Numerical Methods
Using MATLAB
Fourth Edition

John. H. Mathews
California State University, Fullerton

Kurtis D. Fink
Northwest Missouri State University

PEARSON
Prentice
Hall
Upper Saddle River, New Jersey 07458

Library of Congress Cataloging-in-Publication Data
Numerical methods using MATLAB/John H. Mathews,
Kurtis D. Fink—4th ed.
p. cm.
Includes index
ISBN 0-13-065248-2
1. Numerical analysis–Data processing. 2. MATLAB. I. Fink, Kurtis D.
II. Title.

QA297.M39 2004
518–dc22 2003061022

Editor-in-Chief: *Sally Yagan*
Acquisitions Editor: *George Lobell*
Production Editor: *Lynn Savino Wendel*
Vice President/Director of Production and Manufacturing: *David W. Riccardi*
Senior Managing Editor: *Linda Mihatov Behrens*
Assistant Managing Editor: *Bayani Mendoza de Leon*
Executive Managing Editor: *Kathleen Schiaparelli*
Assisant Manufacturing Manager/Buyer: *Michael Bell*
Manufacturing Manager: *Trudy Pisciotti*
Marketing Manager: *Halee Dinsey*
Marketing Assistant: *Rachael Beckman*
Art Director: *Jayne Conte*
Editorial Assistant: *Jennifer Brady*
Cover Designer: *Bruce Kenselaar*
Cover Photo Credits: *Fans Quilt, Quiltmaker unidentified, initialed "PM" Indiana 1925–1935*
Cotton, wool, and rayon, with cotton embroidery 82 × 71 *1/2"*
Collection American Folk Art Museum, New York Gift of David Pottinger 1980.37.86

Pearson Prentice Hall
Pearson Education, Inc.
Upper Saddle River, New Jersey 07458

Formerly published under the title Numerical Methods for Mathematics,
Science, and Engineering, © 1992, 1987 by John H. Mathews.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN 0-13-065248-2

Pearson Education LTD., *London*
Pearson Education Australia PTY, Limited, *Sydney*
Pearson Education Singapore, Pte. Ltd
Pearson Education North Asia Ltd, *Hong Kong*
Pearson Education Canada, Ltd., *Toronto*
Pearson Educacion de Mexico, S.A. de C.V.
Pearson Education - Japan, *Tokyo*
Pearson Education Malaysia, Pte. Ltd

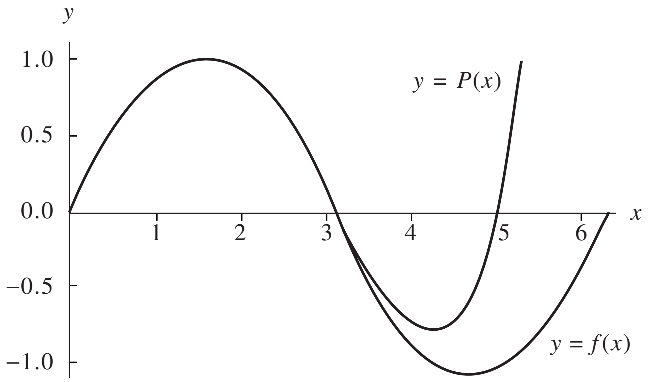


Figure 1.6 The graph of $f(x) = \sin(x)$ and the Taylor polynomial $P_9(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9!$.

the following derivatives at $x = 0$ and substituting the numerical values into formula (17).

$$\begin{aligned}
f(x) &= \sin(x), & f(0) &= 0,\\
f'(x) &= \cos(x), & f'(0) &= 1,\\
f''(x) &= -\sin(x), & f''(0) &= 0,\\
f^{(3)}(x) &= -\cos(x), & f^{(3)}(0) &= -1,\\
&\vdots & &\vdots\\
f^{(9)}(x) &= \cos(x), & f^{(9)}(0) &= 1,
\end{aligned}$$

$$P_9(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}.$$

A graph of both f and P_9 over the interval $[0, 2\pi]$ is shown in Figure 1.6. ■

Corollary 1.1. If $P_n(x)$ is the Taylor polynomial of degree n given in Theorem 1.12, then

(19) $$P_n^{(k)}(x_0) = f^{(k)}(x_0) \quad \text{for } k = 0,\ 1,\ \ldots,\ n.$$

Evaluation of a Polynomial

Let the polynomial $P(x)$ of degree n have the form

(20) $$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_2x^2 + a_1x + a_0.$$

Horner's method or ***synthetic division*** is a technique for evaluating polynomials. It can be thought of as nested multiplication. For example, a fifth-degree polynomial can be written in the nested multiplication form

$$P_5(x) = ((((a_5x + a_4)x + a_3)x + a_2)x + a_1)x + a_0.$$

Theorem 1.13 (Horner's Method for Polynomial Evaluation). Assume that $P(x)$ is the polynomial given in equation (20) and $x = c$ is a number for which $P(c)$ is to be evaluated.

Set $b_n = a_n$ and compute

$$b_k = a_k + cb_{k+1} \qquad \text{for } k = n-1,\ n-2,\ \ldots,\ 1,\ 0; \tag{21}$$

then $b_0 = P(c)$. Moreover, if

$$Q_0(x) = b_nx^{n-1} + b_{n-1}x^{n-2} + \cdots + b_3x^2 + b_2x + b_1, \tag{22}$$

then

$$P(x) = (x-c)Q_0(x) + R_0, \tag{23}$$

where $Q_0(x)$ is the quotient polynomial of degree $n-1$ and $R_0 = b_0 = P(c)$ is the remainder.

Proof. Substituting the right side of equation (22) for $Q_0(x)$ and b_0 for R_0 in equation (23) yields

$$\begin{aligned} P(x) &= (x-c)(b_nx^{n-1} + b_{n-1}x^{n-2} + \cdots + b_3x^2 + b_2x + b_1) + b_0 \\ &= b_nx^n + (b_{n-1} - cb_n)x^{n-1} + \cdots + (b_2 - cb_3)x^2 \\ &\quad + (b_1 - cb_2)x + (b_0 - cb_1). \end{aligned} \tag{24}$$

The numbers b_k are determined by comparing the coefficients of x^k in equations (20) and (24), as shown in Table 1.1.

The value $P(c) = b_0$ is easily obtained by substituting $x = c$ into equation (22) and using the fact that $R_0 = b_0$:

$$P(c) = (c-c)Q_0(c) + R_0 = b_0. \tag{25}$$

•

The recursive formula for b_k given in (21) is easy to implement with a computer. A simple algorithm is

```
$b(n) = a(n);$
for $k = n - 1: -1: 0$
    $b(k) = a(k) + c * b(k+1);$
end
```

Table 1.1 Coefficients b_k for Horner's Method

x^k	Comparing (20) and (24)	Solving for b_k
x^n	$a_n = b_n$	$b_n = a_n$
x^{n-1}	$a_{n-1} = b_{n-1} - cb_n$	$b_{n-1} = a_{n-1} + cb_n$
$\vdots$	$\vdots$	$\vdots$
x^k	$a_k = b_k - cb_{k+1}$	$b_k = a_k + cb_{k+1}$
$\vdots$	$\vdots$	$\vdots$
x^0	$a_0 = b_0 - cb_1$	$b_0 = a_0 + cb_1$

Table 1.2 Horner's Table for the Synthetic Division Process

Input	a_n	a_{n-1}	a_{n-2}	$\cdots$	a_k	$\cdots$	a_2	a_1	a_0
c		xb_n	xb_{n-1}	$\cdots$	xb_{k+1}	$\cdots$	xb_3	xb_2	xb_1
	b_n	b_{n-1}	b_{n-2}	$\cdots$	b_k	$\cdots$	b_2	b_1	$b_0 = P(c)$
									Output

When Horner's method is performed by hand, it is easier to write the coefficients of $P(x)$ on a line and perform the calculation $b_k = a_k + cb_{k+1}$ below a_k in a column. The format for this procedure is illustrated in Table 1.2.

Example 1.9. Use synthetic division (Horner's method) to find $P(3)$ for the polynomial

$$P(x) = x^5 - 6x^4 + 8x^3 + 8x^2 + 4x - 40.$$

	a_5	a_4	a_3	a_2	a_1	a_0
Input	1	−6	8	8	4	−40
$c = 3$		3	−9	−3	15	57
	1	−3	−1	5	19	$17 = P(3) = b_0$
	b_5	b_4	b_3	b_2	b_1	Output

Therefore, $P(3) = 17$. ■

(a) $\left\{\dfrac{1}{2^n}\right\}_{n=0}^{\infty}$ **(b)** $\left\{\dfrac{2}{3^n}\right\}_{n=1}^{\infty}$

(c) $\displaystyle\sum_{n=1}^{\infty} \frac{3}{n(n+1)}$ **(d)** $\displaystyle\sum_{k=1}^{\infty} \frac{1}{4k^2-1}$

12. Find the Taylor polynomial of degree $n = 4$ for each function expanded about the given value of x_0.

(a) $f(x) = \sqrt{x}$, $x_0 = 1$

(b) $f(x) = x^5 + 4x^2 + 3x + 1$, $x_0 = 0$

(c) $f(x) = \cos(x)$, $x_0 = 0$

13. Given that $f(x) = \sin(x)$ and $P(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9!$, show that $P^{(k)}(0) = f^{(k)}(0)$ for $k = 1, 2, \ldots, 9$.

14. Use synthetic division (Horner's method) to find $P(c)$.

(a) $P(x) = x^4 + x^3 - 13x^2 - x - 12$, $c = 3$

(b) $P(x) = 2x^7 + x^6 + x^5 - 2x^4 - x + 23$, $c = -1$

15. Find the average area of all circles centered at the origin with radii between 1 and 3.

16. Assume that a polynomial, $P(x)$, has n real roots in the interval $[a, b]$. Show that $P^{(n-1)}(x)$ has at least one real root in the interval $[a, b]$.

17. Assume that f, f', and f'' are defined on the interval $[a, b]$; $f(a) = f(b) = 0$; and $f(c) > 0$ for $c \in (a, b)$. Show that there is a number $d \in (a, b)$ such that $f''(d) < 0$.

1.2 Binary Numbers

Human beings do arithmetic using the decimal (base 10) number system. Most computers do arithmetic using the binary (base 2) number system. It may seem otherwise, since communication with the computer (input/output) is in base 10 numbers. This transparency does not mean that the computer uses base 10. In fact, it converts inputs to base 2 (or perhaps base 16), then performs base 2 arithmetic, and finally, translates the answer into base 10 before it displays a result. Some experimentation is required to verify this. One computer with nine decimal digits of accuracy gave the answer

$$\sum_{k=1}^{100{,}000} 0.1 = 9999.99447. \tag{1}$$

Here the intent was to add the number $\frac{1}{10}$ repeatedly 100,000 times. The mathematical answer is exactly 10,000. One goal is to understand the reason for the computer's apparently flawed calculation. At the end of this section it will be shown how something is lost when the computer translates the decimal fraction $\frac{1}{10}$ into a binary number.

13. Use Table 1.3 to determine what happens when a computer with a 4-bit mantissa performs the following calculations.

(a) $\left(\frac{1}{3} + \frac{1}{5}\right) + \frac{1}{6}$ **(b)** $\left(\frac{1}{10} + \frac{1}{3}\right) + \frac{1}{5}$

(c) $\left(\frac{3}{17} + \frac{1}{9}\right) + \frac{1}{7}$ **(d)** $\left(\frac{7}{10} + \frac{1}{9}\right) + \frac{1}{7}$

14. Show that when 2 is replaced by 3 in all the formulas in (8), the result is a method for finding the base 3 expansion of a positive integer. Express the following integers in base 3.

(a) 10 **(b)** 23 **(c)** 421 **(d)** 1784

15. Show that when 2 is replaced by 3 in (22), the result is a method for finding the base 3 expansion of a positive number R that lies in the interval $0 < R < 1$. Express the following numbers in base 3.

(a) $1/3$ **(b)** $1/2$ **(c)** $1/10$ **(d)** $11/27$

16. Show that when 2 is replaced by 5 in all the formulas in (8), the result is a method for finding the base 5 expansion of a positive integer. Express the following integers in base 5.

(a) 10 **(b)** 35 **(c)** 721 **(d)** 734

17. Show that when 2 is replaced by 5 in (22), the result is a method for finding the base 5 expansion of a positive number R that lies in the interval $0 < R < 1$. Express the following numbers in base 5.

(a) $1/3$ **(b)** $1/2$ **(c)** $1/10$ **(d)** $154/625$

1.3 Error Analysis

In the practice of numerical analysis it is important to be aware that computed solutions are not exact mathematical solutions. The precision of a numerical solution can be diminished in several subtle ways. Understanding these difficulties can often guide the practitioner in the proper implementation and/or development of numerical algorithms.

Definition 1.7. Suppose that $\widehat{p}$ is an approximation to p. The ***absolute error*** is $E_p = |p - \widehat{p}|$, and the ***relative error*** is $R_p = |p - \widehat{p}|/|p|$, provided that $p \neq 0$. ▲

The absolute error is simply the difference between the true value and the approximate value, whereas the relative error expresses the error as a percentage of the true value.

Example 1.14. Find the error and relative error in the following three cases. Let $x = 3.141592$ and $\widehat{x} = 3.14$; then the error is

$$E_x = |x - \widehat{x}| = |3.141592 - 3.14| = 0.001592, \tag{1a}$$

and the relative error is

$$R_x = \frac{|x-\widehat{x}|}{|x|} = \frac{0.001592}{3.141592} = 0.00507.$$

Let $y = 1{,}000{,}000$ and $\widehat{y} = 999{,}996$; then the error is

(1b) $$E_y = |y-\widehat{y}| = |1{,}000{,}000 - 999, 996| = 4,$$

and the relative error is

$$R_y = \frac{|y-\widehat{y}|}{|y|} = \frac{4}{1{,}000{,}000} = 0.000004.$$

Let $z = 0.000012$ and $\widehat{z} = 0.000009$; then the error is

(1c) $$E_z = |z-\widehat{z}| = |0.000012 - 0.000009| = 0.000003,$$

and the relative error is

$$R_z = \frac{|z-\widehat{z}|}{|z|} = \frac{0.000003}{0.000012} = 0.25. \qquad \blacksquare$$

In case (1a), there is not too much difference between E_x and R_x, and either could be used to determine the accuracy of $\widehat{x}$. In case (1b), the value of y is of magnitude 10^6, the error E_y is large, and the relative error R_y is small. In this case, $\widehat{y}$ would probably be considered a good approximation to y. In case (1c), z is of magnitude 10^{-6} and the error E_z is the smallest of all three cases, but the relative error R_z is the largest. In terms of percentage, it amounts to 25%, and thus $\widehat{z}$ is a bad approximation to z. Observe that as $|p|$ moves away from 1 (greater than or less than) the relative error R_p is a better indicator than E_p of the accuracy of the approximation. Relative error is preferred for floating-point representations since it deals directly with the mantissa.

Definition 1.8. The number $\widehat{p}$ is said to ***approximate*** p to d significant digits if d is the largest nonnegative integer for which

(2) $$\frac{|p-\widehat{p}|}{|p|} < \frac{10^{1-d}}{2}. \qquad \blacktriangle$$

Example 1.15. Determine the number of significant digits for the approximations in Example 1.14.

(3a) If $x = 3.141592$ and $\widehat{x} = 3.14$, then $|x-\widehat{x}|/|x| = 0.000507 < 10^{-2}/2$. Therefore, $\widehat{x}$ approximates x to two significant digits. 3

(3b) If $y = 1{,}000{,}000$ and $\widehat{y} = 999{,}996$, then $|y-\widehat{y}|/|y| = 0.000004 < 10^{-5}/2$. Therefore, $\widehat{y}$ approximates y to six significant digits.

(3c) If $z = 0.000012$ and $\widehat{z} = 0.000009$, then $|z-\widehat{z}|/|z| = 0.25 < 10^{-0}/2$. Therefore, $\widehat{z}$ approximates z to one significant digit. $\blacksquare$

The function $P(x)$ is the Taylor polynomial of degree $n = 2$ for $f(x)$ expanded about $x = 0$.

For the first function

$$f(0.01) = \frac{e^{0.01} - 1 - 0.01}{(0.01)^2} = \frac{1.010050 - 1 - 0.01}{0.001} = 0.5.$$

For the second function

$$\begin{aligned} P(0.01) &= \frac{1}{2} + \frac{0.01}{6} + \frac{0.001}{24} \\ &= 0.5 + 0.001667 + 0.000004 = 0.501671. \end{aligned}$$

The answer $P(0.01) = 0.501671$ contains less error and is the same as that obtained by rounding the true answer 0.50167084168057542... to six digits. ■

For polynomial evaluation, the rearrangement of terms into nested multiplication form will sometimes produce a better result.

Example 1.19. Let $P(x) = x^3 - 3x^2 + 3x - 1$ and $Q(x) = ((x - 3)x + 3)x - 1$. Use three-digit rounding arithmetic to compute approximations to $P(2.19)$ and $Q(2.19)$. Compare them with the true values, $P(2.19) = Q(2.19) = 1.685159$.

$$\begin{aligned} P(2.19) &\approx (2.19)^3 - 3(2.19)^2 + 3(2.19) - 1 \\ &= 10.5 - 14.4 + 6.57 - 1 = 1.67. \\ Q(2.19) &\approx ((2.19 - 3)2.19 + 3)2.19 - 1 = 1.69. \end{aligned}$$

The errors are 0.015159 and −0.004841, respectively. Thus the approximation $Q(2.19) \approx 1.69$ has less error. Exercise 6 explores the situation near the root of this polynomial. ■

$O(h^n)$ Order of Approximation

Clearly, the sequences $\left\{\frac{1}{n^2}\right\}_{n=1}^{\infty}$ and $\left\{\frac{1}{n}\right\}_{n=1}^{\infty}$ are both converging to zero. In addition, it should be observed that the first sequence is converging to zero more rapidly than the second sequence. In the coming chapters some special terminology and notation will be used to describe how rapidly a sequence is converging.

Definition 1.9. The function $f(h)$ is said to be ***big Oh*** of $g(h)$, denoted $f(h) = \boldsymbol{O}(g(h))$, if there exist constants C and c such that

$$(7) \qquad |f(h)| \le C|g(h)| \quad \text{whenever } h \le c. \qquad \blacktriangle$$

Example 1.20. Consider the functions $f(x) = x^2 + 1$ and $g(x) = x^3$. Since $x^2 \le x^3$ and $1 \le x^3$ for $x \ge 1$, it follows that $x^2 + 1 \le 2x^3$ for $x \ge 1$. Therefore, $f(x) = \boldsymbol{O}(g(x))$. ■

The big Oh notation provides a useful way of describing the rate of growth of a function in terms of well-known elementary functions (x^n, $x^{1/n}$, a^x, $\log_a x$, etc.).

The rate of convergence of sequences can be described in a similar manner.

Definition 1.10. Let $\{x_n\}_{n=1}^{\infty}$ and $\{y_n\}_{n=1}^{\infty}$ be two sequences. The sequence $\{x_n\}$ is said to be of order big Oh of $\{y_n\}$, denoted $x_n = \boldsymbol{O}(y_n)$, if there exist constants C and N such that

$$(8) \qquad |x_n| \le C|y_n| \quad \text{whenever } n \ge N. \qquad \blacktriangle$$

Example 1.21. $\dfrac{n^2-1}{n^3} = \boldsymbol{O}\left(\dfrac{1}{n}\right)$, since $\dfrac{n^2-1}{n^3} \le \dfrac{n^2}{n^3} = \dfrac{1}{n}$ whenever $n \ge 1$. ■

Often a function $f(h)$ is approximated by a function $p(h)$ and the error bound is known to be $M|h^n|$. This leads to the following definition.

Definition 1.11. Assume that $f(h)$ is approximated by the function $p(h)$ and that there exist a real constant $M > 0$ and a positive integer n so that

$$(9) \qquad \frac{|f(h)-p(h)|}{|h^n|} \le M \quad \text{for sufficiently small } h.$$

We say that $p(h)$ ***approximates*** $f(h)$ with order of approximation $\boldsymbol{O}(h^n)$ and write

$$(10) \qquad f(h) = p(h) + \boldsymbol{O}(h^n). \qquad \blacktriangle$$

When relation (9) is rewritten in the form $|f(h)-p(h)| \le M|h^n|$, we see that the notation $\boldsymbol{O}(h^n)$ stands in place of the error bound $M|h^n|$. The following results show how to apply the definition to simple combinations of two functions.

