculations are shown in Table 1.3. ■

Table 1.3

Operation	Result	Actual value	Absolute error	Relative error
$x \ominus u$	0.30000×10^{-4}	0.34714×10^{-4}	0.471×10^{-5}	0.136
$(x \ominus u) \oplus w$	0.27000×10^{1}	0.31242×10^{1}	0.424	0.136
$(x \ominus u) \otimes v$	0.29629×10^{1}	0.34285×10^{1}	0.465	0.136
$u \oplus v$	0.98765×10^{5}	0.98766×10^{5}	0.161×10^{1}	0.163×10^{-4}

One of the most common error-producing calculations involves the cancelation of significant digits due to the subtraction of nearly equal numbers. Suppose two nearly equal numbers x and y, with $x > y$, have the k-digit representations

$$fl(x) = 0.d_1d_2 \dots d_p\alpha_{p+1}\alpha_{p+2} \dots \alpha_k \times 10^n$$

and

$$fl(y) = 0.d_1d_2 \dots d_p\beta_{p+1}\beta_{p+2} \dots \beta_k \times 10^n.$$

The floating-point form of $x - y$ is

$$fl(fl(x) - fl(y)) = 0.\sigma_{p+1}\sigma_{p+2} \dots \sigma_k \times 10^{n-p},$$

where

$$0.\sigma_{p+1}\sigma_{p+2} \dots \sigma_k = 0.\alpha_{p+1}\alpha_{p+2} \dots \alpha_k - 0.\beta_{p+1}\beta_{p+2} \dots \beta_k.$$

The floating-point number used to represent $x - y$ has at most $k - p$ digits of significance. However, in most calculation devices, $x - y$ will be assigned k digits, with the last p being either zero or randomly assigned. Any further calculations involving $x - y$ retain the problem of having only $k - p$ digits of significance, since a chain of calculations is no more accurate than its weakest portion.

If a finite-digit representation or calculation introduces an error, further enlargement of the error occurs when dividing by a number with small magnitude (or, equivalently, when multiplying by a number with large magnitude). Suppose, for example, that the number z has the finite-digit approximation $z + \delta$, where the error δ is introduced by representation or by previous calculation. Now divide by $\varepsilon = 10^{-n}$, where $n > 0$. Then

$$\frac{z}{\varepsilon} \approx fl\left(\frac{fl(z)}{fl(\varepsilon)}\right) = (z + \delta) \times 10^n.$$

The absolute error in this approximation, $|\delta| \times 10^n$, is the original absolute error, $|\delta|$, multiplied by the factor 10^n.

Example 5 Let $p = 0.54617$ and $q = 0.54601$. Use four-digit arithmetic to approximate $p - q$ and determine the absolute and relative errors using **(a)** rounding and **(b)** chopping.

Solution The exact value of $r = p - q$ is $r = 0.00016$.

(a) Suppose the subtraction is performed using four-digit rounding arithmetic. Rounding p and q to four digits gives $p^* = 0.5462$ and $q^* = 0.5460$, respectively, and $r^* = p^* - q^* = 0.0002$ is the four-digit approximation to r. Since

$$\frac{|r - r^*|}{|r|} = \frac{|0.00016 - 0.0002|}{|0.00016|} = 0.25,$$

the result has only one significant digit, whereas p^* and q^* were accurate to four and five significant digits, respectively.

(b) If chopping is used to obtain the four digits, the four-digit approximations to p, q, and r are $p^* = 0.5461$, $q^* = 0.5460$, and $r^* = p^* - q^* = 0.0001$. This gives

$$\frac{|r - r^*|}{|r|} = \frac{|0.00016 - 0.0001|}{|0.00016|} = 0.375,$$

which also results in only one significant digit of accuracy. ■

The loss of accuracy due to round-off error can often be avoided by a reformulation of the calculations, as illustrated in the next example.

Illustration The quadratic formula states that the roots of $ax^2 + bx + c = 0$, when $a \neq 0$, are

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}. \tag{1.1}$$

Consider this formula applied to the equation $x^2 + 62.10x + 1 = 0$, whose roots are approximately

$$x_1 = -0.01610723 \quad \text{and} \quad x_2 = -62.08390.$$

The roots x_1 and x_2 of a general quadratic equation are related to the coefficients by the fact that

$$x_1 + x_2 = -\frac{b}{a} \quad \text{and} \quad x_1 x_2 = \frac{c}{a}.$$

This is a special case of Viète's Formulas for the coefficients of polynomials.

We will again use four-digit rounding arithmetic in the calculations to determine the root. In this equation, b^2 is much larger than $4ac$, so the numerator in the calculation for x_1 involves the *subtraction* of nearly equal numbers. Because

$$\begin{aligned}\sqrt{b^2 - 4ac} &= \sqrt{(62.10)^2 - (4.000)(1.000)(1.000)} \\ &= \sqrt{3856. - 4.000} = \sqrt{3852.} = 62.06,\end{aligned}$$

we have

$$fl(x_1) = \frac{-62.10 + 62.06}{2.000} = \frac{-0.04000}{2.000} = -0.02000,$$

a poor approximation to $x_1 = -0.01611$, with the large relative error

$$\frac{|-0.01611 + 0.02000|}{|-0.01611|} \approx 2.4 \times 10^{-1}.$$

On the other hand, the calculation for x_2 involves the *addition* of the nearly equal numbers $-b$ and $-\sqrt{b^2 - 4ac}$. This presents no problem since

$$fl(x_2) = \frac{-62.10 - 62.06}{2.000} = \frac{-124.2}{2.000} = -62.10$$

has the small relative error

$$\frac{|-62.08 + 62.10|}{|-62.08|} \approx 3.2 \times 10^{-4}.$$

To obtain a more accurate four-digit rounding approximation for x_1, we change the form of the quadratic formula by *rationalizing the numerator*:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}\left(\frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}}\right) = \frac{b^2 - (b^2 - 4ac)}{2a(-b - \sqrt{b^2 - 4ac})},$$

which simplifies to an alternate quadratic formula:

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}. \tag{1.2}$$

Using Eq. (1.2) gives

$$fl(x_1) = \frac{-2.000}{62.10 + 62.06} = \frac{-2.000}{124.2} = -0.01610,$$

which has the small relative error 6.2×10^{-4}.