Theorem 1.15. Assume that $f(h) = p(h) + \boldsymbol{O}(h^n)$, $g(h) = q(h) + \boldsymbol{O}(h^m)$, and $r = \min\{m, n\}$. Then

$$(11) \qquad f(h) + g(h) = p(h) + q(h) + \boldsymbol{O}(h^r),$$

$$(12) \qquad f(h)g(h) = p(h)q(h) + \boldsymbol{O}(h^r),$$

and

$$(13) \qquad \frac{f(h)}{g(h)} = \frac{p(h)}{q(h)} + \boldsymbol{O}(h^r) \quad \text{provided that } g(h) \ne 0 \text{ and } q(h) \ne 0.$$

It is instructive to consider $p(x)$ to be the nth Taylor polynomial approximation of $f(x)$; then the remainder term is simply designated $\boldsymbol{O}(h^{n+1})$, which stands for the presence of omitted terms starting with the power h^{n+1}. The remainder term converges to zero with the same rapidity that h^{n+1} converges to zero as h approaches zero, as expressed in the relationship

$$\text{(14)} \qquad \boldsymbol{O}(h^{n+1}) \approx Mh^{n+1} \approx \frac{f^{(n+1)}(c)}{(n+1)!}h^{n+1}$$

for sufficiently small h. Hence the notation $\boldsymbol{O}(h^{n+1})$ stands in place of the quantity Mh^{n+1}, where M is a constant or "behaves like a constant."

Theorem 1.16 (Taylor's Theorem). Assume that $f \in C^{n+1}[a, b]$. If both x_0 and $x = x_0 + h$ lie in $[a, b]$, then

$$\text{(15)} \qquad f(x_0 + h) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}h^k + \boldsymbol{O}(h^{n+1}).$$

The following example illustrates the theorems above. The computations use the addition properties (i) $\boldsymbol{O}(h^p) + \boldsymbol{O}(h^p) = \boldsymbol{O}(h^p)$, (ii) $\boldsymbol{O}(h^p) + \boldsymbol{O}(h^q) = \boldsymbol{O}(h^r)$, where $r = \min\{p, q\}$, and the multiplicative property (iii) $\boldsymbol{O}(h^p)\boldsymbol{O}(h^q) = \boldsymbol{O}(h^s)$, where $s = p + q$.

Example 1.22. Consider the Taylor polynomial expansions

$$e^h = 1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + \boldsymbol{O}(h^4) \quad \text{and} \quad \cos(h) = 1 - \frac{h^2}{2!} + \frac{h^4}{4!} + \boldsymbol{O}(h^6).$$

Determine the order of approximation for their sum and product.

For the sum we have

$$\begin{aligned} e^h + \cos(h) &= 1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + \boldsymbol{O}(h^4) + 1 - \frac{h^2}{2!} + \frac{h^4}{4!} + \boldsymbol{O}(h^6) \\ &= 2 + h + \frac{h^3}{3!} + \boldsymbol{O}(h^4) + \frac{h^4}{4!} + \boldsymbol{O}(h^6). \end{aligned}$$

Since $\boldsymbol{O}(h^4) + \frac{h^4}{4!} = \boldsymbol{O}(h^4)$ and $\boldsymbol{O}(h^4) + \boldsymbol{O}(h^6) = \boldsymbol{O}(h^4)$, this reduces to

$$e^h + \cos(h) = 2 + h + \frac{h^3}{3!} + \boldsymbol{O}(h^4),$$

and the order of approximation is $\boldsymbol{O}(h^4)$.

The product is treated similarly:

$$\begin{aligned}
e^h\cos(h) &= \left(1+h+\frac{h^2}{2!}+\frac{h^3}{3!}+\boldsymbol{O}(h^4)\right)\left(1-\frac{h^2}{2!}+\frac{h^4}{4!}+\boldsymbol{O}(h^6)\right)\\
&= \left(1+h+\frac{h^2}{2!}+\frac{h^3}{3!}\right)\left(1-\frac{h^2}{2!}+\frac{h^4}{4!}\right)\\
&\quad+\left(1+h+\frac{h^2}{2!}+\frac{h^3}{3!}\right)\boldsymbol{O}(h^6)+\left(1-\frac{h^2}{2!}+\frac{h^4}{4!}\right)\boldsymbol{O}(h^4)\\
&\quad+\boldsymbol{O}(h^4)\boldsymbol{O}(h^6)\\
&= 1+h-\frac{h^3}{3}-\frac{5h^4}{24}-\frac{h^5}{24}+\frac{h^6}{48}+\frac{h^7}{144}\\
&\quad+\boldsymbol{O}(h^6)+\boldsymbol{O}(h^4)+\boldsymbol{O}(h^4)\boldsymbol{O}(h^6).
\end{aligned}$$

Since $\boldsymbol{O}(h^4)\boldsymbol{O}(h^6) = \boldsymbol{O}(h^{10})$ and

$$-\frac{5h^4}{24}-\frac{h^5}{24}+\frac{h^6}{48}+\frac{h^7}{144}+\boldsymbol{O}(h^6)+\boldsymbol{O}(h^4)+\boldsymbol{O}(h^{10}) = \boldsymbol{O}(h^4),$$

the preceding equation is simplified to yield

$$e^h\cos(h) = 1+h-\frac{h^3}{3}+\boldsymbol{O}(h^4),$$

and the order of approximation is $\boldsymbol{O}(h^4)$. ■

Order of Convergence of a Sequence

Numerical approximations are often arrived at by computing a sequence of approximations that get closer and closer to the answer desired. The definition of big Oh for sequences was given in Definition 1.10, and the definition of order of convergence for a sequence is analogous to that given for functions in Definition 1.11.

Definition 1.12. Suppose that $\lim_{n\to\infty} x_n = x$ and $\{r_n\}_{n=1}^{\infty}$ is a sequence with $\lim_{n\to\infty} r_n = 0$. We say that $\{x_n\}_{n=1}^{\infty}$ ***converges*** to x with the order of convergence $\boldsymbol{O}(r_n)$, if there exists a constant $K > 0$ such that

$$\frac{|x_n - x|}{|r_n|} \le K \qquad \text{for } n \text{ sufficiently large.}$$

This is indicated by writing $x_n = x + \boldsymbol{O}(r_n)$, or $x_n \to x$ with order of convergence $\boldsymbol{O}(r_n)$. ▲

Example 1.23. Let $x_n = \cos(n)/n^2$ and $r_n = 1/n^2$; then $\lim_{n\to\infty} x_n = 0$ with a rate of convergence $\boldsymbol{O}(1/n^2)$. This follows immediately from the relation

$$\frac{|\cos(n)/n^2|}{|1/n^2|} = |\cos(n)| \le 1 \qquad \text{for all } n.$$

■

Exercises for Error Analysis

1. Find the error E_x and relative error R_x. Also determine the number of significant digits in the approximation.

(a) $x = 2.71828182, \widehat{x} = 2.7182$

(b) $y = 98,350, \widehat{y} = 98,000$

(c) $z = 0.000068, \widehat{z} = 0.00006$

2. Complete the following computation:

$$\int_0^{1/4} e^{x^2}\,dx \approx \int_0^{1/4} \left(1 + x^2 + \frac{x^2}{2!} + \frac{x^6}{3!}\right) dx = \widehat{p}.$$

State what type of error is present in this situation. Compare your answer with the true value $p = 0.2553074606$.

3. **(a)** Consider the data $p_1 = 1.414$ and $p_2 = 0.09125$, which have four significant digits of accuracy. Determine the proper answer for the sum $p_1 + p_2$ and the product $p_1 p_2$.

(b) Consider the data $p_1 = 31.415$ and $p_2 = 0.027182$, which have five significant digits of accuracy. Determine the proper answer for the sum $p_1 + p_2$ and the product $p_1 p_2$.

4. Complete the following computation and state what type of error is present in this situation.

(a) $$\frac{\sin\left(\frac{\pi}{4} + 0.00001\right) - \sin\left(\frac{\pi}{4}\right)}{0.00001} = \frac{0.70711385222 - 0.70710678119}{0.00001} = \cdots$$

(b) $$\frac{\ln(2 + 0.00005) - \ln(2)}{0.00005} = \frac{0.69317218025 - 0.69314718056}{0.00005} = \cdots$$

5. Sometimes the loss of significance error can be avoided by rearranging terms in the function using a known identity from trigonometry or algebra. Find an equivalent formula for the following functions that avoids a loss of significance.

(a) $\ln(x + 1) - \ln(x)$ for large x

(b) $\sqrt{x^2 + 1} - x$ for large x

(c) $\cos^2(x) - \sin^2(x)$ for $x \approx \pi/4$

(d) $\sqrt{\dfrac{1 + \cos(x)}{2}}$ for $x \approx \pi$

6. *Polynomial evaluation.* Let $P(x) = x^3 - 3x^2 + 3x - 1$, $Q(x) = ((x - 3)x + 3)x - 1$, and $R(x) = (x - 1)^3$.

(a) Use four-digit rounding arithmetic and compute $P(2.72)$, $Q(2.72)$, and $R(2.72)$. In the computation of $P(x)$, assume that $(2.72)^3 = 20.12$ and $(2.72)^2 = 7.398$.

(b) Use four-digit rounding arithmetic and compute $P(0.975)$, $Q(0.975)$, and $R(0.975)$. In the computation of $P(x)$, assume that $(0.975)^3 = 0.9268$ and $(0.975)^2 = 0.9506$.

7. Use three-digit rounding arithmetic to compute the following sums (sum in the given order):

(a) $\displaystyle\sum_{k=1}^{6} \frac{1}{3^k}$ **(b)** $\displaystyle\sum_{k=1}^{6} \frac{1}{3^{7-k}}$

8. Discuss the propagation of error for the following:

(a) The sum of three numbers:

$$p + q + r = (\widehat{p} + \epsilon_p) + (\widehat{q} + \epsilon_q) + (\widehat{r} + \epsilon_r).$$

(b) The quotient of two numbers: $\dfrac{p}{q} = \dfrac{\widehat{p} + \epsilon_p}{\widehat{q} + \epsilon_q}$.

(c) The product of three numbers:

$$pqr = (\widehat{p} + \epsilon_p)(\widehat{q} + \epsilon_q)(\widehat{r} + \epsilon_r).$$

9. Given the Taylor polynomial expansions

$$\frac{1}{1-h} = 1 + h + h^2 + h^3 + \boldsymbol{O}(h^4)$$

and

$$\cos(h) = 1 - \frac{h^2}{2!} + \frac{h^4}{4!} + \boldsymbol{O}(h^6),$$

determine the order of approximation for their sum and product.

10. Given the Taylor polynomial expansions

$$e^h = 1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + \frac{h^4}{4!} + \boldsymbol{O}(h^5)$$

and

$$\sin(h) = h - \frac{h^3}{3!} + \boldsymbol{O}(h^5),$$

determine the order of approximation for their sum and product.

11. Given the Taylor polynomial expansions

$$\cos(h) = 1 - \frac{h^2}{2!} + \frac{h^4}{4!} + \boldsymbol{O}(h^6)$$

and

$$\sin(h) = h - \frac{h^3}{3!} + \frac{h^5}{5!} + \boldsymbol{O}(h^7),$$

determine the order of approximation for their sum and product.

Theorem 2.1. Assume that g is a continuous function and that $\{p_n\}_{n=0}^{\infty}$ is a sequence generated by fixed-point iteration. If $\lim_{n\to\infty} p_n = P$, then P is a fixed point of $g(x)$.

Proof. If $\lim_{n\to\infty} p_n = P$, then $\lim_{n\to\infty} p_{n+1} = P$. It follows from this result, the continuity of g, and the relation $p_{n+1} = g\,(p_n)$ that

$$g(P) = g\left(\lim_{n\to\infty} p_n\right) = \lim_{n\to\infty} g(p_n) = \lim_{n\to\infty} p_{n+1} = P. \tag{2}$$

Therefore, P is a fixed point of $g(x)$. •

Example 2.2. Consider the convergent iteration

$$p_0 = 0.5 \qquad \text{and} \qquad p_{k+1} = e^{-p_k} \qquad \text{for } k = 0, 1, \ldots .$$

The first 10 terms are obtained by the calculations

$$\begin{aligned}
p_1 &= e^{-0.500000} = 0.606531\\
p_2 &= e^{-0.606531} = 0.545239\\
p_3 &= e^{-0.545239} = 0.579703\\
&\ \ \vdots \qquad\qquad\ \ \vdots\\
p_9 &= e^{-0.566409} = 0.567560\\
p_{10} &= e^{-0.567560} = 0.566907
\end{aligned}$$

The sequence is converging, and further calculations reveal that

$$\lim_{n\to\infty} p_n = 0.567143\ldots .$$

Thus we have found an approximation for the fixed point of the function $y = e^{-x}$. ■

The following two theorems establish conditions for the existence of a fixed point and the convergence of the fixed-point iteration process to a fixed point.

Theorem 2.2. Assume that $g \in C[a, b]$.

(3) If the range of the mapping $y = g(x)$ satisfies $y \in [a, b]$ for all $x \in [a, b]$, then g has a fixed point in $[a, b]$.

(4) Furthermore, suppose that $g'(x)$ is defined over (a, b) and that a positive constant $K < 1$ exists with $|g'(x)| \leq K < 1$ for all $x \in (a, b)$; then g has a unique fixed point P in $[a, b]$.

Proof of (3). If $g(a) = a$ or $g(b) = b$, the assertion is true. Otherwise, the values of $g(a)$ and $g(b)$ must satisfy $g(a) \in (a, b]$ and $g(b) \in [a, b)$. The function $f(x) \equiv x - g(x)$ has the property that

$$f(a) = a - g(a) < 0 \quad \text{and} \quad f(b) = b - g(b) > 0.$$

Now apply Theorem 1.2, the intermediate value theorem, to $f(x)$, with the constant $L = 0$, and conclude that there exists a number P with $P \in (a, b)$ so that $f(P) = 0$. Therefore, $P = g(P)$ and P is the desired fixed point of $g(x)$.

Proof of (4). Now we must show that this solution is unique. By way of contradiction, let us make the additional assumption that there exist two fixed points P_1 and P_2. Now apply Theorem 1.6, the mean value theorem, and conclude that there exists a number $d \in (a, b)$ so that

$$g'(d) = \frac{g(P_2) - g(P_1)}{P_2 - P_1}. \tag{5}$$

Next, use the facts that $g(P_1) = P_1$ and $g(P_2) = P_2$ to simplify the right side of equation (5) and obtain

$$g'(d) = \frac{P_2 - P_1}{P_2 - P_1} = 1.$$

But this contradicts the hypothesis in (4) that $|g'(x)| < 1$ over (a, b), so it is not possible for two fixed points to exist. Therefore, $g(x)$ has a unique fixed point P in $[a, b]$ under the conditions given in (4). •

Example 2.3. Apply Theorem 2.2 to show rigorously that $g(x) = \cos(x)$ has a unique fixed point in [0, 1].

Clearly, $g \in C[0, 1]$. Also, $g(x) = \cos(x)$ is a decreasing function on [0, 1]; thus its range on [0, 1] is $[\cos(1), 1] \subseteq [0, 1]$. Thus condition (3) of Theorem 2.2 is satisfied and g has a fixed point in [0, 1]. Finally, if $x \in (0, 1)$, then $|g'(x)| = |-\sin(x)| = \sin(x) \leq \sin(1) < 0.8415 < 1$. Thus $K = \sin(1) < 1$, condition (4) of Theorem 2.2 is satisfied, and g has a unique fixed point in [0, 1]. ■

We can now state a theorem that can be used to determine whether the fixed-point iteration process given in (1) will produce a convergent or a divergent sequence.

Theorem 2.3 (Fixed-Point Theorem). Assume that (i) $g, g' \in C[a, b]$, (ii) K is a positive constant, (iii) $p_0 \in (a, b)$, and (iv) $g(x) \in [a, b]$ for all $x \in [a, b]$.

(6) If $|g'(x)| \leq K < 1$ for all $x \in [a, b]$, then the iteration $p_n = g(p_{n-1})$ will converge to the unique fixed point $P \in [a, b]$. In this case, P is said to be an attractive fixed point.

(7) If $|g'(x)| > 1$ for all $x \in [a, b]$, then the iteration $p_n = g(p_{n-1})$ will not converge to P. In this case, P is said to be a repelling fixed point and the iteration exhibits local divergence.

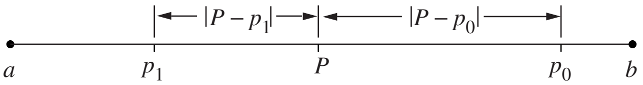


Figure 2.3 The relationship among P, p_0, p_1, $|P - p_0|$, and $|P - p_1|$.

Remark 1. It is assumed that $p_0 \neq P$ in statement (7).

Remark 2. Because g is continuous on an interval containing P, it is permissible to use the simpler criterion $|g'(P)| \leq K < 1$ and $|g'(P)| > 1$ in (6) and (7), respectively.

Proof. We first show that the points $\{p_n\}_{n=0}^{\infty}$ all lie in (a, b). Starting with p_0, we apply Theorem 1.6, the mean value theorem. There exists a value $c_0 \in (a, b)$ so that

$$\begin{aligned} |P - p_1| &= |g(P) - g(p_0)| = |g'(c_0)(P - p_0)| \\ &= |g'(c_0)||P - p_0| \leq K|P - p_0| < |P - p_0|. \end{aligned} \tag{8}$$

Therefore, p_1 is no further from P than p_0 was, and it follows that $p_1 \in (a, b)$ (see Figure 2.3). In general, suppose that $p_{n-1} \in (a, b)$; then

$$\begin{aligned} |P - p_n| &= |g(P) - g(p_{n-1})| = |g'(c_{n-1})(P - p_{n-1})| \\ &= |g'(c_{n-1})||P - p_{n-1}| \leq K|P - p_{n-1}| < |P - p_{n-1}|. \end{aligned} \tag{9}$$

Therefore, $p_n \in (a, b)$ and hence, by induction, all the points $\{p_n\}_{n=0}^{\infty}$ lie in (a, b).

To complete the proof of (6), we will show that

$$\lim_{n\to\infty} |P - p_n| = 0. \tag{10}$$

First, a proof by induction will establish the inequality

$$|P - p_n| \leq K^n |P - p_0|. \tag{11}$$

The case $n = 1$ follows from the details in relation (8). Using the induction hypothesis $|P - p_{n-1}| \leq K^{n-1}|P - p_0|$ and the ideas in (9), we obtain

$$|P - p_n| \leq K|P - p_{n-1}| \leq KK^{n-1}|P - p_0| = K^n|P - p_0|.$$

Thus, by induction, inequality (11) holds for all n. Since $0 < K < 1$, the term K^n goes to zero as n goes to infinity. Hence

$$0 \leq \lim_{n\to\infty} |P - p_n| \leq \lim_{n\to\infty} K^n|P - p_0| = 0. \tag{12}$$

The limit of $|P - p_n|$ is squeezed between zero on the left and zero on the right, so we can conclude that $\lim_{n\to\infty} |P - p_n| = 0$. Thus $\lim_{n\to\infty} p_n = P$ and, by Theorem 2.1, the iteration $p_n = g(p_{n-1})$ converges to the fixed point P. Therefore, statement (6) of Theorem 2.3 is proved. We leave statement (7) for the reader to investigate. •

If $|g'(P)| > 1$, then the iteration $p_{n+1} = g(p_n)$ produces a sequence that diverges away from P. The two simple types of divergent iteration, monotone and oscillating, are illustrated in Figure 2.5(a) and (b), respectively.

Example 2.4. Consider the iteration $p_{n+1} = g(p_n)$ when the function $g(x) = 1+x-x^2/4$ is used. The fixed points can be found by solving the equation $x = g(x)$. The two solutions (fixed points of g) are $x = -2$ and $x = 2$. The derivative of the function is $g'(x) = 1-x/2$, and there are only two cases to consider.

Case (i): $P = -2$

Start with $p_0 = -2.05$

then get $p_1 = -2.100625$

$p_2 = -2.20378135$

$p_3 = -2.41794441$

$$\vdots$$

$$\lim_{n\to\infty} p_n = -\infty.$$

Since $|g'(x)| > \frac{3}{2}$ on $[-3, -1]$, by Theorem 2.3, the sequence will not converge to $P = -2$.

Case (ii): $P = 2$

Start with $p_0 = 1.6$

then get $p_1 = 1.96$

$p_2 = 1.9996$

$p_3 = 1.99999996$

$$\vdots$$

$$\lim_{n\to\infty} p_n = 2.$$

Since $|g'(x)| < \frac{1}{2}$ on $[1, 3]$, by Theorem 2.3, the sequence will converge to $P = 2$.