The rationalization technique can also be applied to give the following alternative quadratic formula for x_2:

$$x_2 = \frac{-2c}{b - \sqrt{b^2 - 4ac}}. \tag{1.3}$$

This is the form to use if b is a negative number. In the illustration, however, the mistaken use of this formula for x_2 would result in not only the subtraction of nearly equal numbers, but also the division by the small result of this subtraction. The inaccuracy that this combination produces,

$$fl(x_2) = \frac{-2c}{b - \sqrt{b^2 - 4ac}} = \frac{-2.000}{62.10 - 62.06} = \frac{-2.000}{0.04000} = -50.00,$$

has the large relative error 1.9×10^{-1}. ■

- The lesson: Think before you compute!

Nested Arithmetic

Accuracy loss due to round-off error can also be reduced by rearranging calculations, as shown in the next example.

Example 6 Evaluate $f(x) = x^3 - 6.1x^2 + 3.2x + 1.5$ at $x = 4.71$ using three-digit arithmetic.

Solution Table 1.4 gives the intermediate results in the calculations.

Table 1.4

	x	x^2	x^3	$6.1x^2$	$3.2x$
Exact	4.71	22.1841	104.487111	135.32301	15.072
Three-digit (chopping)	4.71	22.1	104.	134.	15.0
Three-digit (rounding)	4.71	22.2	105.	135.	15.1

To illustrate the calculations, let us look at those involved with finding x^3 using three-digit rounding arithmetic. First we find

$$x^2 = 4.71^2 = 22.1841 \quad \text{which rounds to } 22.2.$$

Then we use this value of x^2 to find

$$x^3 = x^2 \cdot x = 22.2 \cdot 4.71 = 104.562 \quad \text{which rounds to } 105.$$

Also,

$$6.1x^2 = 6.1(22.2) = 135.42 \quad \text{which rounds to } 135,$$

and

$$3.2x = 3.2(4.71) = 15.072 \quad \text{which rounds to } 15.1.$$

The exact result of the evaluation is

$$\text{Exact:} \quad f(4.71) = 104.487111 - 135.32301 + 15.072 + 1.5 = -14.263899.$$

Using finite-digit arithmetic, the way in which we add the results can effect the final result. Suppose that we add left to right. Then for chopping arithmetic we have

$$\text{Three-digit (chopping):} \quad f(4.71) = ((104. - 134.) + 15.0) + 1.5 = -13.5,$$

and for rounding arithmetic we have

$$\text{Three-digit (rounding):} \quad f(4.71) = ((105. - 135.) + 15.1) + 1.5 = -13.4.$$

(You should carefully verify these results to be sure that your notion of finite-digit arithmetic is correct.) Note that the three-digit chopping values simply retain the leading three digits, with no rounding involved, and differ significantly from the three-digit rounding values.

The relative errors for the three-digit methods are

$$\text{Chopping:} \left| \frac{-14.263899 + 13.5}{-14.263899} \right| \approx 0.05, \quad \text{and} \quad \text{Rounding:} \left| \frac{-14.263899 + 13.4}{-14.263899} \right| \approx 0.06.$$

■

Illustration

Remember that chopping (or rounding) is performed after each calculation.

As an alternative approach, the polynomial $f(x)$ in Example 6 can be written in a **nested** manner as

$$f(x) = x^3 - 6.1x^2 + 3.2x + 1.5 = ((x - 6.1)x + 3.2)x + 1.5.$$

Using three-digit chopping arithmetic now produces

$$\begin{aligned} f(4.71) &= ((4.71 - 6.1)4.71 + 3.2)4.71 + 1.5 = ((-1.39)(4.71) + 3.2)4.71 + 1.5 \\ &= (-6.54 + 3.2)4.71 + 1.5 = (-3.34)4.71 + 1.5 = -15.7 + 1.5 = -14.2. \end{aligned}$$

In a similar manner, we now obtain a three-digit rounding answer of -14.3. The new relative errors are

$$\text{Three-digit (chopping):} \quad \left| \frac{-14.263899 + 14.2}{-14.263899} \right| \approx 0.0045;$$

$$\text{Three-digit (rounding):} \quad \left| \frac{-14.263899 + 14.3}{-14.263899} \right| \approx 0.0025.$$

Nesting has reduced the relative error for the chopping approximation to less than 10% of that obtained initially. For the rounding approximation, the improvement has been even more dramatic; the error in this case has been reduced by more than 95%. ■

Polynomials should *always* be expressed in nested form before performing an evaluation because this form minimizes the number of arithmetic calculations. The decreased error in the illustration is due to the reduction in computations from four multiplications and three additions to two multiplications and three additions. One way to reduce round-off error is to reduce the number of computations.

EXERCISE SET 1.2

1. Compute the absolute error and relative error in approximations of p by p^*.

a. $p = \pi, p^* = 22/7$ **b.** $p = \pi, p^* = 3.1416$

c. $p = e, p^* = 2.718$ **d.** $p = \sqrt{2}, p^* = 1.414$

2. Compute the absolute error and relative error in approximations of p by p^*.

a. $p = e^{10}, p^* = 22000$ **b.** $p = 10^{\pi}, p^* = 1400$

c. $p = 8!, p^* = 39900$ **d.** $p = 9!, p^* = \sqrt{18\pi}(9/e)^9$

3. Suppose p^* must approximate p with relative error at most 10^{-3}. Find the largest interval in which p^* must lie for each value of p.

a. 150 **b.** 900

c. 1500 **d.** 90

4. Find the largest interval in which p^* must lie to approximate p with relative error at most 10^{-4} for each value of p.

a. π **b.** e

c. $\sqrt{2}$ **d.** $\sqrt[3]{7}$

5. Perform the following computations (i) exactly, (ii) using three-digit chopping arithmetic, and (iii) using three-digit rounding arithmetic. (iv) Compute the relative errors in parts (ii) and (iii).