■

Theorem 2.3 does not state what will happen when $g'(P) = 1$. The next example has been specially constructed so that the sequence $\{p_n\}$ converges whenever $p_0 > P$ and it diverges if we choose $p_0 < P$.

Example 2.5. Consider the iteration $p_{n+1} = g(p_n)$ when the function $g(x) = 2(x-1)^{1/2}$ for $x \geq 1$ is used. Only one fixed point $P = 2$ exists. The derivative is $g'(x) = 1/(x-1)^{1/2}$ and $g'(2) = 1$, so Theorem 2.3 does not apply. There are two cases to consider when the starting value lies to the left or right of $P = 2$.

Case (i): Start with $p_0 = 1.5$,

then get $p_1 = 1.41421356$

$p_2 = 1.28718851$

$p_3 = 1.07179943$

$p_4 = 0.53590832$

$$\vdots$$

$$p_5 = 2(-0.46409168)^{1/2}.$$

Since p_4 lies outside the domain of $g(x)$, the term p_5 cannot be computed.

Case (ii): Start with $p_0 = 2.5$,

then get $p_1 = 2.44948974$

$p_2 = 2.40789513$

$p_3 = 2.37309514$

$p_4 = 2.34358284$

$$\vdots$$

$$\lim_{n\to\infty} p_n = 2.$$

This sequence is converging too slowly to the value $P = 2$; indeed, $P_{1000} = 2.00398714$.

■

Exercises for Iteration for Solving $x = g(x)$

1. Determine rigorously if each function has a unique fixed point on the given interval (follow Example 2.3).

(a) $g(x) = 1 - x^2/4$ on $[0, 1]$

(b) $g(x) = 2^{-x}$ on $[0, 1]$

(c) $g(x) = 1/x$ on $[0.5, 5.2]$

2. Investigate the nature of the fixed-point iteration when

$$g(x) = -4 + 4x - \frac{1}{2}x^2.$$

(a) Solve $g(x) = x$ and show that $P = 2$ and $P = 4$ are fixed points.

(b) Use the starting value $p_0 = 1.9$ and compute p_1, p_2, and p_3.

(c) Use the starting value $p_0 = 3.8$ and compute p_1, p_2, and p_3.

(d) Find the errors E_k and relative errors R_k for the values p_k in parts (b) and (c).

(e) What conclusions can be drawn from Theorem 2.3?

3. Graph $g(x)$, the line $y = x$, and the given fixed point P on the same coordinate system. Using the given starting value p_0, compute p_1 and p_2. Construct figures similar to Figures 2.4 and 2.5. Based on your graph, determine geometrically if fixed-point iteration converges.

(a) $g(x) = (6 + x)^{1/2}$, $P = 3$, and $p_0 = 7$

(b) $g(x) = 1 + 2/x$, $P = 2$, and $p_0 = 4$

(c) $g(x) = x^2/3$, $P = 3$, and $p_0 = 3.5$

(d) $g(x) = -x^2 + 2x + 2$, $P = 2$, and $p_0 = 2.5$

4. Let $g(x) = x^2 + x - 4$. Can fixed-point iteration be used to find the solution(s) to the equation $x = g(x)$? Why?

5. Let $g(x) = x\cos(x)$. Solve $x = g(x)$ and find all the fixed points of g (there are infinitely many). Can fixed-point iteration be used to find the solution(s) to the equation $x = g(x)$? Why?

6. Suppose that $g(x)$ and $g'(x)$ are defined and continuous on (a, b); $p_0, p_1, p_2 \in (a, b)$; and $p_1 = g(p_0)$ and $p_2 = g(p_1)$. Also, assume that there exists a constant K such that $|g'(x)| < K$. Show that $|p_2 - p_1| < K|p_1 - p_0|$. *Hint.* Use the mean value theorem.

7. Suppose that $g(x)$ and $g'(x)$ are continuous on (a, b) and that $|g'(x)| > 1$ on this interval. If the fixed point P and the initial approximations p_0 and p_1 lie in the interval (a, b), then show that $p_1 = g(p_0)$ implies that $|E_1| = |P - p_1| > |P - p_0| = |E_0|$. Hence statement (7) of Theorem 2.3 is established (local divergence).

8. Let $g(x) = -0.0001x^2 + x$ and $p_0 = 1$, and consider fixed-point iteration.

(a) Show that $p_0 > p_1 > \cdots > p_n > p_{n+1} > \cdots$.

(b) Show that $p_n > 0$ for all n.

Table 2.1 Bisection Method Solution of $x\sin(x) - 1 = 0$

k	Left endpoint, a_k	Midpoint, c_k	Right endpoint, b_k	Function value, $f(c_k)$
0	0	1.	2.	−0.158529
1	1.0	1.5	2.0	0.496242
2	1.00	1.25	1.50	0.186231
3	1.000	1.125	1.250	0.015051
4	1.0000	1.0625	1.1250	−0.071827
5	1.06250	1.09375	1.12500	−0.028362
6	1.093750	1.109375	1.125000	−0.006643
7	1.1093750	1.1171875	1.1250000	0.004208
8	1.10937500	1.11328125	1.11718750	−0.001216
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

To continue, we squeeze from the left and set $a_1 = c_0$ and $b_1 = b_0$. The midpoint is $c_1 = 1.5$ and $f(c_1) = 0.496242$. Now, $f(1) = -0.158529$ and $f(1.5) = 0.496242$ imply that the root lies in the interval $[a_1, c_1] = [1.0, 1.5]$. The next decision is to squeeze from the right and set $a_2 = a_1$ and $b_2 = c_1$. In this manner we obtain a sequence $\{c_k\}$ that converges to $r \approx 1.114157141$. A sample calculation is given in Table 2.1. ■

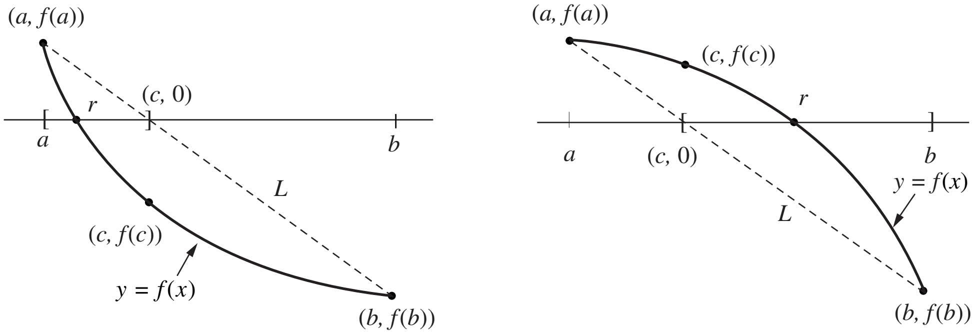
A virtue of the bisection method is that formula (10) provides a predetermined estimate for the accuracy of the computed solution. In Example 2.7 the width of the starting interval was $b_0 - a_0 = 2$. Suppose that Table 2.1 were continued to the thirty-first iterate; then, by (10), the error bound would be $|E_{31}| \le (2-0)/2^{32} \approx 4.656613 \times 10^{-10}$. Hence c_{31} would be an approximation to r with nine decimal places of accuracy. The number N of repeated bisections needed to guarantee that the Nth midpoint c_N is an approximation to a zero and has an error less than the preassigned value δ is

$$N = \operatorname{int}\left(\frac{\ln(b-a) - \ln(\delta)}{\ln(2)}\right). \tag{15}$$

The proof of this formula is left as an exercise.

Another popular algorithm is the ***method of false position*** or the ***regula falsi method***. It was developed because the bisection method converges at a fairly slow speed. As before, we assume that $f(a)$ and $f(b)$ have opposite signs. The bisection method used the midpoint of the interval $[a, b]$ as the next iterate. A better approximation is obtained if we find the point $(c, 0)$ where the secant line L joining the points $(a, f(a))$ and $(b, f(b))$ crosses the x-axis (see Figure 2.8). To find the value c, we write down two versions of the slope m of the line L:

$$m = \frac{f(b) - f(a)}{b - a}, \tag{16}$$



(a) If $f(a)$ and $f(c)$ have opposite signs, then squeeze from the right.

(b) If $f(c)$ and $f(b)$ have opposite signs, then squeeze from the left.

Figure 2.8 The decision process for the false position method.

where the points $(a, f(a))$ and $(b, f(b))$ are used, and

$$m = \frac{0 - f(b)}{c - b}, \tag{17}$$

where the points $(c, 0)$ and $(b, f(b))$ are used.

Equating the slopes in (16) and (17), we have

$$\frac{f(b) - f(a)}{b - a} = \frac{0 - f(b)}{c - b},$$

which is easily solved for c to get

$$c = b - \frac{f(b)(b - a)}{f(b) - f(a)}. \tag{18}$$

The three possibilities are the same as before:

(19) If $f(a)$ and $f(c)$ have opposite signs, a zero lies in $[a, c]$.

(20) If $f(c)$ and $f(b)$ have opposite signs, a zero lies in $[c, b]$.

(21) If $f(c) = 0$, then the zero is c.

Convergence of the False Position Method

The decision process implied by (19) and (20) along with (18) is used to construct a sequence of intervals $\{[a_n, b_n]\}$ each of which brackets the zero. At each step the approximation of the zero r is

$$c_n = b_n - \frac{f(b_n)(b_n - a_n)}{f(b_n) - f(a_n)}, \tag{22}$$

Table 2.2 False Position Method Solution of $x \sin(x) - 1 = 0$

k	Left endpoint, a_k	Midpoint, c_k	Right endpoint, b_k	Function value, $f(c_k)$
0	0.00000000	1.09975017	2.00000000	−0.02001921
1	1.09975017	1.12124074	2.00000000	0.00983461
2	1.09975017	1.11416120	1.12124074	0.00000563
3	1.09975017	1.11415714	1.11416120	0.00000000

Program 2.2 (Bisection Method). To approximate a root of the equation $f(x) = 0$ in the interval $[a, b]$. Proceed with the method only if $f(x)$ is continuous and $f(a)$ and $f(b)$ have opposite signs.

```
function [c,err,yc]=bisect(f,a,b,delta)
%Input  - f is the function input as a string 'f'
%       - a and b are the left and right endpoints
%       - delta is the tolerance
%Output - c is the zero
%       - yc=f(c)
%       - err is the error estimate for c
ya=feval(f,a);
yb=feval(f,b);
if ya*yb>0,break,end
max1=1+round((log(b-a)-log(delta))/log(2));
for k=1:max1
    c=(a+b)/2;
    yc=feval(f,c);
    if yc==0
       a=c;
       b=c;
elseif yb*yc>0
      b=c;
      yb=yc;
   else
      a=c;
      ya=yc;
   end
   if b-a < delta, break,end
end
c=(a+b)/2;
err=abs(b-a);
yc=feval(f,c);
```

Program 2.3 (False Position or Regula Falsi Method). To approximate a root of the equation $f(x) = 0$ in the interval $[a, b]$. Proceed with the method only if $f(x)$ is continuous and $f(a)$ and $f(b)$ have opposite signs.

```
function [c,err,yc]=regula(f,a,b,delta,epsilon,max1)
%Input  - f is the function input as a string 'f'
%       - a and b are the left and right endpoints
%       - delta is the tolerance for the zero
%       - epsilon is the tolerance for the value of f at the zero
%       - max1 is the maximum number of iterations
%Output - c is the zero
%       - yc=f(c)
%       - err is the error estimate for c
ya=feval(f,a);
yb=feval(f,b);
if ya*yb>0
   disp('Note: f(a)*f(b)>0'),
   break,
end
for k=1:max1
   dx=yb*(b-a)/(yb-ya);
   c=b-dx;
   ac=c-a;
   yc=feval(f,c);
   if yc==0,break;
   elseif yb*yc>0
      b=c;
      yb=yc;
   else
      a=c;
      ya=yc;
   end
   dx=min(abs(dx),ac);
   if abs(dx)<delta,break,end
   if abs(yc)<epsilon,break,end
end
c;
err=abs(b-a)/2;
yc=feval(f,c);
```

Exercises for Bracketing Methods

In Exercises 1 and 2, find an approximation for the interest rate I that will yield the total annuity value A if 240 monthly payments P are made. Use the two starting values for I and compute the next three approximations using the bisection method.

1. $P = \$275$, $A = \$250{,}000$, $I_0 = 0.11$, $I_1 = 0.12$

2. $P = \$325$, $A = \$400{,}000$, $I_0 = 0.13$, $I_1 = 0.14$

3. For each function, find an interval $[a, b]$ so that $f(a)$ and $f(b)$ have opposite signs.

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

(b) $f(x) = \cos(x) + 1 - x$

(c) $f(x) = \ln(x) - 5 + x$

(d) $f(x) = x^2 - 10x + 23$

In Exercises 4 through 7, start with $[a_0, b_0]$ and use the false position method to compute c_0, c_1, c_2, and c_3.

4. $e^x - 2 - x = 0$, $[a_0, b_0] = [-2.4, -1.6]$

5. $\cos(x) + 1 - x = 0$, $[a_0, b_0] = [0.8, 1.6]$

6. $\ln(x) - 5 + x = 0$, $[a_0, b_0] = [3.2, 4.0]$

7. $x^2 - 10x + 23 = 0$, $[a_0, b_0] = [6.0, 6.8]$

8. Denote the intervals that arise in the bisection method by $[a_0, b_0]$, $[a_1, b_1]$, $\ldots$, $[a_n, b_n]$.

(a) Show that $a_0 \le a_1 \le \cdots \le a_n \le \cdots$ and that $\cdots \le b_n \le \cdots \le b_1 \le b_0$.

(b) Show that $b_n - a_n = (b_0 - a_0)/2^n$.

(c) Let the midpoint of each interval be $c_n = (a_n + b_n)/2$. Show that

$$\lim_{n\to\infty} a_n = \lim_{n\to\infty} c_n = \lim_{n\to\infty} b_n.$$

Hint. Review convergence of monotone sequences in your calculus book.

9. What will happen if the bisection method is used with the function $f(x) = 1/(x - 2)$ and

(a) the interval is $[3, 7]$? **(b)** the interval is $[1, 7]$?

10. What will happen if the bisection method is used with the function $f(x) = \tan(x)$ and

(a) the interval is $[3, 4]$? **(b)** the interval is $[1, 3]$?

11. Suppose that the bisection method is used to find a zero of $f(x)$ in the interval $[2, 7]$. How many times must this interval be bisected to guarantee that the approximation c_N has an accuracy of 5×10^{-9}?

12. Show that formula (22) for the false position method is algebraically equivalent to

$$c_n = \frac{a_n f(b_n) - b_n f(a_n)}{f(b_n) - f(a_n)}.$$

13. Establish formula (15) for determining the number of iterations required in the bisection method. *Hint.* Use $|b - a|/2^{n+1} < \delta$ and take logarithms.

14. The polynomial $f(x) = (x-1)^3(x-2)(x-3)$ has three zeros: $x = 1$ of multiplicity 3 and $x = 2$ and $x = 3$, each of multiplicity 1. If a_0 and b_0 are any two real numbers such that $a_0 < 1$ and $b_0 > 3$, then $f(a_0)f(b_0) < 0$. Thus, on the interval $[a_0, b_0]$ the bisection method will converge to one of the three zeros. If $a_0 < 1$ and $b_0 > 3$ are selected such that $c_n = (a_n + b_n)/2$ is not equal to 1, 2, or 3 for any $n \geq 1$, then the bisection method will never converge to which zero(s)? Why?

15. If a polynomial, $f(x)$, has an odd number of real zeros in the interval $[a_0, b_0]$, and each of the zeros is of odd multiplicity, then $f(a_0)f(b_0) < 0$, and the bisection method will converge to one of the zeros. If $a_0 < 1$ and $b_0 > 3$ are selected such that $c_n = (a_n + b_n)/2$ is not equal to any of the zeros of $f(x)$ for any $n \geq 1$, then the bisection method will never converge to which zero(s)? Why?

Algorithms and Programs

1. Find an approximation (accurate to 10 decimal places) for the interest rate I that will yield a total annuity value of \$500,000 if 240 monthly payments of \$300 are made.

2. Consider a spherical ball of radius $r = 15$ cm that is constructed from a variety of white oak that has a density of $\rho = 0.710$. How much of the ball (accurate to eight decimal places) will be submerged when it is placed in water?

3. Modify Programs 2.2 and 2.3 to output a matrix analogous to Tables 2.1 and 2.2, respectively (i.e., the first row of the matrix would be $\begin{bmatrix}0 & a_0 & c_0 & b_0 & f(c_0)\end{bmatrix}$).

4. Use your programs from Problem 3 to approximate the three smallest positive roots of $x = \tan(x)$ (accurate to eight decimal places).

5. A unit sphere is cut into two segments by a plane. One segment has three times the volume of the other. Determine the distance x of the plane from the center of the sphere (accurate to 10 decimal places).

2.3 Initial Approximation and Convergence Criteria

The bracketing methods depend on finding an interval $[a, b]$ so that $f(a)$ and $f(b)$ have opposite signs. Once the interval has been found, no matter how large, the iterations will proceed until a root is found. Hence these methods are called ***globally convergent***. However, if $f(x) = 0$ has several roots in $[a, b]$, then a different starting interval must be used to find each root. It is not always easy to locate these smaller intervals on which $f(x)$ changes sign.

In Section 2.4 we develop the Newton-Raphson method and the secant method for solving $f(x) = 0$. Both of these methods require that a close approximation to the root be given to guarantee convergence. Hence these methods are called ***locally convergent***. They usually converge more rapidly than do global ones. Some hybrid algorithms start with a globally convergent method and switch to a locally convergent method when the iteration gets close to a root.

If the computation of roots is one part of a larger project, then a leisurely pace is suggested and the first thing to do is graph the function. We can view the graph $y = f(x)$ and make decisions based on what it looks like (concavity, slope, oscillatory behavior, local extrema, inflection points, etc.). But more important, if the coordinates of points on the graph are available, they can be analyzed and the approximate location of roots determined. These approximations can then be used as starting values in our root-finding algorithms.

We must proceed carefully. Computer software packages use graphics software of varying sophistication. Suppose that a computer is used to graph $y = f(x)$ on $[a, b]$. Typically, the interval is partitioned into $N + 1$ equally spaced points: $a = x_0 < x_1 < \cdots < x_N = b$ and the function values $y_k = f(x_k)$ computed. Then either a line segment or a "fitted curve" is plotted between consecutive points (x_{k-1}, y_{k-1}) and (x_k, y_k) for $k = 1, 2, \ldots, N$. There must be enough points so that we do not miss a root in a portion of the curve where the function is changing rapidly. If $f(x)$ is continuous and two adjacent points (x_{k-1}, y_{k-1}) and (x_k, y_k) lie on opposite sides of the x-axis, then the intermediate value theorem implies that at least one root lies in the interval $[x_{k-1}, x_k]$. But if there is a root, or even several closely spaced roots, in the interval $[x_{k-1}, x_k]$ and the two adjacent points (x_{k-1}, y_{k-1}) and (x_k, y_k) lie on the same side of the x-axis, then the computer-generated graph would not indicate a situation where the intermediate value theorem is applicable. The graph produced by the computer will not be a true representation of the actual graph of the function f. It is not unusual for functions to have "closely" spaced roots; that is, roots where the graph touches but does not cross the x-axis, or roots "close" to a vertical asymptote. Such characteristics of a function need to be considered when applying any numerical root-finding algorithm.

Finally, near two closely spaced roots or near a double root, the computer-generated curve between (x_{k-1}, y_{k-1}) and (x_k, y_k) may fail to cross or touch the x-axis. If $|f(x_k)|$ is smaller than a preassigned value ϵ (i.e.,$f(x_k) \approx 0$), then x_k is a tentative approximate root. But the graph may be close to zero over a wide range of values near x_k, and thus x_k may not be close to an actual root. Hence we add the requirement that the slope change sign near (x_k, y_k); that is, $m_{k-1} = \dfrac{y_k - y_{k-1}}{x_k - x_{k-1}}$ and $m_k = \dfrac{y_{k+1} - y_k}{x_{k+1} - x_k}$ must have opposite signs. Since $x_k - x_{k-1} > 0$ and $x_{k+1} - x_k > 0$, it is not necessary to use the difference quotients, and it will suffice to check to see if the differences $y_k - y_{k-1}$ and $y_{k+1} - y_k$ change sign. In this case, x_k is the approximate root. Unfortunately, we cannot guarantee that this starting value will produce a convergent sequence. If the graph of $y = f(x)$ has a local minimum (or maximum) that is extremely close to zero, then it is possible that x_k will be reported as an approximate root when $f(x_k) \approx 0$, although x_k may not be close to a root.