a. $\frac{4}{5} + \frac{1}{3}$ **b.** $\frac{4}{5} \cdot \frac{1}{3}$

c. $\left(\frac{1}{3} - \frac{3}{11}\right) + \frac{3}{20}$ **d.** $\left(\frac{1}{3} + \frac{3}{11}\right) - \frac{3}{20}$

6. Use three-digit rounding arithmetic to perform the following calculations. Compute the absolute error and relative error with the exact value determined to at least five digits.

a. $133 + 0.921$ **b.** $133 - 0.499$

c. $(121 - 0.327) - 119$ **d.** $(121 - 119) - 0.327$

7. Use three-digit rounding arithmetic to perform the following calculations. Compute the absolute error and relative error with the exact value determined to at least five digits.

a. $\dfrac{\frac{13}{14} - \frac{6}{7}}{2e - 5.4}$ **b.** $-10\pi + 6e - \dfrac{3}{62}$

c. $\left(\frac{2}{9}\right) \cdot \left(\frac{9}{7}\right)$ **d.** $\dfrac{\sqrt{13} + \sqrt{11}}{\sqrt{13} - \sqrt{11}}$

8. Repeat Exercise 7 using four-digit rounding arithmetic.

9. Repeat Exercise 7 using three-digit chopping arithmetic.

10. Repeat Exercise 7 using four-digit chopping arithmetic.

11. The first three nonzero terms of the Maclaurin series for the arctangent function are $x - (1/3)x^3 + (1/5)x^5$. Compute the absolute error and relative error in the following approximations of π using the polynomial in place of the arctangent:

a. $4\left[\arctan\left(\frac{1}{2}\right) + \arctan\left(\frac{1}{3}\right)\right]$

b. $16 \arctan\left(\frac{1}{5}\right) - 4 \arctan\left(\frac{1}{239}\right)$

12. The number e can be defined by $e = \sum_{n=0}^{\infty}(1/n!)$, where $n! = n(n-1)\cdots 2\cdot 1$ for $n \neq 0$ and $0! = 1$. Compute the absolute error and relative error in the following approximations of e:

a. $\displaystyle\sum_{n=0}^{5}\frac{1}{n!}$ **b.** $\displaystyle\sum_{n=0}^{10}\frac{1}{n!}$

13. Let

$$f(x) = \frac{x\cos x - \sin x}{x - \sin x}.$$

a. Find $\lim_{x\to 0} f(x)$.

b. Use four-digit rounding arithmetic to evaluate $f(0.1)$.

c. Replace each trigonometric function with its third Maclaurin polynomial and repeat part (b).

d. The actual value is $f(0.1) = -1.99899998$. Find the relative error for the values obtained in parts (b) and (c).

14. Let

$$f(x) = \frac{e^x - e^{-x}}{x}.$$

a. Find $\lim_{x\to 0}(e^x - e^{-x})/x$.

b. Use three-digit rounding arithmetic to evaluate $f(0.1)$.

c. Replace each exponential function with its third Maclaurin polynomial and repeat part (b).

d. The actual value is $f(0.1) = 2.003335000$. Find the relative error for the values obtained in parts (b) and (c).

15. Use four-digit rounding arithmetic and the formulas (1.1), (1.2), and (1.3) to find the most accurate approximations to the roots of the following quadratic equations. Compute the absolute errors and relative errors.

a. $\frac{1}{3}x^2 - \frac{123}{4}x + \frac{1}{6} = 0$

b. $\frac{1}{3}x^2 + \frac{123}{4}x - \frac{1}{6} = 0$

c. $1.002x^2 - 11.01x + 0.01265 = 0$

d. $1.002x^2 + 11.01x + 0.01265 = 0$

16. Use four-digit rounding arithmetic and the formulas (1.1), (1.2), and (1.3) to find the most accurate approximations to the roots of the following quadratic equations. Compute the absolute errors and relative errors.

a. $x^2 - \sqrt{7}x + \sqrt{2} = 0$

b. $\pi x^2 + 13x + 1 = 0$

c. $x^2 + x - e = 0$

d. $x^2 - \sqrt{35}x - 2 = 0$

17. Repeat Exercise 15 using four-digit chopping arithmetic.

18. Repeat Exercise 16 using four-digit chopping arithmetic.

19. Use the 64-bit-long real format to find the decimal equivalent of the following floating-point machine numbers.

a. 0 10000001010 1001001100000000000000000000000000000000000000000000000000

b. 1 10000001010 1001001100000000000000000000000000000000000000000000000000

c. 0 01111111111 0101001100000000000000000000000000000000000000000000000000

d. 0 01111111111 0101001100000000000000000000000000000000000000000000000001

20. Find the next largest and smallest machine numbers in decimal form for the numbers given in Exercise 19.

21. Suppose two points (x_0, y_0) and (x_1, y_1) are on a straight line with $y_1 \neq y_0$. Two formulas are available to find the x-intercept of the line:

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

a. Show that both formulas are algebraically correct.

b. Use the data $(x_0, y_0) = (1.31, 3.24)$ and $(x_1, y_1) = (1.93, 4.76)$ and three-digit rounding arithmetic to compute the x-intercept both ways. Which method is better, and why?

22. The Taylor polynomial of degree n for $f(x) = e^x$ is $\sum_{i=0}^{n}(x^i/i!)$. Use the Taylor polynomial of degree nine and three-digit chopping arithmetic to find an approximation to e^{-5} by each of the following methods.

a. $e^{-5} \approx \sum_{i=0}^{9} \frac{(-5)^i}{i!} = \sum_{i=0}^{9} \frac{(-1)^i 5^i}{i!}$

b. $e^{-5} = \frac{1}{e^5} \approx \frac{1}{\sum_{i=0}^{9} \frac{5^i}{i!}}$.

c. An approximate value of e^{-5} correct to three digits is 6.74×10^{-3}. Which formula, (a) or (b), gives the most accuracy, and why?