Table 2.3 Finding Approximate Locations for Roots

	Function values		Differences in y		Significant changes in $f(x)$ or $f'(x)$
x_k	y_{k-1}	y_k	$y_k - y_{k-1}$	$y_{k+1} - y_k$	
−1.2	−3.125	−0.968	2.157	1.329	
−0.9	−0.968	0.361	1.329	0.663	f changes sign in $[x_{k-1}, x_k]$
−0.6	0.361	1.024	0.663	0.159	
−0.3	1.024	1.183	0.159	−0.183	f' changes sign near x_k
0.0	1.183	1.000	−0.183	−0.363	
0.3	1.000	0.637	−0.363	−0.381	
0.6	0.637	0.256	−0.381	−0.237	
0.9	0.256	0.019	−0.237	0.069	f' changes sign near x_k
1.2	0.019	0.088	0.069	0.537	

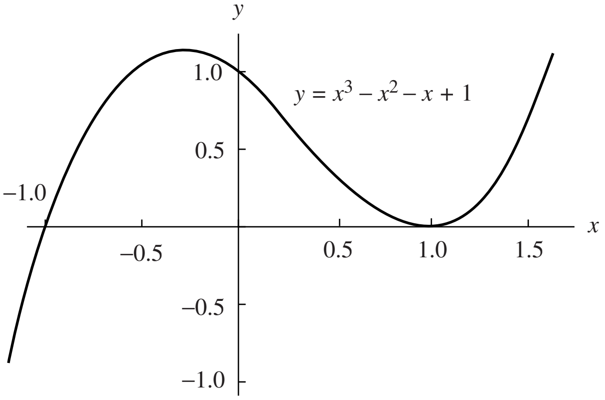


Figure 2.10 The graph of the cubic polynomial $y = x^3 - x^2 - x + 1$.

Example 2.9. Find the approximate location of the roots of $x^3 - x^2 - x + 1 = 0$ on the interval $[-1.2, 1.2]$. For illustration, choose $N = 8$ and look at Table 2.3.

The three abscissas for consideration are -1.05, -0.3, and 0.9. Because $f(x)$ changes sign on the interval $[-1.2, -0.9]$, the value -1.05 is an approximate root; indeed, $f(-1.05) = -0.210$.

Although the slope changes sign near -0.3, we find that $f(-0.3) = 1.183$; hence -0.3 is not near a root. Finally, the slope changes sign near 0.9 and$f(0.9) = 0.019$, so 0.9 is an approximate root (see Figure 2.10). ■

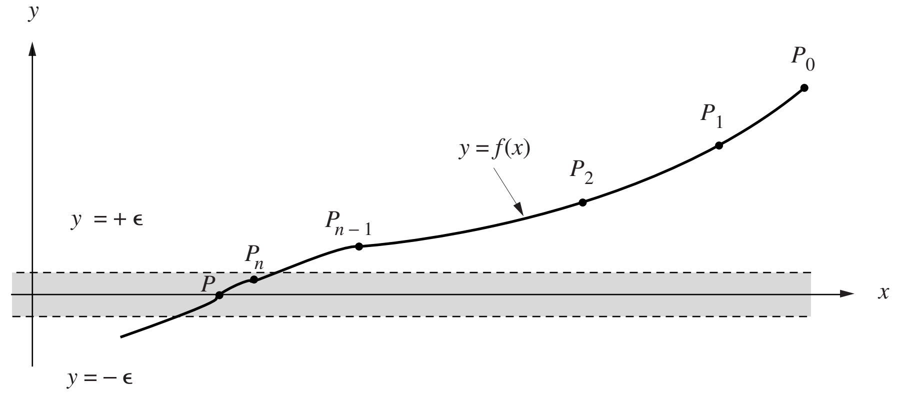


Figure 2.11 (a) The horizontal convergence band for locating a solution to $f(x) = 0$.

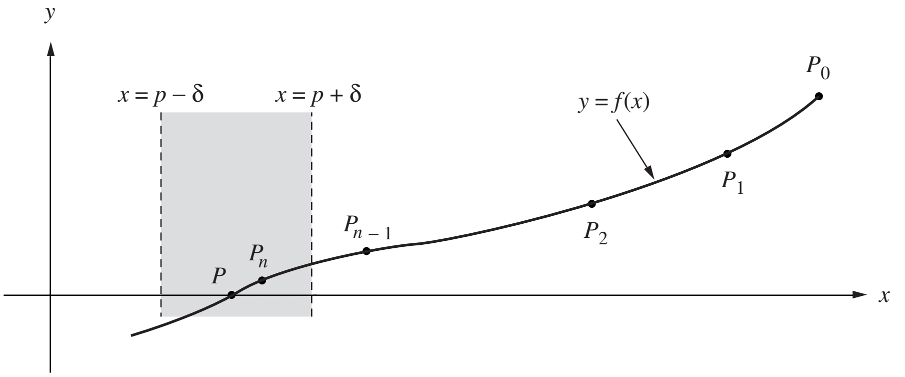


Figure 2.11 (b) The vertical convergence band for locating a solution to $f(x) = 0$.

Checking for Convergence

A graph can be used to see the approximate location of a root, but an algorithm must be used to compute a value p_n that is an acceptable computer solution. Iteration is often used to produce a sequence $\{p_k\}$ that converges to a root p, and a termination criterion or strategy must be designed ahead of time so that the computer will stop when an accurate approximation is reached. Since the goal is to solve $f(x) = 0$, the final value p_n should have the property that $|f(p_n)| < \epsilon$.

The user can supply a tolerance value ϵ for the size of $|f(p_n)|$ and then an iterative process produces points $P_k = (p_k, f(p_k))$ until the last point P_n lies in the horizontal band bounded by the lines $y = +\epsilon$ and $y = -\epsilon$, as shown in Figure 2.11(a). This criterion is useful if the user is trying to solve $h(x) = L$ by applying a root-finding algorithm to the function $f(x) = h(x) - L$.

Another termination criterion involves the abscissas, and we can try to determine if the sequence $\{p_k\}$ is converging. If we draw the vertical lines $x = p + \delta$ and $x = p - \delta$ on each side of $x = p$, we could decide to stop the iteration when the point P_n lies between these two vertical lines, as shown in Figure 2.11(b).

The latter criterion is often desired, but it is difficult to implement because it involves the unknown solution p. We adapt this idea and terminate further calculations when the consecutive iterates p_{n-1} and p_n are sufficiently close or if they agree within M significant digits.

Sometimes the user of an algorithm will be satisfied if $p_n \approx p_{n-1}$ and other times when $f(p_n) \approx 0$. Correct logical reasoning is required to understand the consequences. If we require that $|p_n - p| < \delta$ **and** $|f(p_n)| < \epsilon$, the point P_n will be located in the rectangular region about the solution $(p, 0)$, as shown in Figure 2.12(a). If we stipulate that $|p_n - p| < \delta$ **or** $|f(p_n)| < \epsilon$, the point P_n could be located anywhere in the region formed by the union of the horizontal and vertical stripes, as shown in Figure 2.12(b). The size of the tolerances δ and ϵ are crucial. If the tolerances are chosen too small, iteration may continue forever. They should be chosen about 100 times larger than 10^{-M}, where M is the number of decimal digits in the computer's floating-point numbers. The closeness of the abscissas is checked with one of the criteria

$$|p_n - p_{n-1}| < \delta \qquad \text{(estimate for the absolute error)}$$

or

$$\frac{2|p_n - p_{n-1}|}{|p_n| + |p_{n-1}|} < \delta \qquad \text{(estimate for the relative error).}$$

The closeness of the ordinate is usually checked by $|f(p_n)| < \epsilon$.

Troublesome Functions

A computer solution to $f(x) = 0$ will almost always be in error due to round off and/or instability in the calculations. If the graph $y = f(x)$ is steep near the root $(p, 0)$, then the root-finding problem is well conditioned (i.e., a solution with several significant digits is easy to obtain). If the graph $y = f(x)$ is shallow near $(p, 0)$, then the root-finding problem is ill conditioned (i.e., the computed root may have only a few significant digits). This occurs when $f(x)$ has a multiple root at p. This is discussed further in the next section.

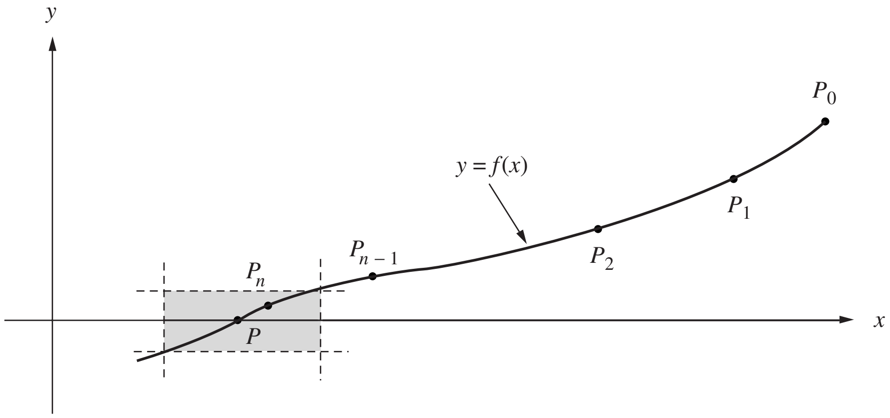


Figure 2.12 (a) The rectangular region defined by $|x - p| < \delta$ AND $|y| < \epsilon$.

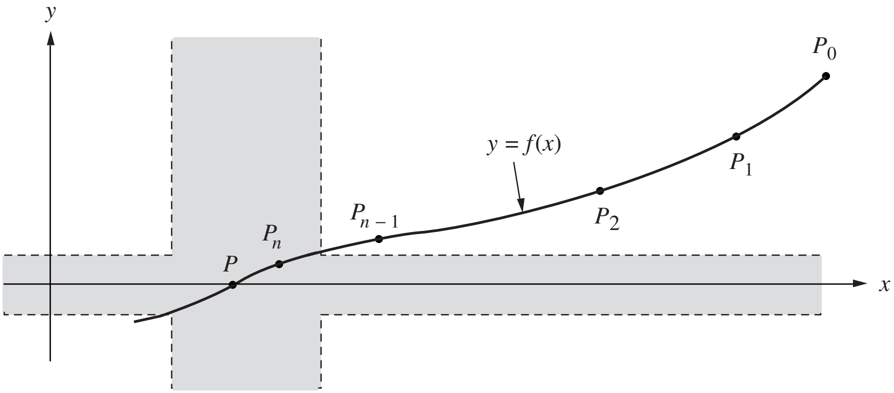


Figure 2.12 (b) The unbounded region defined by $|x - p| < \delta$ OR $|y| < \epsilon$.

Program 2.4 (Approximate Location of Roots). To roughly estimate the locations of the roots of the equation $f(x) = 0$ over the interval $[a, b]$, by using the equally spaced sample points $(x_k, f(x_k))$ and the following criteria:

(i) $(y_{k-1})(y_k) < 0$, or

(ii) $|y_k| < \epsilon$ and $(y_k - y_{k-1})(y_{k+1} - y_k) < 0$.

That is, either $f(x_{k-1})$ and $f(x_k)$ have opposite signs or $|f(x_k)|$ is small and the slope of the curve $y = f(x)$ changes sign near $(x_k, f(x_k))$.

```
function R = approot (X,epsilon)

% Input  - f is the object function saved as an M-file named f.m
%        - X is the vector of abscissas
%        - epsilon is the tolerance
% Output - R is the vector of approximate roots

Y=f(X);
yrange = max(Y)-min(Y);
epsilon2 = yrange*epsilon;
n=length(X);
m=0;
X(n+1)=X(n);
Y(n+1)=Y(n);

for k=2:n,
   if Y(k-1)*Y(k)<=0,
      m=m+1;
      R(m)=(X(k-1)+X(k))/2;
   end
   s=(Y(k)-Y(k-1))*(Y(k+1)-Y(k));
   if (abs(Y(k)) < epsilon2) & (s<=0),
      m=m+1;
      R(m)=X(k);
   end
end
```

Example 2.10. Use `approot` to find approximate locations for the roots of $f(x) = \sin(\cos(x^3))$ in the interval $[-2, 2]$. First save f as an M-file named f.m. Since the results will be used as initial approximations for a root-finding algorithm, we will construct `X` so that the approximations will be accurate to four decimal places.

```
>>X=-2:.001:2;

>>approot (X,0.00001)

ans=

-1.9875 -1.6765 -1.1625 1.1625 1.6765 1.9875
```

Comparing the results with the graph of f, we now have good initial approximations for one of our root-finding algorithms. ■

Exercises for Initial Approximation

In Exercises 1 through 6, use a computer or graphics calculator to graphically determine the approximate location of the roots of $f(x) = 0$ in the given interval. In each case, determine an interval $[a, b]$ over which Programs 2.2 and 2.3 could be used to determine the roots (i.e., $f(a)f(b) < 0$).

1. $f(x) = x^2 - e^x$ for $-2 \le x \le 2$

2. $f(x) = x - \cos(x)$ for $-2 \le x \le 2$

3. $f(x) = \sin(x) - 2\cos(x)$ for $-2 \le x \le 2$

4. $f(x) = \cos(x) + (1 + x^2)^{-1}$ for $-2 \le x \le 2$

5. $f(x) = (x - 2)^2 - \ln(x)$ for $0.5 \le x \le 4.5$

6. $f(x) = 2x - \tan(x)$ for $-1.4 \le x \le 1.4$

Algorithms and Programs

In Problems 1 and 2 use a computer or graphics calculator and Program 2.4 to approximate the real roots, to four decimal places, of each function over the given interval. Then use Program 2.2 or Program 2.3 to approximate each root to 12 decimal places.

1. $f(x) = 1{,}000{,}000x^3 - 111{,}000x^2 + 1110x - 1$ for $-2 \le x \le 2$

2. $f(x) = 5x^{10} - 38x^9 + 21x^8 - 5\pi x^6 - 3\pi x^5 - 5x^2 + 8x - 3$ for $-15 \le x \le 15$.

3. A computer program that plots the graph of $y = f(x)$ over the interval $[a, b]$ using the points $(x_0, y_0), (x_1, y_1), \ldots,$ and (x_N, y_N) usually scales the vertical height of the graph, and a procedure must be written to determine the minimum and maximum values of f over the interval.

(a) Construct an algorithm that will find the values $Y_{\max} = \max_k\{y_k\}$ and $Y_{\min} = \min_k\{y_k\}$.

(b) Write a MATLAB program that will find the approximate location and value of the extreme values of $f(x)$ on the interval $[a, b]$.

(c) Use your program from part (b) to find the approximate location and value of the extreme values of the functions in Problems 1 and 2. Compare your approximations with the actual values.

2.4 Newton-Raphson and Secant Methods

Slope Methods for Finding Roots

If $f(x)$, $f'(x)$, and $f''(x)$ are continuous near a root p, then this extra information regarding the nature of $f(x)$ can be used to develop algorithms that will produce sequences $\{p_k\}$ that converge faster to p than either the bisection or false position method. The Newton-Raphson (or simply Newton's) method is one of the most useful and best known algorithms that relies on the continuity of $f'(x)$ and $f''(x)$. We shall introduce it graphically and then give a more rigorous treatment based on the Taylor polynomial.

Assume that the initial approximation p_0 is near the root p. Then the graph of $y = f(x)$ intersects the x-axis at the point $(p, 0)$, and the point $(p_0, f(p_0))$ lies on the curve near the point $(p, 0)$ (see Figure 2.13). Define p_1 to be the point of intersection of the x-axis and the line tangent to the curve at the point $(p_0, f(p_0))$. Then Figure 2.13 shows that p_1 will be closer to p than p_0 in this case. An equation relating p_1 and p_0 can be found if we write down two versions for the slope of the tangent line L:

$$m = \frac{0 - f(p_0)}{p_1 - p_0}, \tag{1}$$

which is the slope of the line through $(p_1, 0)$ and $(p_0, f(p_0))$, and

$$m = f'(p_0), \tag{2}$$

which is the slope at the point $(p_0, f(p_0))$. Equating the values of the slope m in equations (1) and (2) and solving for p_1 results in

$$p_1 = p_0 - \frac{f(p_0)}{f'(p_0)}. \tag{3}$$

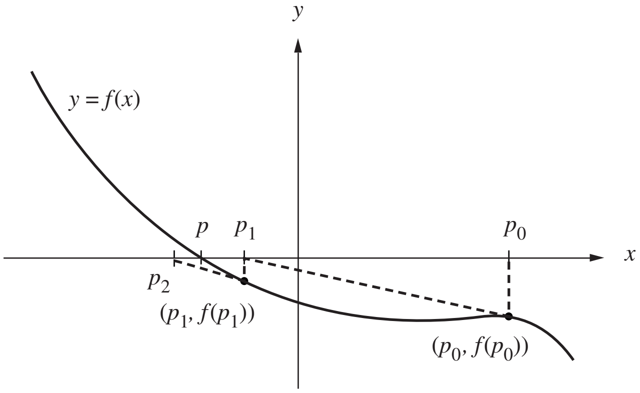


Figure 2.13 The geometric construction of p_1 and p_2 for the Newton-Raphson method.

The process above can be repeated to obtain a sequence $\{p_k\}$ that converges to p. We now make these ideas more precise.

Theorem 2.5 (Newton-Raphson Theorem). Assume that $f \in C^2[a, b]$ and there exists a number $p \in [a, b]$, where $f(p) = 0$. If $f'(p) \neq 0$, then there exists a $\delta > 0$ such that the sequence $\{p_k\}_{k=0}^{\infty}$ defined by the iteration

$$p_k = g(p_{k-1}) = p_{k-1} - \frac{f(p_{k-1})}{f'(p_{k-1})} \quad \text{for } k = 1,\ 2,\ \ldots \tag{4}$$

will converge to p for any initial approximation $p_0 \in [p - \delta, p + \delta]$.

Remark. The function $g(x)$ defined by the formula

$$g(x) = x - \frac{f(x)}{f'(x)} \tag{5}$$

is called the ***Newton-Raphson iteration function***. Since $f(p) = 0$, it is easy to see that $g(p) = p$. Thus the Newton-Raphson iteration for finding the root of the equation $f(x) = 0$ is accomplished by finding a fixed point of the function $g(x)$.

Proof. The geometric construction of p_1 shown in Figure 2.13 does not help in understanding why p_0 needs to be close to p or why the continuity of $f''(x)$ is essential. Our analysis starts with the Taylor polynomial of degree $n = 1$ and its remainder term:

$$f(x) = f(p_0) + f'(p_0)(x - p_0) + \frac{f''(c)(x - p_0)^2}{2!}, \tag{6}$$

where c lies somewhere between p_0 and x. Substituting $x = p$ into equation (6) and using the fact that $f(p) = 0$ produces

$$0 = f(p_0) + f'(p_0)(p - p_0) + \frac{f''(c)(p - p_0)^2}{2!}. \tag{7}$$

If p_0 is close enough to p, the last term on the right side of (7) will be small compared to the sum of the first two terms. Hence it can be neglected and we can use the approximation

$$0 \approx f(p_0) + f'(p_0)(p - p_0). \tag{8}$$

Solving for p in equation (8), we get $p \approx p_0 - f(p_0)/f'(p_0)$. This is used to define the next approximation p_1 to the root

$$p_1 = p_0 - \frac{f(p_0)}{f'(p_0)}. \tag{9}$$

When p_{k-1} is used in place of p_0 in equation (9), the general rule (4) is established. For most applications this is all that needs to be understood. However, to fully comprehend what is happening, we need to consider the fixed-point iteration function and apply Theorem 2.2 in our situation. The key is in the analysis of $g'(x)$:

$$g'(x) = 1 - \frac{f'(x)f'(x) - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2}.$$

By hypothesis, $f(p) = 0$; thus $g'(p) = 0$. Since $g'(p) = 0$ and $g'(x)$ is continuous, it is possible to find a $\delta > 0$ so that the hypothesis $|g'(x)| < 1$ of Theorem 2.2 is satisfied on $(p - \delta, p + \delta)$. Therefore, a sufficient condition for p_0 to initialize a convergent sequence $\{p_k\}_{k=0}^{\infty}$, which converges to a root of $f(x) = 0$, is that $p_0 \in (p - \delta, p + \delta)$ and that δ be chosen so that

$$(10) \qquad \frac{|f(x)f''(x)|}{|f'(x)|^2} < 1 \quad \text{for all } x \in (p - \delta, p + \delta). \qquad \bullet$$

Corollary 2.2 (Newton's Iteration for Finding Square Roots). Assume that $A > 0$ is a real number and let $p_0 > 0$ be an initial approximation to $\sqrt{A}$. Define the sequence $\{p_k\}_{k=0}^{\infty}$ using the recursive rule

$$(11) \qquad p_k = \frac{p_{k-1} + \dfrac{A}{p_{k-1}}}{2} \quad \text{for } k = 1, 2, \ldots.$$

Then the sequence $\{p_k\}_{k=0}^{\infty}$ converges to $\sqrt{A}$; that is, $\lim_{n\to\infty} p_k = \sqrt{A}$.

Outline of Proof. Start with the function $f(x) = x^2 - A$, and notice that the roots of the equation $x^2 - A = 0$ are $\pm\sqrt{A}$. Now use $f(x)$ and the derivative $f'(x)$ in formula (5) and write down the Newton-Raphson iteration formula

$$(12) \qquad g(x) = x - \frac{f(x)}{f'(x)} = x - \frac{x^2 - A}{2x}.$$

This formula can be simplified to obtain

$$(13) \qquad g(x) = \frac{x + \dfrac{A}{x}}{2}.$$

When $g(x)$ in (13) is used to define the recursive iteration in (4), the result is formula (11). It can be proved that the sequence that is generated in (11) will converge for any starting value $p_0 > 0$. The details are left for the exercises. •

An important point of Corollary 2.2 is the fact that the iteration function $g(x)$ involved only the arithmetic operations $+$, $-$, $\times$, and $/$. If $g(x)$ had involved the calculation of a square root, we would be caught in the circular reasoning that being able to calculate the square root would permit you to recursively define a sequence that will converge to $\sqrt{A}$. For this reason, $f(x) = x^2 - A$ was chosen, because it involved only the arithmetic operations.

Example 2.11. Use Newton's square-root algorithm to find $\sqrt{5}$.

Starting with $p_0 = 2$ and using formula (11), we compute

$$\begin{aligned}
p_1 &= \frac{2 + 5/2}{2} = 2.25 \\
p_2 &= \frac{2.25 + 5/2.25}{2} = 2.236111111 \\
p_3 &= \frac{2.236111111 + 5/2.236111111}{2} = 2.236067978 \\
p_4 &= \frac{2.36067978 + 5/2.236067978}{2} = 2.236067978.
\end{aligned}$$

Further iterations produce $p_k \approx 2.236067978$ for $k > 4$, so we see that convergence accurate to nine decimal places has been achieved. ■

Now let us turn to a familiar problem from elementary physics and see why determining the location of a root is an important task. Suppose that a projectile is fired from the origin with an angle of elevation b_0 and initial velocity v_0. In elementary courses, air resistance is neglected and we learn that the height $y = y(t)$ and the distance traveled $x = x(t)$, measured in feet, obey the rules

$$(14) \qquad y = v_y t - 16t^2 \quad \text{and} \quad x = v_x t,$$

where the horizontal and vertical components of the initial velocity are $v_x = v_0 \cos(b_0)$ and $v_y = v_0 \sin(b_0)$, respectively. The mathematical model expressed by the rules in (14) is easy to work with, but tends to give too high an altitude and too long a range for the projectile's path. If we make the additional assumption that the air resistance is proportional to the velocity, the equations of motion become

$$(15) \qquad y = f(t) = (Cv_y + 32C^2)\left(1 - e^{-t/C}\right) - 32Ct$$

and

$$(16) \qquad x = r(t) = Cv_x\left(1 - e^{-t/C}\right),$$

where $C = m/k$ and k is the coefficient of air resistance and m is the mass of the projectile. A larger value of C will result in a higher maximum altitude and a longer range for the projectile. The graph of a flight path of a projectile when air resistance is considered is shown in Figure 2.14. This improved model is more realistic but requires the use of a root-finding algorithm for solving $f(t) = 0$ to determine the elapsed time until the projectile hits the ground. The elementary model in (14) does not require a sophisticated procedure to find the elapsed time.

Table 2.6 Newton's Method Converges Linearly at a Double Root

k	p_k	$p_{k+1} - p_k$	$E_k = p - p_k$	$\dfrac{\lvert E_{k+1}\rvert}{\lvert E_k\rvert}$
0	1.200000000	−0.096969697	−0.200000000	0.515151515
1	1.103030303	−0.050673883	−0.103030303	0.508165253
2	1.052356420	−0.025955609	−0.052356420	0.496751115
3	1.026400811	−0.013143081	−0.026400811	0.509753688
4	1.013257730	−0.006614311	−0.013257730	0.501097775
5	1.006643419	−0.003318055	−0.006643419	0.500550093
⋮	⋮	⋮	⋮	⋮

Example 2.15 (Linear Convergence at a Double Root). Start with $p_0 = 1.2$ and use Newton-Raphson iteration to find the double root $p = 1$ of the polynomial $f(x) = x^3 - 3x + 2$. Using formula (20) to check for linear convergence, we get the values in Table 2.6. ■

Notice that the Newton-Raphson method is converging to the double root, but at a slow rate. The values of $f(p_k)$ in Example 2.15 go to zero faster than the values of $f'(p_k)$, so the quotient $f(p_k)/f'(p_k)$ in formula (4) is defined when $p_k \neq p$. The sequence is converging linearly, and the error is decreasing by a factor of approximately $1/2$ with each successive iteration. The following theorem summarizes the performance of Newton's method on simple and double roots.

Theorem 2.6 (Convergence Rate for Newton-Raphson Iteration). Assume that Newton-Raphson iteration produces a sequence $\{p_n\}_{n=0}^{\infty}$ that converges to the root p of the function $f(x)$. If p is a simple root, convergence is quadratic and

$$(22) \qquad |E_{n+1}| \approx \frac{|f''(p)|}{2|f'(p)|}|E_n|^2 \quad \text{for } n \text{ sufficiently large.}$$

If p is a multiple root of order M, convergence is linear and

$$(23) \qquad |E_{n+1}| \approx \frac{M-1}{M}|E_n| \quad \text{for } n \text{ sufficiently large.}$$

Pitfalls

The division-by-zero error was easy to anticipate, but there are other difficulties that are not so easy to spot. Suppose that the function is $f(x) = x^2 - 4x + 5$; then the sequence $\{p_k\}$ of real numbers generated by formula (4) will wander back and forth from left to right and not converge. A simple analysis of the situation reveals that $f(x) > 0$ and has no real roots.

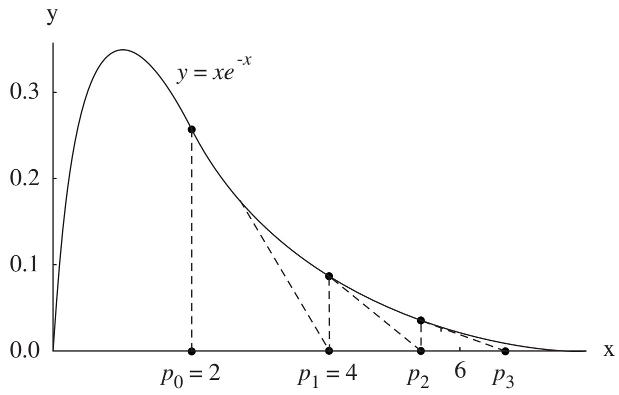


Figure 2.15 (a) Newton-Raphson iteration for $f(x) = xe^{-x}$ can produce a divergent sequence.

Sometimes the initial approximation p_0 is too far away from the desired root and the sequence $\{p_k\}$ converges to some other root. This usually happens when the slope $f'(p_0)$ is small and the tangent line to the curve $y = f(x)$ is nearly horizontal. For example, if $f(x) = \cos(x)$ and we seek the root $p = \pi/2$ and start with $p_0 = 3$, calculation reveals that $p_1 = -4.01525255$, $p_2 = -4.85265757, \ldots$, and $\{p_k\}$ will converge to a different root $-3\pi/2 \approx -4.71238898$.

Suppose that $f(x)$ is positive and monotone decreasing on the unbounded interval $[a, \infty)$ and $p_0 > a$; then the sequence $\{p_k\}$ might diverge to $+\infty$. For example, if $f(x) = xe^{-x}$ and $p_0 = 2.0$, then

$$p_1 = 4.0, \qquad p_2 = 5.333333333, \qquad \ldots, \qquad p_{15} = 19.723549434, \qquad \ldots,$$

and $\{p_k\}$ diverges slowly to $+\infty$ (see Figure 2.15(a)). This particular function has another surprising problem. The value of $f(x)$ goes to zero rapidly as x gets large, for example, $f(p_{15}) = 0.0000000536$, and it is possible that p_{15} could be mistaken for a root. For this reason we designed the stopping criterion in Program 2.5 to involve the relative error $2|p_{k+1} - p_k|/(|p_k|+10^{-6})$, and when $k = 15$, this value is 0.106817, so the tolerance $\delta = 10^{-6}$ will help guard against reporting a false root.

Another phenomenon, ***cycling***, occurs when the terms in the sequence $\{p_k\}$ tend to repeat or almost repeat. For example, if $f(x) = x^3 - x - 3$ and the initial approximation is $p_0 = 0$, then the sequence is

$$\begin{array}{llll} p_1 = -3.000000, & p_2 = -1.961538, & p_3 = -1.147176, & p_4 = -0.006579, \\ p_5 = -3.000389, & p_6 = -1.961818, & p_7 = -1.147430, & \ldots \end{array}$$

and we are stuck in a cycle where $p_{k+4} \approx p_k$ for $k = 0, 1, \ldots$ (see Figure 2.15(b)). But if the starting value p_0 is sufficiently close to the root $p \approx 1.671699881$, then $\{p_k\}$

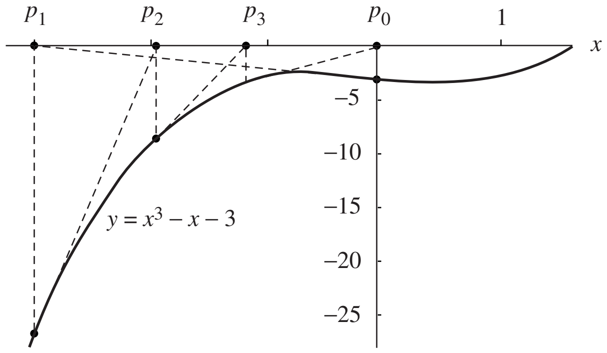


Figure 2.15 (b) Newton-Raphson iteration for $f(x) = x^3 - x - 3$ can produce a cyclic sequence.

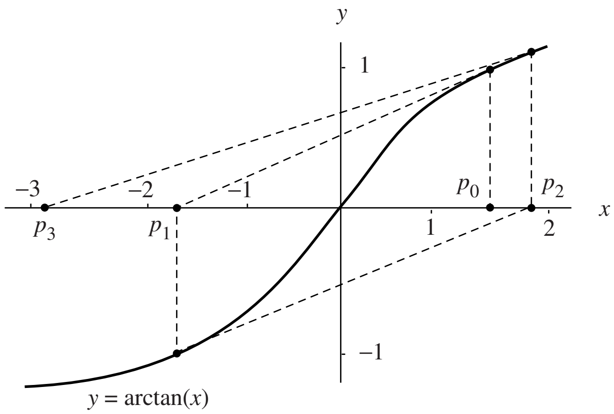


Figure 2.15 (c) Newton-Raphson iteration for $f(x) = \arctan(x)$ can produce a divergent oscillating sequence.

converges. If $p_0 = 2$, the sequence converges: $p_1 = 1.72727272$, $p_2 = 1.67369173$, $p_3 = 1.671702570$, and $p_4 = 1.671699881$.

When $|g'(x)| \geq 1$ on an interval containing the root p, there is a chance of divergent oscillation. For example, let $f(x) = \arctan(x)$; then the Newton-Raphson iteration function is $g(x) = x - (1 + x^2)\arctan(x)$, and $g'(x) = -2x\arctan(x)$. If the starting value $p_0 = 1.45$ is chosen, then

$$p_1 = -1.550263297, \qquad p_2 = 1.845931751, \qquad p_3 = -2.889109054,$$

etc. (see Figure 2.15(c)). But if the starting value is sufficiently close to the root $p = 0$,

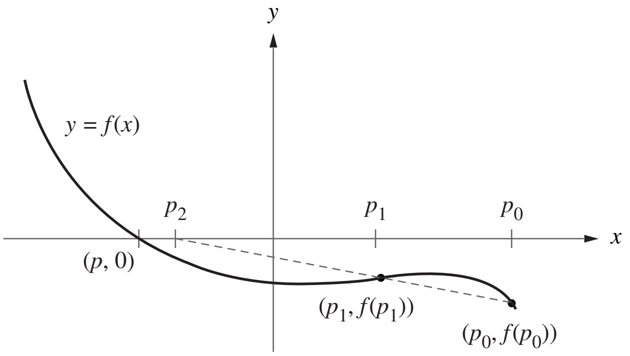


Figure 2.16 The geometric construction of p_2 for the secant method.

a convergent sequence results. If $p_0 = 0.5$, then

$$p_1 = -0.079559511, \qquad p_2 = 0.000335302, \qquad p_3 = 0.000000000.$$

The situations above point to the fact that we must be honest in reporting an answer. Sometimes the sequence does not converge. It is not always the case that after N iterations a solution is found. The user of a root-finding algorithm needs to be warned of the situation when a root is not found. If there is other information concerning the context of the problem, then it is less likely that an erroneous root will be found. Sometimes $f(x)$ has a definite interval in which a root is meaningful. If knowledge of the behavior of the function or an "accurate" graph is available, then it is easier to choose p_0.

Secant Method

The Newton-Raphson algorithm requires the evaluation of two functions per iteration, $f(p_{k-1})$ and $f'(p_{k-1})$. Traditionally, the calculation of derivatives of elementary functions could involve considerable effort. But with modern computer algebra software packages, this has become less of an issue. Still many functions have nonelementary forms (integrals, sums, etc.), and it is desirable to have a method that converges almost as fast as Newton's method yet involves only evaluations of $f(x)$ and not of $f'(x)$. The secant method will require only one evaluation of $f(x)$ per step and at a simple root has an order of convergence $R \approx 1.618033989$. It is almost as fast as Newton's method, which has order 2.

The formula involved in the secant method is the same one that was used in the regula falsi method, except that the logical decisions regarding how to define each succeeding term are different. Two initial points $(p_0, f(p_0))$ and $(p_1, f(p_1))$ near the point $(p, 0)$ are needed, as shown in Figure 2.16. Define p_2 to be the abscissa

Table 2.7 Convergence of the Secant Method at a Simple Root

k	p_k	$p_{k+1} - p_k$	$E_k = p - p_k$	$\dfrac{\lvert E_{k+1}\rvert}{\lvert E_k\rvert^{1.618}}$
0	−2.600000000	0.200000000	0.600000000	0.914152831
1	−2.400000000	0.293401015	0.400000000	0.469497765
2	−2.106598985	0.083957573	0.106598985	0.847290012
3	−2.022641412	0.021130314	0.022641412	0.693608922
4	−2.001511098	0.001488561	0.001511098	0.825841116
5	−2.000022537	0.000022515	0.000022537	0.727100987
6	−2.000000022	0.000000022	0.000000022	
7	−2.000000000	0.000000000	0.000000000	

of the point of intersection of the line through these two points and the x-axis; then Figure 2.16 shows that p_2 will be closer to p than to either p_0 or p_1. The equation relating p_2, p_1, and p_0 is found by considering the slope

$$(24) \qquad m = \frac{f(p_1) - f(p_0)}{p_1 - p_0} \quad \text{and} \quad m = \frac{0 - f(p_1)}{p_2 - p_1}.$$

The values of m in (25) are the slope of the secant line through the first two approximations and the slope of the line through $(p_1, f(p_1))$ and $(p_2, 0)$, respectively. Set the right-hand sides equal in (25) and solve for $p_2 = g(p_1, p_0)$ and get

$$(25) \qquad p_2 = g(p_1, p_0) = p_1 - \frac{f(p_1)(p_1 - p_0)}{f(p_1) - f(p_0)}.$$

The general term is given by the two-point iteration formula

$$(26) \qquad p_{k+1} = g(p_k, p_{k-1}) = p_k - \frac{f(p_k)(p_k - p_{k-1})}{f(p_k) - f(p_{k-1})}.$$

Example 2.16 (Secant Method at a Simple Root). Start with $p_0 = -2.6$ and $p_1 = -2.4$ and use the secant method to find the root $p = -2$ of the polynomial function $f(x) = x^3 - 3x + 2$.

In this case the iteration formula (27) is

$$(27) \qquad p_{k+1} = g(p_k, p_{k-1}) = p_k - \frac{(p_k^3 - 3p_k + 2)(p_k - p_{k-1})}{p_k^3 - p_{k-1}^3 - 3p_k + 3p_{k-1}}.$$

This can be algebraically manipulated to obtain

$$(28) \qquad p_{k+1} = g(p_k, p_{k-1}) = \frac{p_k^2 p_{k-1} + p_k p_{k-1}^2 - 2}{p_k^2 + p_k p_{k-1} + p_{k-1}^2 - 3}.$$

The sequence of iterates is given in Table 2.7. ■

There is a relationship between the secant method and Newton's method. For a polynomial function $f(x)$, the secant method two-point formula $p_{k+1} = g(p_k, p_{k-1})$ will reduce to Newton's one-point formula $p_{k+1} = g(p_k)$ if p_k is replaced by p_{k-1}. Indeed, if we replace p_k by p_{k-1} in (29), then the right side becomes the same as the right side of (22) in Example 2.14.

Proofs about the rate of convergence of the secant method can be found in advanced texts on numerical analysis. Let us state that the error terms satisfy the relationship

$$|E_{k+1}| \approx |E_k|^{1.618} \left| \frac{f''(p)}{2f'(p)} \right|^{0.618} \tag{29}$$

where the order of convergence is $R = (1 + \sqrt{5})/2 \approx 1.618$ and the relation in (30) is valid only at simple roots.

To check this, we make use of Example 2.16 and the specific values

$$|p - p_5| = 0.000022537$$
$$|p - p_4|^{1.618} = 0.001511098^{1.618} = 0.000027296,$$

and

$$A = |f''(-2)/2f'(-2)|^{0.618} = (2/3)^{0.618} = 0.778351205.$$

Combine these and it is easy to see that

$$|p - p_5| = 0.000022537 \approx 0.000021246 = A|p - p_4|^{1.618}.$$

Accelerated Convergence

We could hope that there are root-finding techniques that converge faster than linearly when p is a root of order M. Our final result shows that a modification can be made to Newton's method so that convergence becomes quadratic at a multiple root.

Theorem 2.7 (Acceleration of Newton-Raphson Iteration). Suppose that the Newton-Raphson algorithm produces a sequence that converges linearly to the root $x = p$ of order $M > 1$. Then the Newton-Raphson iteration formula

$$p_k = p_{k-1} - \frac{Mf(p_{k-1})}{f'(p_{k-1})} \tag{30}$$

will produce a sequence $\{p_k\}_{k=0}^{\infty}$ that converges quadratically to p.

```
   p2=p1-feval(f,p1)*(p1-p0)/(feval(f,p1)-feval(f,p0));
   err=abs(p2-p1);
   relerr=2*err/(abs(p2)+delta);
   p0=p1;
   p1=p2;
   y=feval(f,p1);
   if (err<delta)|(relerr<delta)|(abs(y)<epsilon),break,end
end
```

Exercises for Newton-Raphson and Secant Methods

For problems involving calculations, you can use either a calculator or a computer.

1. Let $f(x) = x^2 - x + 2$.

(a) Find the Newton-Raphson formula $p_k = g(p_{k-1})$.

(b) Start with $p_0 = -1.5$ and find p_1, p_2, and p_3.

2. Let $f(x) = x^2 - x - 3$.

(a) Find the Newton-Raphson formula $p_k = g(p_{k-1})$.

(b) Start with $p_0 = 1.6$ and find p_1, p_2, and p_3.

(c) Start with $p_0 = 0.0$ and find p_1, p_2, p_3, and p_4. What do you conjecture about this sequence?

3. Let $f(x) = (x - 2)^4$.

(a) Find the Newton-Raphson formula $p_k = g(p_{k-1})$.

(b) Start with $p_0 = 2.1$ and find p_1, p_2, p_3, and p_4.

(c) Is the sequence converging quadratically or linearly?

4. Let $f(x) = x^3 - 3x - 2$.

(a) Find the Newton-Raphson formula $p_k = g(p_{k-1})$.

(b) Start with $p_0 = 2.1$ and find p_1, p_2, p_3, and p_4.

(c) Is the sequence converging quadratically or linearly?

5. Consider the function $f(x) = \cos(x)$.

(a) Find the Newton-Raphson formula $p_k = g(p_{k-1})$.