23. The two-by-two linear system

$$\begin{aligned} ax + by &= e, \\ cx + dy &= f, \end{aligned}$$

where a, b, c, d, e, f are given, can be solved for x and y as follows:

$$\begin{aligned} &\text{set } m = \frac{c}{a}, \quad \text{provided } a \neq 0; \\ &d_1 = d - mb; \\ &f_1 = f - me; \\ &y = \frac{f_1}{d_1}; \\ &x = \frac{(e - by)}{a}. \end{aligned}$$

Solve the following linear systems using four-digit rounding arithmetic.

a. $1.130x - 6.990y = 14.20$
$1.013x - 6.099y = 14.22$

b. $8.110x + 12.20y = -0.1370$
$-18.11x + 112.2y = -0.1376$

24. Repeat Exercise 23 using four-digit chopping arithmetic.

25. **a.** Show that the polynomial nesting technique described in Example 6 can also be applied to the evaluation of

$$f(x) = 1.01e^{4x} - 4.62e^{3x} - 3.11e^{2x} + 12.2e^{x} - 1.99.$$

b. Use three-digit rounding arithmetic, the assumption that $e^{1.53} = 4.62$, and the fact that $e^{nx} = (e^x)^n$ to evaluate $f(1.53)$ as given in part (a).

c. Redo the calculation in part (b) by first nesting the calculations.

d. Compare the approximations in parts (b) and (c) to the true three-digit result $f(1.53) = -7.61$.

APPLIED EXERCISES

26. The opening example to this chapter described a physical experiment involving the temperature of a gas under pressure. In this application, we were given $P = 1.00$ atm, $V = 0.100$ m^3, $N = 0.00420$ mol, and $R = 0.08206$. Solving for T in the ideal gas law gives

$$T = \frac{PV}{NR} = \frac{(1.00)(0.100)}{(0.00420)(0.08206)} = 290.15 \text{ K} = 17^\circ\text{C}.$$

In the laboratory, it was found that T was 15°C under these conditions, and when the pressure was doubled and the volume halved, T was 19°C. Assume that the data are rounded values accurate to the places given, and show that both laboratory figures are within the bounds of accuracy for the ideal gas law.

THEORETICAL EXERCISES

27. The binomial coefficient

$$\binom{m}{k} = \frac{m!}{k!\,(m-k)!}$$

describes the number of ways of choosing a subset of k objects from a set of m elements.

a. Suppose decimal machine numbers are of the form

$$\pm 0.d_1d_2d_3d_4 \times 10^n, \quad \text{with } 1 \le d_1 \le 9,\ 0 \le d_i \le 9,$$

$$\text{if } i = 2, 3, 4 \quad \text{and} \quad |n| \le 15.$$

What is the largest value of m for which the binomial coefficient $\binom{m}{k}$ can be computed for all k by the definition without causing overflow?

b. Show that $\binom{m}{k}$ can also be computed by

$$\binom{m}{k} = \left(\frac{m}{k}\right)\left(\frac{m-1}{k-1}\right)\cdots\left(\frac{m-k+1}{1}\right).$$

c. What is the largest value of m for which the binomial coefficient $\binom{m}{3}$ can be computed by the formula in part (b) without causing overflow?

d. Use the equation in (b) and four-digit chopping arithmetic to compute the number of possible five-card hands in a 52-card deck. Compute the actual and relative errors.

28. Suppose that $fl(y)$ is a k-digit rounding approximation to y. Show that

$$\left|\frac{y - fl(y)}{y}\right| \le 0.5 \times 10^{-k+1}.$$

[*Hint:* If $d_{k+1} < 5$, then $fl(y) = 0.d_1d_2\ldots d_k \times 10^n$. If $d_{k+1} \ge 5$, then $fl(y) = 0.d_1d_2\ldots d_k \times 10^n + 10^{n-k}$.]

29. Let $f \in C[a, b]$ be a function whose derivative exists on (a, b). Suppose f is to be evaluated at x_0 in (a, b), but instead of computing the actual value $f(x_0)$, the approximate value, $\tilde{f}(x_0)$, is the actual value of f at $x_0 + \epsilon$; that is, $\tilde{f}(x_0) = f(x_0 + \epsilon)$.

a. Use the Mean Value Theorem 1.8 to estimate the absolute error $|f(x_0) - \tilde{f}(x_0)|$ and the relative error $|f(x_0) - \tilde{f}(x_0)|/|f(x_0)|$, assuming $f(x_0) \neq 0$.

b. If $\epsilon = 5 \times 10^{-6}$ and $x_0 = 1$, find bounds for the absolute and relative errors for

i. $f(x) = e^x$

ii. $f(x) = \sin x$

c. Repeat part (b) with $\epsilon = (5 \times 10^{-6})x_0$ and $x_0 = 10$.

DISCUSSION QUESTIONS

1. Discuss the difference between the arithmetic performed by a computer and traditional arithmetic. Why is it so important to recognize the difference?
2. Provide several real-life examples of catastrophic errors that have occurred from the use of finite digital arithmetic and explain what went wrong.
3. Discuss the various different ways to round numbers.
4. Discuss the difference between a number written in standard notation and one that is written in normalized decimal floating-point form. Provide several examples.

1.3 Algorithms and Convergence

Throughout the text, we will be examining approximation procedures, called *algorithms*, involving sequences of calculations. An **algorithm** is a procedure that describes, in an unambiguous manner, a finite sequence of steps to be performed in a specified order. The object of the algorithm is to implement a procedure to solve a problem or approximate a solution to the problem.

The use of an algorithm is as old as formal mathematics, but the name derives from the Arabic mathematician Muhammad ibn-Mŝâ al-Khwarârizmî (c. 780–850). The Latin translation of his works begins with the words "Dixit Algorismi," meaning "al-Khwarârizmî says."

We use a **pseudocode** to describe the algorithms. This pseudocode specifies the form of the input to be supplied and the form of the desired output. Not all numerical procedures give satisfactory output for arbitrarily chosen input. As a consequence, a stopping technique independent of the numerical technique is incorporated into each algorithm to avoid infinite loops.

Two punctuation symbols are used in the algorithms:

- A period (.) indicates the termination of a step.
- A semicolon (;) separates tasks within a step.

Indentation is used to indicate that groups of statements are to be treated as a single entity.

Looping techniques in the algorithms are either counter-controlled, such as

For $i = 1, 2, \ldots, n$
Set $x_i = a + i \cdot h$

or condition-controlled, such as

While $i < N$ do Steps 3–6.

To allow for conditional execution, we use the standard

If . . . then or If . . . then
else

constructions.

The steps in the algorithms follow the rules of structured program construction. They have been arranged so that there should be minimal difficulty translating pseudocode into any programming language suitable for scientific applications.

The algorithms are liberally laced with comments. These are written in italics and contained within parentheses to distinguish them from the algorithmic statements.

NOTE: When the termination of certain nested steps is difficult to determine, we will use a comment such as (End Step 14) to the right of or below the terminating statement. See, for example, the comment on step 5 in Example 1.

Illustration The following algorithm computes $x_1 + x_2 + \cdots + x_N = \sum_{i=1}^{N} x_i$, given N and the numbers $x_1, x_2, \ldots, x_N$.

INPUT $N, x_1, x_2, \ldots, x_n$.

OUTPUT $SUM = \sum_{i=1}^{N} x_i$.

Step 1 Set $SUM = 0$. (*Initialize accumulator.*)

Step 2 For $i = 1, 2, \ldots, N$ do
set $SUM = SUM + x_i$. (*Add the next term.*)

Step 3 OUTPUT (SUM);
STOP. ■

Example 1 The Nth Taylor polynomial for $f(x) = \ln x$ expanded about $x_0 = 1$ is

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

and the value of $\ln 1.5$ to eight decimal places is 0.40546511. Construct an algorithm to determine the minimal value of N required for

$$|\ln 1.5 - P_N(1.5)| < 10^{-5}$$

without using the Taylor polynomial remainder term.

Solution From calculus, we know that if $\sum_{n=1}^{\infty} a_n$ is an alternating series with limit A whose terms decrease in magnitude, then A and the Nth partial sum $A_N = \sum_{n=1}^{N} a_n$ differ by less than the magnitude of the $(N+1)$st term; that is,

$$|A - A_N| \leq |a_{N+1}|.$$

The following algorithm uses this bound.

INPUT value x, tolerance TOL, maximum number of iterations M.
OUTPUT degree N of the polynomial or a message of failure.

Step 1 Set $N = 1$;
$y = x - 1$;
$SUM = 0$;
$POWER = y$;
$TERM = y$;
$SIGN = -1$. (*Used to implement alternation of signs.*)

Step 2 While $N \leq M$ do Steps 3–5.

Step 3 Set $SIGN = -SIGN$; (*Alternate the signs.*)
$SUM = SUM + SIGN \cdot TERM$; (*Accumulate the terms.*)

$POWER = POWER \cdot y$;
$TERM = POWER/(N + 1)$. (*Calculate the next term.*)

Step 4 If $|TERM| < TOL$ then (*Test for accuracy.*)
OUTPUT (N);
STOP. (*The procedure was successful.*)

Step 5 Set $N = N + 1$. (*Prepare for the next iteration. (End Step 2)*)

Step 6 OUTPUT ('Method Failed'); (*The procedure was unsuccessful.*)
STOP.

The input for our problem is $x = 1.5$, $TOL = 10^{-5}$, and perhaps $M = 15$. This choice of M provides an upper bound for the number of calculations we are willing to perform, recognizing that the algorithm is likely to fail if this bound is exceeded. Whether the output is a value for N or the failure message depends on the precision of the computational device. ■

Characterizing Algorithms

We will be considering a variety of approximation problems throughout the text, and in each case we need to determine approximation methods that produce dependably accurate results for a wide class of problems. Because of the differing ways in which the approximation methods are derived, we need a variety of conditions to categorize their accuracy. Not all of these conditions will be appropriate for any particular problem.

The word *stable* has the same root as the words *stand* and *standard*. In mathematics, the term *stable* applied to a problem indicates that a small change in initial data or conditions does not result in a dramatic change in the solution to the problem.

One criterion we will impose on an algorithm whenever possible is that small changes in the initial data produce correspondingly small changes in the final results. An algorithm that satisfies this property is called **stable**; otherwise, it is **unstable**. Some algorithms are stable only for certain choices of initial data and are called **conditionally stable**. We will characterize the stability properties of algorithms whenever possible.

To further consider the subject of round-off error growth and its connection to algorithm stability, suppose an error with magnitude $E_0 > 0$ is introduced at some stage in the calculations and that the magnitude of the error after n subsequent operations is denoted by E_n. The two cases that arise most often in practice are defined as follows.

Definition 1.17 Suppose that $E_0 > 0$ denotes an error introduced at some stage in the calculations and E_n represents the magnitude of the error after n subsequent operations.

- If $E_n \approx CnE_0$, where C is a constant independent of n, then the growth of error is said to be **linear**.
- If $E_n \approx C^nE_0$, for some $C > 1$, then the growth of error is called **exponential**. ■

Linear growth of error is usually unavoidable, and when C and E_0 are small, the results are generally acceptable. Exponential growth of error should be avoided because the term C^n becomes large for even relatively small values of n. This leads to unacceptable inaccuracies, regardless of the size of E_0. As a consequence, an algorithm that exhibits linear growth of error is stable, whereas an algorithm exhibiting exponential error growth is unstable. (See Figure 1.10.)