(b) We want to find the root $p = 3\pi/2$. Can we use $p_0 = 3$? Why?

(c) We want to find the root $p = 3\pi/2$. Can we use $p_0 = 5$? Why?

6. Consider the function $f(x) = \arctan(x)$.

(a) Find the Newton-Raphson formula $p_k = g(p_{k-1})$.

(b) If $p_0 = 1.0$, then find p_1, p_2, p_3, and p_4. What is $\lim_{n\to\infty} p_k$?

(c) If $p_0 = 2.0$, then find p_1, p_2, p_3, and p_4. What is $\lim_{n\to\infty} p_k$?

7. Consider the function $f(x) = xe^{-x}$.

(a) Find the Newton-Raphson formula $p_k = g(p_{k-1})$.

(b) If $p_0 = 0.2$, then find p_1, p_2, p_3, and p_4. What is $\lim_{n\to\infty} p_k$?

(c) If $p_0 = 20$, then find p_1, p_2, p_3, and p_4. What is $\lim_{n\to\infty} p_k$?

(d) What is the value of $f(p_4)$ in part (c)?

In Exercises 8 through 10, use the secant method and formula (27) and compute the next two iterates p_2 and p_3.

8. Let $f(x) = x^2 - 2x - 1$. Start with $p_0 = 2.6$ and $p_1 = 2.5$.

9. Let $f(x) = x^2 - x - 3$. Start with $p_0 = 1.7$ and $p_1 = 1.67$.

10. Let $f(x) = x^3 - x + 2$. Start with $p_0 = -1.5$ and $p_1 = -1.52$.

11. *Cube-root algorithm.* Start with $f(x) = x^3 - A$, where A is any real number, and derive the recursive formula

$$p_k = \frac{2p_{k-1} + A/p_{k-1}^2}{3} \quad \text{for } k = 1, 2, \ldots.$$

12. Consider $f(x) = x^N - A$, where N is a positive integer.

(a) What real values are the solution to $f(x) = 0$ for the various choices of N and A that can arise?

(b) Derive the recursive formula

$$p_k = \frac{(N-1)p_{k-1} + A/p_{k-1}^{N-1}}{N} \quad \text{for } k = 1, 2, \ldots.$$

for finding the Nth root of A.

13. Can Newton-Raphson iteration be used to solve $f(x) = 0$ if $f(x) = x^2 - 14x + 50$? Why?

14. Can Newton-Raphson iteration be used to solve $f(x) = 0$ if $f(x) = x^{1/3}$? Why?

15. Can Newton-Raphson iteration be used to solve $f(x) = 0$ if $f(x) = (x-3)^{1/2}$ and the starting value is $p_0 = 4$? Why?

16. Establish the limit of the sequence in (11).

17. Prove that the sequence $\{p_k\}$ in equation (4) of Theorem 2.5 converges to p. Use the following steps.

(a) Show that if p is a fixed point of $g(x)$ in equation (5), then p is a zero of $f(x)$.

(b) If p is a zero of $f(x)$ and $f'(p) \neq 0$, show that $g'(p) = 0$. Use part (b) and Theorem 2.3 to show that the sequence $\{p_k\}$ in equation (4) converges to p.

18. Prove equation (23) of Theorem 2.6. Use the following steps. By Theorem 1.11, we can expand $f(x)$ about $x = p_k$ to get

$$f(x) = f(p_k) + f'(p_k)(x - p_k) + \frac{1}{2}f''(c_k)(x - p_k)^2.$$

Exercises for Introduction to Vectors and Matrices

The reader is encouraged to carry out the following exercises by hand and with MATLAB.

1. Given the vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$, find **(a)** $\boldsymbol{X}+\boldsymbol{Y}$, **(b)** $\boldsymbol{X}-\boldsymbol{Y}$, **(c)** $3\boldsymbol{X}$, **(d)** $\|\boldsymbol{X}\|$, **(e)** $7\boldsymbol{Y}-4\boldsymbol{X}$, **(f)** $\boldsymbol{X}\cdot\boldsymbol{Y}$, and **(g)** $\|7\boldsymbol{Y}-4\boldsymbol{X}\|$.

(i) $\boldsymbol{X} = (3, -4)$ and $\boldsymbol{Y} = (-2, 8)$

(ii) $\boldsymbol{X} = (-6, 3, 2)$ and $\boldsymbol{Y} = (-8, 5, 1)$

(iii) $\boldsymbol{X} = (4, -8, 1)$ and $\boldsymbol{Y} = (1, -12, -11)$

(iv) $\boldsymbol{X} = (1, -2, 4, 2)$ and $\boldsymbol{Y} = (3, -5, -4, 0)$

2. Using the law of cosines, it can be shown that the angle θ between two vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ is given by the relation

$$\cos(\theta) = \frac{\boldsymbol{X}\cdot\boldsymbol{Y}}{\|\boldsymbol{X}\|\,\|\boldsymbol{Y}\|}.$$

Find the angle, in radians, between the following vectors:

(a) $\boldsymbol{X} = (-6, 3, 2)$ and $\boldsymbol{Y} = (2, -2, 1)$

(b) $\boldsymbol{X} = (4, -8, 1)$ and $\boldsymbol{Y} = (3, 4, 12)$

3. Two vectors $\boldsymbol{X}$and $\boldsymbol{Y}$ are said to be orthogonal (perpendicular) if the angle between them is $\pi/2$.

(a) Prove that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are orthogonal if and only if $\boldsymbol{X}\cdot\boldsymbol{Y} = 0$.

Use part (a) to determine if the following vectors are orthogonal.

(b) $\boldsymbol{X} = (-6, 4, 2)$ and $\boldsymbol{Y} = (6, 5, 8)$

(c) $\boldsymbol{X} = (-4, 8, 3)$ and $\boldsymbol{Y} = (2, 5, 16)$

(d) $\boldsymbol{X} = (-5, 7, 2)$ and $\boldsymbol{Y} = (4, 1, 6)$

(e) Find two different vectors that are orthogonal to $\boldsymbol{X} = (1, 2, -5)$.

4. Find **(a)** $\boldsymbol{A}+\boldsymbol{B}$, **(b)** $\boldsymbol{A}-\boldsymbol{B}$, and **(c)** $3\boldsymbol{A}-2\boldsymbol{B}$ for the matrices

$$\boldsymbol{A} = \begin{bmatrix} -1 & 9 & 4 \\ 2 & -3 & -6 \\ 0 & 5 & 7 \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} -4 & 9 & 2 \\ 3 & -5 & 7 \\ 8 & 1 & -6 \end{bmatrix}.$$

5. The ***transpose*** of an $M \times N$ matrix $\boldsymbol{A}$, denoted $\boldsymbol{A}'$, is the $N \times M$ matrix obtained from $\boldsymbol{A}$ by converting the rows of $\boldsymbol{A}$ to columns of $\boldsymbol{A}'$. That is, if $\boldsymbol{A} = [a_{ij}]_{M\times N}$ and $\boldsymbol{A}' = [b_{ij}]_{N\times M}$, then the elements satisfy the relation

$$b_{ji} = a_{ij} \qquad \text{for} \qquad 1 \le i \le M, 1 \le j \le N.$$

Find the transpose of the following matrices.

(a) $\begin{bmatrix} -2 & 5 & 12 \\ 1 & 4 & -1 \\ 7 & 0 & 6 \\ 11 & -3 & 8 \end{bmatrix}$ **(b)** $\begin{bmatrix} 4 & 9 & 2 \\ 3 & 5 & 7 \\ 8 & 1 & 6 \end{bmatrix}$

Exercises for Properties of Vectors and Matrices

The reader is encouraged to carry out the following exercises by hand and with MATLAB.

1. Find $\boldsymbol{AB}$ and $\boldsymbol{BA}$ for the following matrices:

$$\boldsymbol{A} = \begin{bmatrix} -3 & 2 \\ 1 & 4 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 5 & 0 \\ 2 & -6 \end{bmatrix}.$$

2. Find $\boldsymbol{AB}$ and $\boldsymbol{BA}$ for the following matrices.

$$\boldsymbol{A} = \begin{bmatrix} 1 & -2 & 3 \\ 2 & 0 & 5 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 3 & 0 \\ -1 & 5 \\ 3 & -2 \end{bmatrix}.$$

3. Let $\boldsymbol{A}$, $\boldsymbol{B}$,and $\boldsymbol{C}$ be given by

$$\boldsymbol{A} = \begin{bmatrix} 3 & 1 \\ 0 & 4 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 & 2 \\ -2 & -6 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 2 & -5 \\ 3 & 4 \end{bmatrix}.$$

(a) Find $(\boldsymbol{AB})\boldsymbol{C}$ and $\boldsymbol{A}(\boldsymbol{BC})$.

(b) Find $\boldsymbol{A}(\boldsymbol{B} + \boldsymbol{C})$ and $\boldsymbol{AB} + \boldsymbol{AC}$.

(c) Find $(\boldsymbol{A} + \boldsymbol{B})\boldsymbol{C}$ and $\boldsymbol{AC} + \boldsymbol{BC}$.

(d) Find $(\boldsymbol{AB})'$ and $\boldsymbol{B}'\boldsymbol{A}'$.

4. We use the notation $\boldsymbol{A}^2 = \boldsymbol{AA}$. Find $\boldsymbol{A}^2$ and $\boldsymbol{B}^2$ for the following matrices:

$$\boldsymbol{A} = \begin{bmatrix} -1 & -7 \\ 5 & 2 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 2 & 0 & 6 \\ -1 & 5 & -4 \\ 3 & -5 & 2 \end{bmatrix}$$

5. Find the determinant of the following matrices, if it exists.

(a) $\begin{bmatrix} -1 & -7 \\ 5 & 2 \end{bmatrix}$

(b) $\begin{bmatrix} 2 & 0 & 6 \\ -1 & 5 & -4 \\ 3 & -5 & 2 \end{bmatrix}$

(c) $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 0 \end{bmatrix}$

(d) $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 2 & 4 & 6 \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 0 & 7 \end{bmatrix}$

6. Show that $\boldsymbol{R}_x(\alpha)\boldsymbol{R}_x(-\alpha) = \boldsymbol{I}$ by direct multiplication of the matrices $\boldsymbol{R}_x(\alpha)$ and $\boldsymbol{R}_x(-\alpha)$ (see formula (26)).

7. (a) Show that

$$\boldsymbol{R}_x(\alpha)\boldsymbol{R}_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ \sin(\beta)\sin(\alpha) & \cos(\alpha) & -\cos(\beta)\sin(\alpha) \\ -\cos(\alpha)\sin(\beta) & \sin(\alpha) & \cos(\beta)\cos(\alpha) \end{bmatrix}$$

(see formulas (26) and (27)).

```
X(n)=B(n)/A(n,n);
for k=n-1:-1:1
   X(k)=(B(k)-A(k,k+1:n)*X(k+1:n))/A(k,k);
end
```

Exercises for Upper-Triangular Linear Systems

In Exercises 1 through 3, solve the upper-triangular system and find the value of the determinant of the coefficient matrix.

1.
$$\begin{aligned} 3x_1 - 2x_2 + x_3 - x_4 &= 8 \\ 4x_2 - x_3 + 2x_4 &= -3 \\ 2x_3 + 3x_4 &= 11 \\ 5x_4 &= 15 \end{aligned}$$

2.
$$\begin{aligned} 5x_1 - 3x_2 - 7x_3 + x_4 &= -14 \\ 11x_2 + 9x_3 + 5x_4 &= 22 \\ 3x_3 - 13x_4 &= -11 \\ 7x_4 &= 14 \end{aligned}$$

3.
$$\begin{aligned} 4x_1 - x_2 + 2x_3 + 2x_4 - x_5 &= 4 \\ -2x_2 + 6x_3 + 2x_4 + 7x_5 &= 0 \\ x_3 - x_4 - 2x_5 &= 3 \\ - 2x_4 - x_5 &= 10 \\ 3x_5 &= 6 \end{aligned}$$

4. (a) Consider the two upper-triangular matrices

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} \quad \text{and} \quad \boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ 0 & b_{22} & b_{23} \\ 0 & 0 & b_{33} \end{bmatrix}.$$

Show that their product $\boldsymbol{C} = \boldsymbol{AB}$ is also upper triangular.

(b) Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two $N \times N$ upper-triangular matrices. Show that their product is also upper triangular.

5. Solve the lower-triangular system $\boldsymbol{AX} = \boldsymbol{B}$ and find det($\boldsymbol{A}$).

$$\begin{aligned} 2x_1 \qquad\qquad\qquad &= 6 \\ -x_1 + 4x_2 \qquad\qquad &= 5 \\ 3x_1 - 2x_2 - x_3 \qquad &= 4 \\ x_1 - 2x_2 + 6x_3 + 3x_4 &= 2 \end{aligned}$$

6. Solve the lower-triangular system $\boldsymbol{AX} = \boldsymbol{B}$ and find det($\boldsymbol{A}$).

$$\begin{aligned} 5x_1 \qquad\qquad\qquad &= -10 \\ x_1 + 3x_2 \qquad\qquad &= 4 \\ 3x_1 + 4x_2 + 2x_3 \qquad &= 2 \\ -x_1 + 3x_2 - 6x_3 - x_4 &= 5 \end{aligned}$$

7. Show that back substitution requires N divisions, $(N^2 - N)/2$ multiplications, and $(N^2 - N)/2$ additions or subtractions. *Hint.* You can use the formula

$$\sum_{k=1}^{M} k = M(M + 1)/2.$$

Algorithms and Programs

1. Use Program 3.1 to solve the system $\boldsymbol{UX} = \boldsymbol{B}$, where

$$\boldsymbol{U} = [u_{ij}]_{10\times 10} \quad \text{and} \quad u_{ij} = \begin{cases} \cos(ij) & i \le j, \\ 0 & i > j. \end{cases}$$

and $\boldsymbol{B} = [b_{i1}]_{10\times 1}$ and $b_{i1} = \tan(i)$.

2. *Forward-substitution algorithm.* A linear system $\boldsymbol{AX} = \boldsymbol{B}$ is called lower triangular provided that $a_{ij} = 0$ when $i < j$. Construct a program `forsub`, analogous to Program 3.1, to solve the following lower-triangular system. *Remark.* This program will be used in Section 3.5.

$$\begin{array}{llllll}
a_{11}x_1 & & & & & = b_1 \\
a_{21}x_1 + & a_{22}x_2 & & & & = b_2 \\
a_{31}x_1 + & a_{32}x_2 + & a_{33}x_3 & & & = b_3 \\
\vdots & \vdots & \vdots & & & \vdots \\
a_{N-1\,1}x_1 + & a_{N-1\,2}x_2 + & a_{N-1\,3}x_3 + \cdots + & a_{N-1\,N-1}x_{N-1} & & = b_{N-1} \\
a_{N\,1}x_1 + & a_{N\,2}x_2 + & a_{N\,3}x_3 + \cdots + & a_{N\,N-1}x_{N-1} + & a_{NN}x_N & = b_N
\end{array}$$

3. Use `forsub` to solve the system $\boldsymbol{LX} = \boldsymbol{B}$, where

$$\boldsymbol{L} = [l_{ij}]_{20\times 20} \quad \text{and} \quad l_{ij} = \begin{cases} i + j & i \ge j, \\ 0 & i < j, \end{cases} \quad \text{and} \quad \boldsymbol{B} = [b_{i1}]_{20\times 1} \quad \text{and} \quad b_{i1} = i.$$

3.4 Gaussian Elimination and Pivoting

In this section we develop a scheme for solving a general system $\boldsymbol{AX} = \boldsymbol{B}$ of N equations and N unknowns. The goal is to construct an equivalent upper-triangular system $\boldsymbol{UX} = \boldsymbol{Y}$ that can be solved by the method of Section 3.3.

Two linear systems of dimension $N \times N$ are said to be ***equivalent*** provided that their solution sets are the same. Theorems from linear algebra show that when certain transformations are applied to a given system, the solution sets do not change.

Theorem 3.7 (Elementary Transformations). The following operations applied to a linear system yield an equivalent system:

(1) Interchanges: The order of two equations can be changed.

(2) Scaling: Multiplying an equation by a nonzero constant.

(3) Replacement: An equation can be replaced by the sum of itself and a nonzero multiple of any other equation.

It is common to use (3) by replacing an equation with the difference of that equation and a multiple of another equation. These concepts are illustrated in the next example.

Example 3.15. Find the parabola $y = A + Bx + Cx^2$ that passes through the three points $(1, 1)$, $(2, -1)$, and $(3, 1)$.

For each point we obtain an equation relating the value of x to the value of y. The result is the linear system

$$\begin{array}{rcrl}
A + B + C & = & 1 & \quad \text{at } (1, 1) \\
A + 2B + 4C & = & -1 & \quad \text{at } (2, -1) \\
A + 3B + 9C & = & 1 & \quad \text{at } (3, 1).
\end{array} \tag{4}$$

The variable A is eliminated from the second and third equations by subtracting the first equation from them. This is an application of the replacement transformation (3), and the resulting equivalent linear system is

$$\begin{array}{rcr}
A + B + C & = & 1 \\
B + 3C & = & -2 \\
2B + 8C & = & 0.
\end{array} \tag{5}$$

The variable B is eliminated from the third equation in (5) by subtracting from it two times the second equation. We arrive at the equivalent upper-triangular system:

$$\begin{array}{rcr}
A + B + C & = & 1 \\
B + 3C & = & -2 \\
2C & = & 4.
\end{array} \tag{6}$$

The back-substitution algorithm is now used to find the coefficients $C = 4/2 = 2$, $B = -2 - 3(2) = -8$, and $A = 1 - (-8) - 2 = 7$, and the equation of the parabola is $y = 7 - 8x + 2x^2$. ■

It is efficient to store all the coefficients of the linear system $\boldsymbol{AX} = \boldsymbol{B}$ in an array of dimension $N \times (N + 1)$. The coefficients of $\boldsymbol{B}$ are stored in column $N + 1$ of the array (i.e., $a_{kN+1} = b_k$). Each row contains all the coefficients necessary to represent an equation in the linear system. The ***augmented matrix*** is denoted $[\boldsymbol{A}|\boldsymbol{B}]$ and the linear system is represented as follows:

$$[\boldsymbol{A}|\boldsymbol{B}] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1N} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2N} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} & b_N \end{array}\right]. \tag{7}$$

The system $\boldsymbol{AX} = \boldsymbol{B}$, with augmented matrix given in (7), can be solved by performing row operations on the augmented matrix $[\boldsymbol{A}|\boldsymbol{B}]$. The variables x_k are placeholders for the coefficients and can be omitted until the end of the calculation.

Theorem 3.8 (Elementary Row Operations). The following operations applied to the augmented matrix (7) yield an equivalent linear system.

(8) Interchanges: The order of two rows can be changed.

(9) Scaling: Multiplying a row by a nonzero constant.

(10) Replacement: The row can be replaced by the sum of that row and
a nonzero multiple of any other row; that is:
$\text{row}_r = \text{row}_r - m_{rp} \times \text{row}_p$.

It is common to use (10) by replacing a row with the difference of that row and a multiple of another row.

Definition 3.3. The number a_{rr} in the coefficient matrix $\boldsymbol{A}$ that is used to eliminate a_{kr}, where $k = r + 1, r + 2, \ldots, N$, is called the rth ***pivotal element***, and the rth row is called the ***pivot row***. ▲

The following example illustrates how to use the operations in Theorem 3.8 to obtain an equivalent upper-triangular system $\boldsymbol{UX} = \boldsymbol{Y}$ from a linear system $\boldsymbol{AX} = \boldsymbol{B}$, where $\boldsymbol{A}$ is an $N \times N$ matrix.

Example 3.16. Express the following system in augmented matrix form and find an equivalent upper-triangular system and the solution.

$$\begin{aligned} x_1 + 2x_2 + x_3 + 4x_4 &= 13 \\ 2x_1 + 0x_2 + 4x_3 + 3x_4 &= 28 \\ 4x_1 + 2x_2 + 2x_3 + x_4 &= 20 \\ -3x_1 + x_2 + 3x_3 + 2x_4 &= 6. \end{aligned}$$

The augmented matrix is

$$\begin{array}{l} \text{pivot} \rightarrow \\ m_{21} = 2 \\ m_{31} = 4 \\ m_{41} = -3 \end{array} \left[\begin{array}{rrrr|r} \underline{1} & 2 & 1 & 4 & 13 \\ 2 & 0 & 4 & 3 & 28 \\ 4 & 2 & 2 & 1 & 20 \\ -3 & 1 & 3 & 2 & 6 \end{array}\right].$$

```
for p=1:N-1
   %Partial pivoting for column p
   [Y,j]=max(abs(Aug(p:N,p)));
   %Interchange row p and j
   C=Aug(p,:);
   Aug(p,:)=Aug(j+p-1,:);
   Aug(j+p-1,:)=C;
if Aug(p,p)==0
   'A was singular.  No unique solution'
   break
end
%Elimination process for column p
   for k=p+1:N
      m=Aug(k,p)/Aug(p,p);
      Aug(k,p:N+1)=Aug(k,p:N+1)-m*Aug(p,p:N+1);
   end
end
%Back Substitution on [U|Y] using Program 3.1
X=backsub(Aug(1:N,1:N),Aug(1:N,N+1));
```

Exercises for Gaussian Elimination and Pivoting

In Exercises 1 through 4, show that $\boldsymbol{AX} = \boldsymbol{B}$ is equivalent to the upper-triangular system $\boldsymbol{UX} = \boldsymbol{Y}$ and find the solution.