Figure 1.10

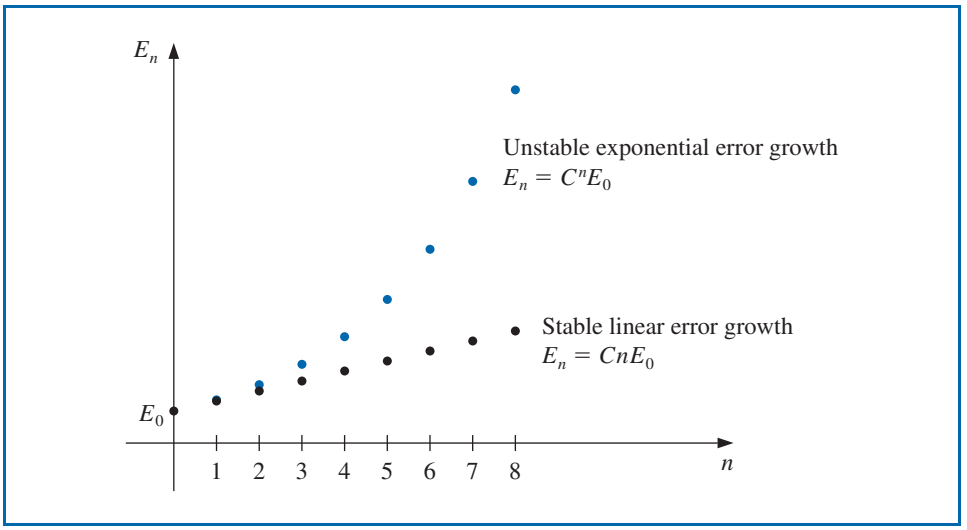


Illustration For any constants c_1 and c_2,

$$p_n = c_1\left(\frac{1}{3}\right)^n + c_2 3^n, \tag{1.4}$$

is a solution to the recursive equation

$$p_n = \frac{10}{3}p_{n-1} - p_{n-2}, \quad \text{for } n = 2, 3, \ldots .$$

This can be seen by noting that

$$\begin{aligned}\frac{10}{3}p_{n-1} - p_{n-2} &= \frac{10}{3}\left[c_1\left(\frac{1}{3}\right)^{n-1} + c_2 3^{n-1}\right] - \left[c_1\left(\frac{1}{3}\right)^{n-2} + c_2 3^{n-2}\right] \\ &= c_1\left(\frac{1}{3}\right)^{n-2}\left[\frac{10}{3}\cdot\frac{1}{3} - 1\right] + c_2 3^{n-2}\left[\frac{10}{3}\cdot 3 - 1\right] \\ &= c_1\left(\frac{1}{3}\right)^{n-2}\left(\frac{1}{9}\right) + c_2 3^{n-2}(9) = c_1\left(\frac{1}{3}\right)^n + c_2 3^n = p_n.\end{aligned}$$

Suppose that we are given $p_0 = 1$ and $p_1 = \frac{1}{3}$. Using these values and Eq. (1.4) we can determine unique values for the constants as $c_1 = 1$ and $c_2 = 0$. So, $p_n = \left(\frac{1}{3}\right)^n$ for all n.

If five-digit rounding arithmetic is used to compute the terms of the sequence given by this equation, then $\hat{p}_0 = 1.0000$ and $\hat{p}_1 = 0.33333$, which requires modifying the constants to $\hat{c}_1 = 1.0000$ and $\hat{c}_2 = -0.12500 \times 10^{-5}$. The sequence $\{\hat{p}_n\}_{n=0}^{\infty}$ generated is then given by

$$\hat{p}_n = 1.0000\left(\frac{1}{3}\right)^n - 0.12500 \times 10^{-5}(3)^n,$$

which has round-off error,

$$p_n - \hat{p}_n = 0.12500 \times 10^{-5}(3^n).$$

This procedure is unstable because the error grows *exponentially* with n, which is reflected in the extreme inaccuracies after the first few terms, as shown in Table 1.5.

Table 1.5

n	Computed $\hat{p}_n$	Correct p_n	Relative error
0	0.10000×10^{1}	0.10000×10^{1}	
1	0.33333×10^{0}	0.33333×10^{0}	
2	0.11110×10^{0}	0.11111×10^{0}	9×10^{-5}
3	0.37000×10^{-1}	0.37037×10^{-1}	1×10^{-3}
4	0.12230×10^{-1}	0.12346×10^{-1}	9×10^{-3}
5	0.37660×10^{-2}	0.41152×10^{-2}	8×10^{-2}
6	0.32300×10^{-3}	0.13717×10^{-2}	8×10^{-1}
7	-0.26893×10^{-2}	0.45725×10^{-3}	7×10^{0}
8	-0.92872×10^{-2}	0.15242×10^{-3}	6×10^{1}

Now consider this recursive equation:

$$p_n = 2p_{n-1} - p_{n-2}, \quad \text{for } n = 2, 3, \ldots .$$

It has the solution $p_n = c_1 + c_2 n$ for any constants c_1 and c_2 because

$$\begin{aligned} 2p_{n-1} - p_{n-2} &= 2(c_1 + c_2(n-1)) - (c_1 + c_2(n-2)) \\ &= c_1(2-1) + c_2(2n - 2 - n + 2) = c_1 + c_2 n = p_n. \end{aligned}$$

If we are given $p_0 = 1$ and $p_1 = \frac{1}{3}$, then constants in this equation are uniquely determined to be $c_1 = 1$ and $c_2 = -\frac{2}{3}$. This implies that $p_n = 1 - \frac{2}{3}n$.

If five-digit rounding arithmetic is used to compute the terms of the sequence given by this equation, then $\hat{p}_0 = 1.0000$ and $\hat{p}_1 = 0.33333$. As a consequence, the five-digit rounding constants are $\hat{c}_1 = 1.0000$ and $\hat{c}_2 = -0.66667$. Thus,

$$\hat{p}_n = 1.0000 - 0.66667n,$$

which has round-off error

$$p_n - \hat{p}_n = \left(0.66667 - \frac{2}{3}\right) n.$$

This procedure is stable because the error grows *linearly* with n, which is reflected in the approximations shown in Table 1.6. ■

Table 1.6

n	Computed $\hat{p}_n$	Correct p_n	Relative error
0	0.10000×10^{1}	0.10000×10^{1}	
1	0.33333×10^{0}	0.33333×10^{0}	
2	-0.33330×10^{0}	-0.33333×10^{0}	9×10^{-5}
3	-0.10000×10^{1}	-0.10000×10^{1}	0
4	-0.16667×10^{1}	-0.16667×10^{1}	0
5	-0.23334×10^{1}	-0.23333×10^{1}	4×10^{-5}
6	-0.30000×10^{1}	-0.30000×10^{1}	0
7	-0.36667×10^{1}	-0.36667×10^{1}	0
8	-0.43334×10^{1}	-0.43333×10^{1}	2×10^{-5}

The effects of round-off error can be reduced by using high-order-digit arithmetic such as the double- or multiple-precision option available on most computers. Disadvantages in using double-precision arithmetic are that it takes more computation time and the growth of round-off error is not entirely eliminated.

One approach to estimating round-off error is to use interval arithmetic (that is, to retain the largest and smallest possible values at each step) so that, in the end, we obtain an interval that contains the true value. Unfortunately, a very small interval may be needed for reasonable implementation.

Rates of Convergence

Since iterative techniques involving sequences are often used, this section concludes with a brief discussion of some terminology used to describe the rate at which convergence occurs. In general, we would like the technique to converge as rapidly as possible. The following definition is used to compare the convergence rates of sequences.

Definition 1.18 Suppose $\{\beta_n\}_{n=1}^{\infty}$ is a sequence known to converge to zero and $\{\alpha_n\}_{n=1}^{\infty}$ converges to a number α. If a positive constant K exists with

$$|\alpha_n - \alpha| \le K|\beta_n|, \quad \text{for large } n,$$

then we say that $\{\alpha_n\}_{n=1}^{\infty}$ converges to α with **rate, or order, of convergence** $O(\beta_n)$. (This expression is read "big oh of β_n".) It is indicated by writing $\alpha_n = \alpha + O(\beta_n)$. ■

Although Definition 1.18 permits $\{\alpha_n\}_{n=1}^{\infty}$ to be compared with an arbitrary sequence $\{\beta_n\}_{n=1}^{\infty}$, in nearly every situation we use

$$\beta_n = \frac{1}{n^p},$$

for some number $p > 0$. We are generally interested in the largest value of p with $\alpha_n = \alpha + O(1/n^p)$.

Example 2 Suppose that, for $n \ge 1$,

$$\alpha_n = \frac{n+1}{n^2} \quad \text{and} \quad \hat{\alpha}_n = \frac{n+3}{n^3}.$$

Both $\lim_{n\to\infty} \alpha_n = 0$ and $\lim_{n\to\infty} \hat{\alpha}_n = 0$, but the sequence $\{\hat{\alpha}_n\}$ converges to this limit much faster than the sequence $\{\alpha_n\}$. Using five-digit rounding arithmetic, we have the values shown in Table 1.7. Determine rates of convergence for these two sequences.

Table 1.7

n	1	2	3	4	5	6	7
α_n	2.00000	0.75000	0.44444	0.31250	0.24000	0.19444	0.16327
$\hat{\alpha}_n$	4.00000	0.62500	0.22222	0.10938	0.064000	0.041667	0.029155

There are numerous other ways of describing the growth of sequences and functions, some of which require bounds both above and below the sequence or function under consideration. Any good book that analyzes algorithms, for example, [CLRS], will include this information.

Solution Define the sequences $\beta_n = 1/n$ and $\hat{\beta}_n = 1/n^2$. Then

$$|\alpha_n - 0| = \frac{n+1}{n^2} \le \frac{n+n}{n^2} = 2 \cdot \frac{1}{n} = 2\beta_n$$

and

$$|\hat{\alpha}_n - 0| = \frac{n+3}{n^3} \le \frac{n+3n}{n^3} = 4 \cdot \frac{1}{n^2} = 4\hat{\beta}_n.$$

Hence, the rate of convergence of $\{\alpha_n\}$ to zero is similar to the convergence of $\{1/n\}$ to zero, whereas $\{\hat{\alpha}_n\}$ converges to zero at a rate similar to the more rapidly convergent sequence $\{1/n^2\}$. We express this by writing

$$\alpha_n = 0 + O\left(\frac{1}{n}\right) \quad \text{and} \quad \hat{\alpha}_n = 0 + O\left(\frac{1}{n^2}\right). \qquad \blacksquare$$

We also use the O (*big oh*) notation to describe the rate at which functions converge.

Definition 1.19 Suppose that $\lim_{h\to 0} G(h) = 0$ and $\lim_{h\to 0} F(h) = L$. If a positive constant K exists with

$$|F(h) - L| \le K|G(h)|, \quad \text{for sufficiently small } h,$$

then we write $F(h) = L + O(G(h))$. $\blacksquare$

The functions we use for comparison generally have the form $G(h) = h^p$, where $p > 0$. We are interested in the largest value of p for which $F(h) = L + O(h^p)$.

Example 3 Use the third Taylor polynomial about $h = 0$ to show that $\cos h + \frac{1}{2}h^2 = 1 + O(h^4)$.

Solution In Example 3(b) of Section 1.1, we found that this polynomial is

$$\cos h = 1 - \frac{1}{2}h^2 + \frac{1}{24}h^4 \cos\tilde{\xi}(h),$$

for some number $\tilde{\xi}(h)$ between zero and h. This implies that

$$\cos h + \frac{1}{2}h^2 = 1 + \frac{1}{24}h^4 \cos\tilde{\xi}(h).$$

Hence,

$$\left|\left(\cos h + \frac{1}{2}h^2\right) - 1\right| = \left|\frac{1}{24}\cos\tilde{\xi}(h)\right| h^4 \le \frac{1}{24}h^4,$$

so as $h \to 0$, $\cos h + \frac{1}{2}h^2$ converges to its limit, 1, about as fast as h^4 converges to 0. That is,

$$\cos h + \frac{1}{2}h^2 = 1 + O(h^4). \qquad \blacksquare$$

EXERCISE SET 1.3

1. Use three-digit chopping arithmetic to compute the following sums. For each part, which method is more accurate, and why?

a. $\sum_{i=1}^{10}(1/i^2)$ first by $\frac{1}{1} + \frac{1}{4} + \cdots + \frac{1}{100}$ and then by $\frac{1}{100} + \frac{1}{81} + \cdots + \frac{1}{1}$.

b. $\sum_{i=1}^{10}(1/i^3)$ first by $\frac{1}{1} + \frac{1}{8} + \frac{1}{27} + \cdots + \frac{1}{1000}$ and then by $\frac{1}{1000} + \frac{1}{729} + \cdots + \frac{1}{1}$.