1.
$$\begin{aligned} 2x_1 + 4x_2 - 6x_3 &= -4 \\ x_1 + 5x_2 + 3x_3 &= 10 \\ x_1 + 3x_2 + 2x_3 &= 5 \end{aligned} \qquad \begin{aligned} 2x_1 + 4x_2 - 6x_3 &= -4 \\ 3x_2 + 6x_3 &= 12 \\ 3x_3 &= 3 \end{aligned}$$

2.
$$\begin{aligned} x_1 + x_2 + 6x_3 &= 7 \\ -x_1 + 2x_2 + 9x_3 &= 2 \\ x_1 - 2x_2 + 3x_3 &= 10 \end{aligned} \qquad \begin{aligned} x_1 + x_2 + 6x_3 &= 7 \\ 3x_2 + 15x_3 &= 9 \\ 12x_3 &= 12 \end{aligned}$$

3.
$$\begin{aligned} 2x_1 - 2x_2 + 5x_3 &= 6 \\ 2x_1 + 3x_2 + x_3 &= 13 \\ -x_1 + 4x_2 - 4x_3 &= 3 \end{aligned} \qquad \begin{aligned} 2x_1 - 2x_2 + 5x_3 &= 6 \\ 5x_2 - 4x_3 &= 7 \\ 0.9x_3 &= 1.8 \end{aligned}$$

4.
$$\begin{aligned} -5x_1 + 2x_2 - x_3 &= -1 \\ x_1 + 0x_2 + 3x_3 &= 5 \\ 3x_1 + x_2 + 6x_3 &= 17 \end{aligned} \qquad \begin{aligned} -5x_1 + 2x_2 - x_3 &= -1 \\ 0.4x_2 + 2.8x_3 &= 4.8 \\ -10x_3 &= -10 \end{aligned}$$

5. Find the parabola $y = A + Bx + Cx^2$ that passes through $(1, 4)$, $(2, 7)$, and $(3, 14)$.

6. Find the parabola $y = A + Bx + Cx^2$ that passes through $(1, 6)$, $(2, 5)$, and $(3, 2)$.

7. Find the cubic $y = A + Bx + Cx^2 + Dx^3$ that passes through $(0, 0)$, $(1, 1)$, $(2, 2)$, and $(3, 2)$.

In Exercises 8 through 10, show that $\boldsymbol{AX} = \boldsymbol{B}$ is equivalent to the upper-triangular system $\boldsymbol{UX} = \boldsymbol{Y}$ and find the solution.

8.

$$\begin{array}{rcrcrcl}
4x_1 + 8x_2 + 4x_3 + 0x_4 &=& 8 & \qquad & 4x_1 + 8x_2 + 4x_3 + 0x_4 &=& 8 \\
x_1 + 5x_2 + 4x_3 - 3x_4 &=& -4 & & 3x_2 + 3x_3 - 3x_4 &=& -6 \\
x_1 + 4x_2 + 7x_3 + 2x_4 &=& 10 & & 4x_3 + 4x_4 &=& 12 \\
x_1 + 3x_2 + 0x_3 - 2x_4 &=& -4 & & x_4 &=& 2
\end{array}$$

9.

$$\begin{array}{rcrcrcl}
2x_1 + 4x_2 - 4x_3 + 0x_4 &=& 12 & \qquad & 2x_1 + 4x_2 - 4x_3 + 0x_4 &=& 12 \\
x_1 + 5x_2 - 5x_3 - 3x_4 &=& 18 & & 3x_2 - 3x_3 - 3x_4 &=& 12 \\
2x_1 + 3x_2 + x_3 + 3x_4 &=& 8 & & 4x_3 + 2x_4 &=& 0 \\
x_1 + 4x_2 - 2x_3 + 2x_4 &=& 8 & & 3x_4 &=& -6
\end{array}$$

10.

$$\begin{array}{rcrcrcl}
x_1 + 2x_2 + 0x_3 - x_4 &=& 9 & \qquad & x_1 + 2x_2 + 0x_3 - x_4 &=& 9 \\
2x_1 + 3x_2 - x_3 + 0x_4 &=& 9 & & -x_2 - x_3 + 2x_4 &=& -9 \\
0x_1 + 4x_2 + 2x_3 - 5x_4 &=& 26 & & -2x_3 + 3x_4 &=& -10 \\
5x_1 + 5x_2 + 2x_3 - 4x_4 &=& 32 & & 1.5x_4 &=& -3
\end{array}$$

11. Find the solution to the following linear system.

$$\begin{aligned}
x_1 + 2x_2 \qquad\qquad &= 7 \\
2x_1 + 3x_2 - x_3 \qquad &= 9 \\
4x_2 + 2x_3 + 3x_4 &= 10 \\
2x_3 - 4x_4 &= 12
\end{aligned}$$

12. Find the solution to the following linear system.

$$\begin{aligned}
x_1 + x_2 \qquad\qquad &= 5 \\
2x_1 - x_2 + 5x_3 \qquad &= -9 \\
3x_2 - 4x_3 + 2x_4 &= 19 \\
2x_3 + 6x_4 &= 2
\end{aligned}$$

13. The Rockmore Corp. is considering the purchase of a new computer and will choose either the DoGood 174 or the MightDo 11. They test both computers' ability to solve the linear system

$$\begin{aligned}
34x + 55y - 21 &= 0 \\
55x + 89y - 34 &= 0.
\end{aligned}$$

The DoGood 174 computer gives $x = -0.11$ and $y = 0.45$, and its check for accuracy can be obtained by defining $\boldsymbol{Y} = \boldsymbol{UX}$ and then solving two systems:

$$\text{(3)} \qquad \text{first solve } \boldsymbol{LY} = \boldsymbol{B} \text{ for } \boldsymbol{Y}; \qquad \text{then solve } \boldsymbol{UX} = \boldsymbol{Y} \text{ for } \boldsymbol{X}.$$

In equation form, we must first solve the lower-triangular system

$$\text{(4)} \qquad \begin{aligned} y_1 \qquad\qquad\qquad\qquad &= b_1 \\ m_{21}y_1 + \quad y_2 \qquad\qquad\quad &= b_2 \\ m_{31}y_1 + m_{32}y_2 + \quad y_3 \qquad &= b_3 \\ m_{41}y_1 + m_{42}y_2 + m_{43}y_3 + y_4 &= b_4 \end{aligned}$$

to obtain y_1, y_2, y_3, and y_4 and use them in solving the upper-triangular system

$$\text{(5)} \qquad \begin{aligned} u_{11}x_1 + u_{12}x_2 + u_{13}x_3 + u_{14}x_4 &= y_1 \\ u_{22}x_2 + u_{23}x_3 + u_{24}x_4 &= y_2 \\ u_{33}x_3 + u_{34}x_4 &= y_3 \\ u_{44}x_4 &= y_4. \end{aligned}$$

Example 3.20. Solve

$$\begin{aligned} x_1 + 2x_2 + 4x_3 + x_4 &= 21 \\ 2x_1 + 8x_2 + 6x_3 + 4x_4 &= 52 \\ 3x_1 + 10x_2 + 8x_3 + 8x_4 &= 79 \\ 4x_1 + 12x_2 + 10x_3 + 6x_4 &= 82. \end{aligned}$$

Use the triangular factorization method and the fact that

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 4 & 1 \\ 2 & 8 & 6 & 4 \\ 3 & 10 & 8 & 8 \\ 4 & 12 & 10 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 1 & 1 & 0 \\ 4 & 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 & 1 \\ 0 & 4 & -2 & 2 \\ 0 & 0 & -2 & 3 \\ 0 & 0 & 0 & -6 \end{bmatrix} = \boldsymbol{LU}.$$

Use the forward-substitution method to solve $\boldsymbol{LY} = \boldsymbol{B}$:

$$\text{(6)} \qquad \begin{aligned} y_1 \qquad\qquad\qquad &= 21 \\ 2y_1 + y_2 \qquad\qquad &= 52 \\ 3y_1 + y_2 + y_3 \qquad &= 79 \\ 4y_1 + y_2 + 2y_3 + y_4 &= 82. \end{aligned}$$

Compute the values $y_1 = 21$, $y_2 = 52 - 2(21) = 10$, $y_3 = 79 - 3(21) - 10 = 6$, and $y_4 = 82 - 4(21) - 10 - 2(6) = -24$, or $\boldsymbol{Y} = \begin{bmatrix} 21 & 10 & 6 & -24 \end{bmatrix}'$. Next write the system $\boldsymbol{UX} = \boldsymbol{Y}$:

$$\text{(7)} \qquad \begin{aligned} x_1 + 2x_2 + 4x_3 + x_4 &= 21 \\ 4x_2 - 2x_3 + 2x_4 &= 10 \\ -2x_3 + 3x_4 &= 6 \\ -6x_4 &= -24. \end{aligned}$$

Now use back substitution and compute the solution $x_4 = -24/(-6) = 4, x_3 = (6 - 3(4))/(-2) = 3, x_2 = (10 - 2(4) + 2(3))/4 = 2$, and $x_1 = 21 - 4 - 4(3) - 2(2) = 1$, or $\boldsymbol{X} = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}'$. ■

Triangular Factorization

We now discuss how to obtain the triangular factorization. If row interchanges are not necessary when using Gaussian elimination, the multipliers m_{ij} are the subdiagonal entries in $\boldsymbol{L}$.

Example 3.21. Use Gaussian elimination to construct the triangular factorization of the matrix

$$\boldsymbol{A} = \begin{bmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{bmatrix}.$$

The matrix $\boldsymbol{L}$ will be constructed from an identity matrix placed at the left. For each row operation used to construct the upper-triangular matrix, the multipliers m_{ij} will be put in their proper places at the left. Start with

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{bmatrix}.$$

Row 1 is used to eliminate the elements of $\boldsymbol{A}$ in column 1 below a_{11}. The multiples $m_{21} = -0.5$ and $m_{31} = 0.25$ of row 1 are subtracted from rows 2 and 3, respectively. These multipliers are put in the matrix at the left and the result is

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 1.25 & 6.25 \end{bmatrix}.$$

Row 2 is used to eliminate the elements in column 2 below the diagonal of the second factor of $\boldsymbol{A}$ in the above product. The multiple $m_{32} = -0.5$ of the second row is subtracted from row 3, and the multiplier is entered in the matrix at the left and we have the desired triangular factorization of $\boldsymbol{A}$.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & -0.5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 3 & -1 \\ 0 & -2.5 & 4.5 \\ 0 & 0 & 8.5 \end{bmatrix}. \tag{8}$$ ■

Theorem 3.10 (Direct Factorization $\boldsymbol{A} = \boldsymbol{LU}$: No Row Interchanges). Suppose that Gaussian elimination, without row interchanges, can be performed successfully to solve the general linear system $\boldsymbol{AX} = \boldsymbol{B}$. Then the matrix $\boldsymbol{A}$ can be factored as the product of a lower-triangular matrix $\boldsymbol{L}$ and an upper-triangular matrix $\boldsymbol{U}$:

$$\boldsymbol{A} = \boldsymbol{LU}.$$

MATLAB

The MATLAB command `[L,U,P]=lu(A)` creates the lower-triangular matrix $\boldsymbol{L}$, the upper-triangular matrix $\boldsymbol{U}$ (from the triangular factorization of $\boldsymbol{A}$), and the permutation matrix $\boldsymbol{P}$ from Theorem 3.14.

Example 3.25. Use the MATLAB command `[L,U,P]=lu(A)` on the matrix $\boldsymbol{A}$ in Example 3.22. Verify that $\boldsymbol{A} = \boldsymbol{P}^{-1}\boldsymbol{L}\boldsymbol{U}$ (equivalent to showing that $\boldsymbol{P}\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$).

```
>>A=[1 2 6 ;4 8 -1;-2 3 -5];
>>[L,U,P]=lu(A)

L=
    1.0000  0       0
    -0.5000 1.0000  0
    0.2500  0       1.0000

U=
    4.0000 8.0000 -1.0000
    0      7.0000 4.5000
    0      0      6.2500

P=
   0 1 0
   0 0 1
   1 0 0
>>inv(P)*L*U
   1  2 6
   4  8 -1
   -2 3 5
```

■

As indicated previously, the triangular factorization method is often chosen over the elimination method. In addition, it is used in the `inv(A)` and `det(A)` commands in MATLAB. For example, from the study of linear algebra we know that the determinant of a nonsingular matrix $\boldsymbol{A}$ equals $(-1)^q \det \boldsymbol{U}$, where $\boldsymbol{U}$ is the upper-triangular matrix from the triangular factorization of $\boldsymbol{A}$ and q is the number of row interchanges required to obtain $\boldsymbol{P}$ from the identity matrix $\boldsymbol{I}$. Since $\boldsymbol{U}$ is an upper-triangular matrix, we know that the determinant of $\boldsymbol{U}$ is just the product of the elements on its main diagonal (Theorem 3.6). The reader should verify in Example 3.25 that

$$\det(\boldsymbol{A}) = 175 = (-1)^2(175) = (-1)^2 \det(\boldsymbol{U}).$$

The following program implements the process described in the proof of Theorem 3.10. It is an extension of Program 3.2 and uses partial pivoting. The interchanging of rows due to partial pivoting is recorded in the matrix $\boldsymbol{R}$. The matrix $\boldsymbol{R}$ is then used in the forward substitution step to find the matrix $\boldsymbol{Y}$.

```
for k=N-1:-1:1
   X(k)=(Y(k)-A(k,k+1:N)*X(k+1:N))/A(k,k);
end
```

Exercises for Triangular Factorization

1. Solve $\boldsymbol{LY} = \boldsymbol{B}$, $\boldsymbol{UX} = \boldsymbol{Y}$, and verify that $\boldsymbol{B} = \boldsymbol{AX}$ for **(a)** $\boldsymbol{B} = \begin{bmatrix}-4 & 10 & 5\end{bmatrix}'$ and **(b)** $\boldsymbol{B} = \begin{bmatrix}20 & 49 & 32\end{bmatrix}'$, where $\boldsymbol{A} = \boldsymbol{LU}$ is

$$\begin{bmatrix}2 & 4 & -6\\ 1 & 5 & 3\\ 1 & 3 & 2\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 1/2 & 1 & 0\\ 1/2 & 1/3 & 1\end{bmatrix}\begin{bmatrix}2 & 4 & -6\\ 0 & 3 & 6\\ 0 & 0 & 3\end{bmatrix}.$$

2. Solve $\boldsymbol{LY} = \boldsymbol{B}$, $\boldsymbol{UX} = \boldsymbol{Y}$, and verify that $\boldsymbol{B} = \boldsymbol{AX}$ for **(a)** $\boldsymbol{B} = \begin{bmatrix}7 & 2 & 10\end{bmatrix}'$ and **(b)** $\boldsymbol{B} = \begin{bmatrix}23 & 35 & 7\end{bmatrix}'$, where $\boldsymbol{A} = \boldsymbol{LU}$ is

$$\begin{bmatrix}1 & 1 & 6\\ -1 & 2 & 9\\ 1 & -2 & 3\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ -1 & 1 & 0\\ 1 & -1 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & 6\\ 0 & 3 & 15\\ 0 & 0 & 12\end{bmatrix}.$$

3. Find the triangular factorization $\boldsymbol{A} = \boldsymbol{LU}$ for the matrices

(a) $\begin{bmatrix}-5 & 2 & -1\\ 1 & 0 & 3\\ 3 & 1 & 6\end{bmatrix}$ **(b)** $\begin{bmatrix}1 & 0 & 3\\ 3 & 1 & 6\\ -5 & 2 & -1\end{bmatrix}$

4. Find the triangular factorization $\boldsymbol{A} = \boldsymbol{LU}$ for the matrices

(a) $\begin{bmatrix}4 & 2 & 1\\ 2 & 5 & -2\\ 1 & -2 & 7\end{bmatrix}$ **(b)** $\begin{bmatrix}1 & -2 & 7\\ 4 & 2 & 1\\ 2 & 5 & -2\end{bmatrix}$

5. Solve $\boldsymbol{LY} = \boldsymbol{B}$, $\boldsymbol{UX} = \boldsymbol{Y}$, and verify that $\boldsymbol{B} = \boldsymbol{AX}$ for **(a)** $\boldsymbol{B} = \begin{bmatrix}8 & -4 & 10 & -4\end{bmatrix}'$ and **(b)** $\boldsymbol{B} = \begin{bmatrix}28 & 13 & 23 & 4\end{bmatrix}'$, where $\boldsymbol{A} = \boldsymbol{LU}$ is

$$\begin{bmatrix}4 & 8 & 4 & 0\\ 1 & 5 & 4 & -3\\ 1 & 4 & 7 & 2\\ 1 & 3 & 0 & -2\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ \frac{1}{4} & 1 & 0 & 0\\ \frac{1}{4} & \frac{2}{3} & 1 & 0\\ \frac{1}{4} & \frac{1}{3} & -\frac{1}{2} & 1\end{bmatrix}\begin{bmatrix}4 & 8 & 4 & 0\\ 0 & 3 & 3 & -3\\ 0 & 0 & 4 & 4\\ 0 & 0 & 0 & 1\end{bmatrix}.$$

6. Find the triangular factorization $\boldsymbol{A} = \boldsymbol{LU}$ for the matrix

$$\begin{bmatrix}1 & 1 & 0 & 4\\ 2 & -1 & 5 & 0\\ 5 & 2 & 1 & 2\\ -3 & 0 & 2 & 6\end{bmatrix}.$$

7. Establish the formula in (12).

Theorem 3.15 (Jacobi Iteration). Suppose that $\boldsymbol{A}$ is a strictly diagonally dominant matrix. Then $\boldsymbol{AX} = \boldsymbol{B}$ has a unique solution $\boldsymbol{X} = \boldsymbol{P}$. Iteration using formula (10) will produce a sequence of vectors $\{\boldsymbol{P}_k\}$ that will converge to $\boldsymbol{P}$ for any choice of the starting vector $\boldsymbol{P}_0$.

Proof. The proof can be found in advanced texts on numerical analysis. •

It can be proved that the Gauss-Seidel method will also converge when the matrix $\boldsymbol{A}$ is strictly diagonally dominant. In many cases the Gauss-Seidel method will converge faster than the Jacobi method; hence it is usually preferred (compare Examples 3.26 and 3.28). It is important to understand the slight modification of formula (10) that has been made to obtain formula (11). In some cases the Jacobi method will converge even though the Gauss-Seidel method will not.

Convergence

A measure of the closeness between vectors is needed so that we can determine if $\{\boldsymbol{P}_k\}$ is converging to $\boldsymbol{P}$. The Euclidean distance (see Section 3.1) between $\boldsymbol{P} = (x_1, x_2, \ldots, x_N)$ and $\boldsymbol{Q} = (y_1, y_2, \ldots, y_N)$ is

$$(12) \qquad \|\boldsymbol{P} - \boldsymbol{Q}\| = \left(\sum_{j=1}^{N}(x_j - y_j)^2\right)^{1/2}.$$

Its disadvantage is that it requires considerable computing effort. Hence we introduce a different norm, $\|\boldsymbol{X}\|_1$:

$$(13) \qquad \|\boldsymbol{X}\|_1 = \sum_{j=1}^{N} |x_j|.$$

The following result ensures that $\|\boldsymbol{X}\|_1$ has the mathematical structure of a metric and hence is suitable to use as a generalized "distance formula." From the study of linear algebra we know that on a finite-dimensional vector space all norms are equivalent; that is, if two vectors are close in the $\|*\|_1$ norm, then they are also close in the Euclidean norm $\|*\|$.