2. The number e is defined by $e = \sum_{n=0}^{\infty}(1/n!)$, where $n! = n(n-1)\cdots 2 \cdot 1$ for $n \ne 0$ and $0! = 1$. Use four-digit chopping arithmetic to compute the following approximations to e and determine the absolute and relative errors.

a. $e \approx \sum_{n=0}^{5} \frac{1}{n!}$ **b.** $e \approx \sum_{j=0}^{5} \frac{1}{(5-j)!}$

c. $e \approx \sum_{n=0}^{10} \frac{1}{n!}$ **d.** $e \approx \sum_{j=0}^{10} \frac{1}{(10-j)!}$

3. The Maclaurin series for the arctangent function converges for $-1 < x \leq 1$ and is given by

$$\arctan x = \lim_{n\to\infty} P_n(x) = \lim_{n\to\infty} \sum_{i=1}^{n} (-1)^{i+1} \frac{x^{2i-1}}{2i-1}.$$

a. Use the fact that $\tan \pi/4 = 1$ to determine the number of n terms of the series that need to be summed to ensure that $|4P_n(1) - \pi| < 10^{-3}$.

b. The C++ programming language requires the value of π to be within 10^{-10}. How many terms of the series would we need to sum to obtain this degree of accuracy?

4. Exercise 3 details a rather inefficient means of obtaining an approximation to π. The method can be improved substantially by observing that $\pi/4 = \arctan \frac{1}{2} + \arctan \frac{1}{3}$ and evaluating the series for the arctangent at $\frac{1}{2}$ and at $\frac{1}{3}$. Determine the number of terms that must be summed to ensure an approximation to π to within 10^{-3}.

5. Another formula for computing π can be deduced from the identity $\pi/4 = 4 \arctan \frac{1}{5} - \arctan \frac{1}{239}$. Determine the number of terms that must be summed to ensure an approximation to π to within 10^{-3}.

6. Find the rates of convergence of the following sequences as $n \to \infty$.

a. $\lim_{n\to\infty} \sin \frac{1}{n} = 0$ **b.** $\lim_{n\to\infty} \sin \frac{1}{n^2} = 0$

c. $\lim_{n\to\infty} \left(\sin \frac{1}{n}\right)^2 = 0$ **d.** $\lim_{n\to\infty} [\ln(n+1) - \ln(n)] = 0$

7. Find the rates of convergence of the following functions as $h \to 0$.

a. $\lim_{h\to 0} \frac{\sin h}{h} = 1$ **b.** $\lim_{h\to 0} \frac{1-\cos h}{h} = 0$

c. $\lim_{h\to 0} \frac{\sin h - h\cos h}{h} = 0$ **d.** $\lim_{h\to 0} \frac{1-e^h}{h} = -1$

THEORETICAL EXERCISES

8. Suppose that $0 < q < p$ and that $\alpha_n = \alpha + O\left(n^{-p}\right)$.

a. Show that $\alpha_n = \alpha + O\left(n^{-q}\right)$.

b. Make a table listing $1/n$, $1/n^2$, $1/n^3$, and $1/n^4$ for $n = 5$, 10, 100, and 1000 and discuss the varying rates of convergence of these sequences as n becomes large.

9. Suppose that $0 < q < p$ and that $F(h) = L + O(h^p)$.

a. Show that $F(h) = L + O(h^q)$.

b. Make a table listing h, h^2, h^3, and h^4 for $h = 0.5$, 0.1, 0.01, and 0.001 and discuss the varying rates of convergence of these powers of h as h approaches zero.

10. Suppose that as x approaches zero,

$$F_1(x) = L_1 + O(x^\alpha) \quad \text{and} \quad F_2(x) = L_2 + O(x^\beta).$$

Let c_1 and c_2 be nonzero constants and define

$$F(x) = c_1F_1(x) + c_2F_2(x) \quad \text{and} \quad G(x) = F_1(c_1x) + F_2(c_2x).$$

Show that if γ = minimum $\{\alpha, \beta\}$, then, as x approaches zero,

a. $F(x) = c_1L_1 + c_2L_2 + O(x^\gamma)$

b. $G(x) = L_1 + L_2 + O(x^\gamma)$.

11. The sequence $\{F_n\}$ described by $F_0 = 1, F_1 = 1$, and $F_{n+2} = F_n + F_{n+1}$, if $n \geq 0$, is called a *Fibonacci sequence*. Its terms occur naturally in many botanical species, particularly those with petals or scales arranged in the form of a logarithmic spiral. Consider the sequence $\{x_n\}$, where $x_n = F_{n+1}/F_n$. Assuming that $\lim_{n\to\infty} x_n = x$ exists, show that $x = (1 + \sqrt{5})/2$. This number is called the *golden ratio.*

12. Show that the Fibonacci sequence also satisfies the equation

$$F_n \equiv \tilde{F}_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right].$$

13. Describe the output of the following algorithm. How does this algorithm compare to the illustration on page 32?

INPUT $n, x_1, x_2, \ldots, x_n$.
OUTPUT SUM.
Step 1 Set SUM = x_1.
Step 2 For $i = 2, 3, \ldots, n$ do Step 3.
Step 3 SUM = SUM + x_i.
Step 4 OUTPUT SUM;
STOP.

14. Compare the following three algorithms. When is the algorithm of part 1a correct.?

a. INPUT $n, x_1, x_2, \ldots, x_n$.
OUTPUT PRODUCT.
Step 1 Set PRODUCT = 0.
Step 2 For $i = 1, 2, \ldots, n$ do
Set PRODUCT = PRODUCT * x_i.
Step 3 OUTPUT PRODUCT;
STOP.

b. INPUT $n, x_1, x_2, \ldots, x_n$.
OUTPUT PRODUCT.
Step 1 Set PRODUCT = 1.
Step 2 For $i = 1, 2, \ldots, n$ do
Set PRODUCT = PRODUCT * x_i.
Step 3 OUTPUT PRODUCT;
STOP.

c. INPUT $n, x_1, x_2, \ldots, x_n$.
OUTPUT PRODUCT.
Step 1 Set PRODUCT = 1.
Step 2 For $i = 1, 2, \ldots, n$ do
if $x_i = 0$ then set PRODUCT = 0;
OUTPUT PRODUCT;
STOP
else set PRODUCT = PRODUCT * x_i.
Step 3 OUTPUT PRODUCT;
STOP.

15. **a.** How many multiplications and additions are required to determine a sum of the form

$$\sum_{i=1}^{n}\sum_{j=1}^{i} a_i b_j?$$

b. Modify the sum in part (a) to an equivalent form that reduces the number of computations.