Theorem 3.16. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be N-dimensional vectors and c be a scalar. Then the function $\|\boldsymbol{X}\|_1$ has the following properties:

$$(14) \qquad \|\boldsymbol{X}\|_1 \geq 0,$$

$$(15) \qquad \|\boldsymbol{X}\|_1 = 0 \quad \text{if and only if} \quad \boldsymbol{X} = \boldsymbol{0},$$

$$(16) \qquad \|c\boldsymbol{X}\|_1 = |c|\,\|\boldsymbol{X}\|_1,$$

$$(17) \qquad \|\boldsymbol{X} + \boldsymbol{Y}\|_1 \leq \|\boldsymbol{X}\|_1 + \|\boldsymbol{Y}\|_1.$$

```
      X(j)=(B(j)-A(j,1:j-1)*X(1:j-1)'
          -A(j,j+1:N)*P(j+1:N))/A(j,j);
    end
  end
  err=abs(norm(X'-P));
  relerr=err/(norm(X)+eps);
  P=X';
    if(err<delta)|(relerr<delta)
      break
    end
end
X=X';
```

Exercises for Iterative Methods for Linear Systems

In Exercises 1 through 8:
(a) Start with $\boldsymbol{P}_0 = \mathbf{0}$ and use Jacobi iteration to find $\boldsymbol{P}_k$ for $k = 1, 2, 3$. Will Jacobi iteration converge to the solution?
(b) Start with $\boldsymbol{P}_0 = \mathbf{0}$ and use Gauss-Seidel iteration to find $\boldsymbol{P}_k$ for $k = 1, 2, 3$. Will Gauss-Seidel iteration converge to the solution?

1.
$$\begin{aligned} 4x - y &= 15 \\ x + 5y &= 9 \end{aligned}$$

2.
$$\begin{aligned} 8x - 3y &= 10 \\ -x + 4y &= 6 \end{aligned}$$

3.
$$\begin{aligned} -x + 3y &= 1 \\ 6x - 2y &= 2 \end{aligned}$$

4.
$$\begin{aligned} 2x + 3y &= 1 \\ 7x - 2y &= 1 \end{aligned}$$

5.
$$\begin{aligned} 5x - y + z &= 10 \\ 2x + 8y - z &= 11 \\ -x + y + 4z &= 3 \end{aligned}$$

6.
$$\begin{aligned} 2x + 8y - z &= 11 \\ 5x - y + z &= 10 \\ -x + y + 4z &= 3 \end{aligned}$$

7.
$$\begin{aligned} x - 5y - z &= -8 \\ 4x + y - z &= 13 \\ 2x - y - 6z &= -2 \end{aligned}$$

8.
$$\begin{aligned} 4x + y - z &= 13 \\ x - 5y - z &= -8 \\ 2x - y - 6z &= -2 \end{aligned}$$

9. Let $\boldsymbol{X} = (x_1, x_2, \ldots, x_N)$. Prove that the $\|*\|_1$ norm

$$\|\boldsymbol{X}\|_1 = \sum_{k=1}^{N} |x_k|$$

satisfies the three properties (14)–(16).

Example 4.1. Show why 15 terms are all that are needed to obtain the 13-digit approximation $e = 2.718281828459$ in Table 4.2.

Expand $f(x) = e^x$ in a Taylor polynomial of degree 15 using the fixed value $x_0 = 0$ and involving the powers $(x - 0)^k = x^k$. The derivatives required are $f'(x) = f''(x) = \cdots = f^{(16)} = e^x$. The first 15 derivatives are used to calculate the coefficients $a_k = e^0/k!$ and are used to write

$$(4) \qquad P_{15}(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^{15}}{15!}.$$

Setting $x = 1$ in (4) gives the partial sum $S_{15} = P_{15}(1)$. The remainder term is needed to show the accuracy of the approximation:

$$(5) \qquad E_{15}(x) = \frac{f^{(16)}(c)x^{16}}{16!}.$$

Since we chose $x_0 = 0$ and $x = 1$, the value c lies between them (i.e., $0 < c < 1$), which implies that $e^c < e^1$. Notice that the partial sums in Table 4.2 are bounded above by 3. Combining these two inequalities yields $e^c < 3$, which is used in the following calculation

$$|E_{15}(1)| = \frac{|f^{(16)}(c)|}{16!} \le \frac{e^c}{16!} < \frac{3}{16!} < 1.433844 \times 10^{-13}.$$

Therefore, all the digits in the approximation $e \approx 2.718281828459$ are correct, because the actual error (whatever it is) must be less than 2 in the thirteenth decimal place. ■

Instead of giving a rigorous proof of Theorem 4.1, we shall discuss some of the features of the approximation; the reader can look in any standard reference text on calculus for more details. For illustration, we again use the function $f(x) = e^x$ and the value $x_0 = 0$. From elementary calculus we know that the slope of the curve $y = e^x$ at the point (x, e^x) is $f'(x) = e^x$. Hence the slope at the point $(0, 1)$ is $f'(0) = 1$. Therefore, the tangent line to the curve at the point $(0, 1)$ is $y = 1 + x$. This is the same formula that would be obtained if we used $N = 1$ in Theorem 4.1; that is, $P_1(x) = f(0) + f'(0)x/1! = 1 + x$. Therefore, $P_1(x)$ is the equation of the tangent line to the curve. The graphs are shown in Figure 4.3.

Observe that the approximation $e^x \approx 1 + x$ is good near the center $x_0 = 0$ and that the distance between the curves grows as x moves away from 0. Notice that the slopes of the curves agree at $(0, 1)$. In calculus we learned that the second derivative indicates whether a curve is concave up or down. The study of curvature[1] shows that if two curves $y = f(x)$ and $y = g(x)$ have the property that $f(x_0) = g(x_0)$, $f'(x_0) = g'(x_0)$, and $f''(x_0) = g''(x_0)$ then they have the same curvature at x_0. This property would be desirable for a polynomial function that approximates $f(x)$. Corollary 4.1 shows that the Taylor polynomial has this property for $N \ge 2$.

[1] The curvature K of a graph $y = f(x)$ at (x_0, y_0) is defined by $K = |f''(x_0)|/(1+[f'(x_0)]^2)^{3/2}$.

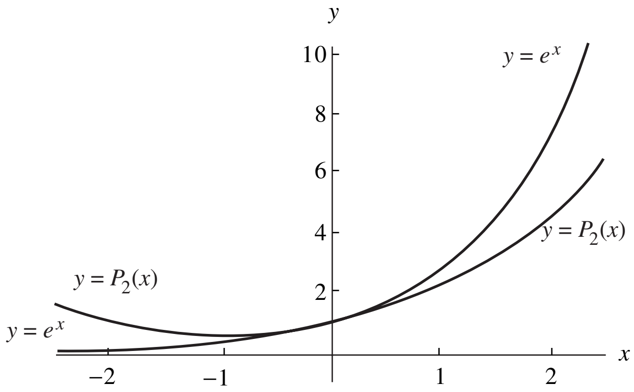


Figure 4.4 The graphs of $y = e^x$ and $y = P_2(x) = 1 + x + x^2/2$.

Table 4.3 Values for the Error Bound $|\text{error}| < e^R R^{N+1}/(N+1)!$ Using the Approximation $e^x \approx P_N(x)$ for $|x| \le R$

	$R = 2.0$, $\lvert x\rvert \le 2.0$	$R = 1.5$, $\lvert x\rvert \le 1.5$	$R = 1.0$, $\lvert x\rvert \le 1.0$	$R = 0.5$, $\lvert x\rvert \le 0.5$
$e^x \approx P_5(x)$	0.65680499	0.07090172	0.00377539	0.00003578
$e^x \approx P_6(x)$	0.18765857	0.01519323	0.00053934	0.00000256
$e^x \approx P_7(x)$	0.04691464	0.00284873	0.00006742	0.00000016
$e^x \approx P_8(x)$	0.01042548	0.00047479	0.00000749	0.00000001

the absolute value of the error satisfies the relation

$$|\text{error}| = |E_N(x)| \le \frac{MR^{N+1}}{(N+1)!}, \tag{8}$$

where $M \le \max\{|f^{(N+1)}(z)| : x_0 - R \le z \le x_0 + R\}$. If the derivatives are uniformly bounded, the error bound in (8) is proportional to $R^{N+1}/(N+1)!$ and decreases for fixed R, when N gets large or, for fixed N, when R goes to 0. Table 4.3 shows how the choices of these two parameters affect the accuracy of the approximation $e^x \approx P_N(x)$ over the interval $|x| \le R$. The error is smallest when N is largest and R smallest. Graphs for P_2, P_3, and P_4 are given in Figure 4.5.

Example 4.2. Establish the error bounds for the approximation $e^x \approx P_8(x)$ on each of the intervals $|x| \le 1.0$ and $|x| \le 0.5$.

If $|x| \le 1.0$, then letting $R = 1.0$ and $|f^{(9)}(c)| = |e^c| \le e^{1.0} = M$ in (8) implies that

$$|\text{error}| = |E_8(x)| \le \frac{e^{1.0}(1.0)^9}{9!} \approx 0.00000749.$$

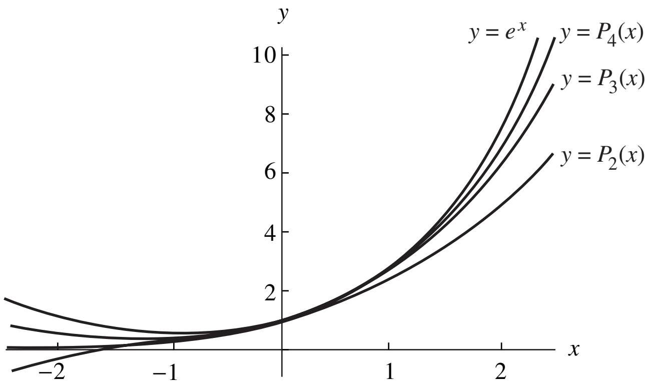


Figure 4.5 The graphs of $y = e^x$, $y = P_2(x)$, $y = P_3(x)$, and $y = P_4(x)$.

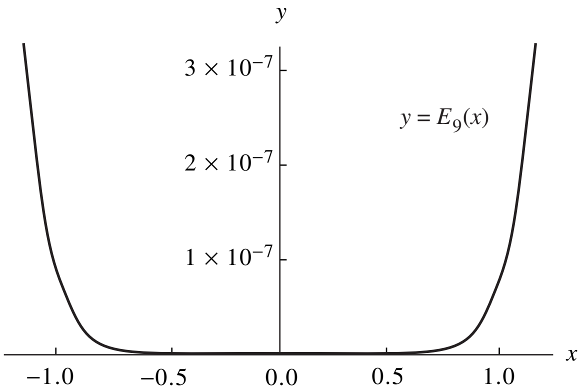


Figure 4.6 The graph of the error $y = E_9(x) = e^x - P_9(x)$.

If $|x| \le 0.5$, then letting $R = 0.5$ and $|f^{(9)}(c)| = |e^c| \le e^{0.5} = M$ in (8) implies that

$$|\text{error}| = |E_8(x)| \le \frac{e^{0.5}(0.5)^9}{9!} \approx 0.00000001. \qquad \blacksquare$$

Example 4.3. If $f(x) = e^x$, show that $N = 9$ is the smallest integer, so that the $|\text{error}| = |E_N(x)| \le 0.0000005$ for x in $[-1, 1]$. Hence $P_9(x)$ can be used to compute approximate values of e^x that will be accurate in the sixth decimal place.

We need to find the smallest integer N so that

$$|\text{error}| = |E_N(x)| \le \frac{e^c(1)^{N+1}}{(N+1)!} < 0.0000005.$$

In Example 4.2 we saw that $N = 8$ was too small, so we try $N = 9$ and discover that $|E_N(x)| \le e^1(1)^{9+1}/(9+1)! \le 0.000000749$. This value is slightly larger than desired; hence we would be likely to choose $N = 10$. But we used $e^c \le e^1$ as a crude estimate in finding the error bound. Hence 0.000000749 is a little larger than the actual error. Figure 4.6 shows a graph of $E_9(x) = e^x - P_9(x)$. Notice that the maximum vertical range is about 3×10^{-7} and occurs at the right endpoint $(1, E_9(1))$. Indeed, the maximum error on the interval is $E_9(1) = 2.718281828 - 2.718281526 \approx 3.024 \times 10^{-7}$. Therefore, $N = 9$ is justified. ■

Methods for Evaluating a Polynomial

There are several mathematically equivalent ways to evaluate a polynomial. Consider, for example, the function

$$f(x) = (x-1)^8. \tag{9}$$

The evaluation of f will require the use of an exponential function. Or the binomial formula can be used to expand $f(x)$ in powers of x:

$$\begin{aligned} f(x) &= \sum_{k=0}^{8} \binom{8}{k} x^{8-k}(-1)^k \\ &= x^8 - 8x^7 + 28x^6 - 56x^5 + 70x^4 - 56x^3 + 28x^2 - 8x + 1. \end{aligned} \tag{10}$$

Horner's method (see Section 1.1), which is also called ***nested multiplication***, can now be used to evaluate the polynomial in (10). When applied to formula (10), nested multiplication permits us to write

$$f(x) = (((((((x-8)x+28)x-56)x+70)x-56)x+28)x-8)x+1. \tag{11}$$

To evaluate $f(x)$ now requires seven multiplications and eight additions or subtractions. The necessity of using an exponential function to evaluate the polynomial has now been eliminated.

We end this section with the theorem that relates the Taylor series in Table 4.1 and the Taylor polynomials of Theorem 4.1.

Theorem 4.2 (Taylor Series). Assume that $f(x)$ is analytic on an interval (a, b) containing x_0. Suppose that the Taylor polynomials (2) tend to a limit

$$S(x) = \lim_{N\to\infty} P_N(x) = \lim_{N\to\infty} \sum_{k=0}^{N} \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k; \tag{12}$$

then $f(x)$ has the Taylor series expansion

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k. \tag{13}$$

Proof. This follows directly from the definition of convergence of series in Section 1.1. The limit condition is often stated by saying that the error term must go to zero as N goes to infinity. Therefore, a necessary and sufficient condition for (13) to hold is that

$$\lim_{N\to\infty} E_N(x) = \lim_{N\to\infty} \frac{f^{(N+1)}(c)(x-x_0)^{N+1}}{(N+1)!} = 0, \tag{14}$$

where c depends on N and x. •

Exercises for Taylor Series and Calculation of Functions

1. Let $f(x) = \sin(x)$ and apply Theorem 4.1.

(a) Use $x_0 = 0$ and find $P_5(x)$, $P_7(x)$, and $P_9(x)$.

(b) Show that if $|x| \le 1$, then the approximation

$$\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}$$

has the error bound $|E_9(x)| < 1/10! \le 2.75574 \times 10^{-7}$.

(c) Use $x_0 = \pi/4$ and find $P_5(x)$, which involves powers of $(x - \pi/4)$.

2. Let $f(x) = \cos(x)$ and apply Theorem 4.1.

(a) Use $x_0 = 0$ and find $P_4(x)$, $P_6(x)$, and $P_8(x)$.

(b) Show that if $|x| \le 1$, then the approximation

$$\cos(x) \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!}$$

has the error bound $|E_8(x)| < 1/9! \le 2.75574 \times 10^{-6}$.

(c) Use $x_0 = \pi/4$ and find $P_4(x)$, which involves powers of $(x - \pi/4)$.

3. Does $f(x) = x^{1/2}$ have a Taylor series expansion about $x_0 = 0$? Justify your answer. Does the function $f(x) = x^{1/2}$ have a Taylor series expansion about $x_0 = 1$? Justify your answer.

4. **(a)** Find a Taylor polynomial of degree $N = 5$ for $f(x) = 1/(1 + x)$ expanded about $x_0 = 0$.

(b) Find the error term $E_5(x)$ for the polynomial in part (a).

5. Find the Taylor polynomial of degree $N = 3$ for $f(x) = e^{-x^2/2}$ expanded about $x_0 = 0$.

6. Find the Taylor polynomial of degree $N = 3$, $P_3(x)$, for $f(x) = x^3 - 2x^2 + 2x$ expanded about $x_0 = 1$. Show that $f(x) = P_3(x)$.

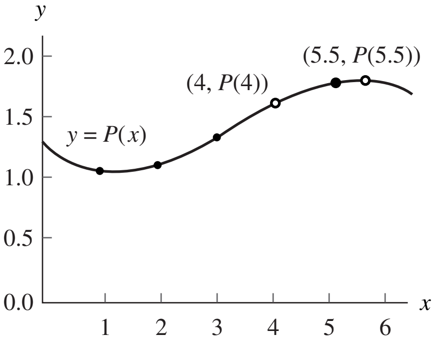


Figure 4.7 (a) The approximating polynomial $P(x)$ can be used for interpolation at the point (4, $P(4)$) and extrapolation at the point (5.5, $P(5.5)$).

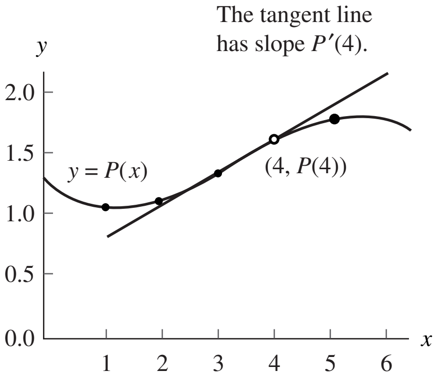


Figure 4.7 (b) The approximating polynomial $P(x)$ is differentiated and $P'(x)$ is used to find the slope at the interpolation point (4, $P(4)$).

Let us briefly mention how to evaluate the polynomial $P(x)$:

(1) $$P(x) = a_N x^N + a_{N-1} x^{N-1} + \cdots + a_2 x^2 + a_1 x + a_0.$$

Horner's method of synthetic division is an efficient way to evaluate $P(x)$. The derivative $P'(x)$ is

(2) $$P'(x) = N a_N x^{N-1} + (N-1) a_{N-1} x^{N-2} + \cdots + 2 a_2 x + a_1$$

and the indefinite integral $I(x) = \int P(x)\,dx$, which satisfies $I'(x) = P(x)$, is

(3) $$I(x) = \frac{a_N x^{N+1}}{N+1} + \frac{a_{N-1} x^N}{N} + \cdots + \frac{a_2 x^3}{3} + \frac{a_1 x^2}{2} + a_0 x + C,$$

where C is the constant of integration. Algorithm 4.1 (end of Section 4.2) shows how to adapt Horner's method to $P'(x)$ and $I(x)$.

Example 4.4. The polynomial $P(x) = -0.02x^3 + 0.2x^2 - 0.4x + 1.28$ passes through the four points (1, 1.06), (2, 1.12), (3, 1.34), and (5, 1.78). Find **(a)** $P(4)$, **(b)** $P'(4)$, **(c)** $\int_1^4 P(x)dx$, and **(d)** $P(5.5)$. Finally, **(e)** show how to find the coefficients of $P(x)$.

Use Algorithm 4.1(i)–(iii) (this is equivalent to the process in Table 1.2) with $x = 4$.

(a)
$$\begin{aligned}
b_3 &= a_3 = -0.02\\
b_2 &= a_2 + b_3 x = 0.2 + (-0.02)(4) = 0.12\\
b_1 &= a_1 + b_2 x = -0.4 + (0.12)(4) = 0.08\\
b_0 &= a_0 + b_1 x = 1.28 + (0.08)(4) = 1.60.
\end{aligned}$$

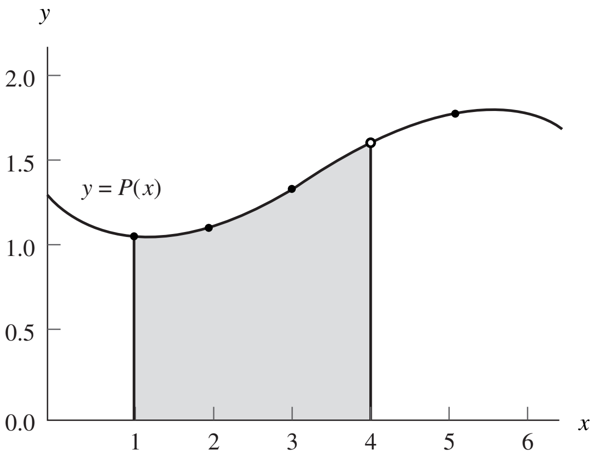


Figure 4.8 The approximating polynomial $P(x)$ is integrated and its antiderivative is used to find the area under the curve for $1 \leq x \leq 4$.

The interpolated value is $P(4) = 1.60$ (see Figure 4.7(a)).

(**b**)
$$\begin{aligned} d_2 &= 3a_3 = -0.06 \\ d_1 &= 2a_2 + d_2x = 0.4 + (-0.06)(4) = 0.16 \\ d_0 &= a_1 + d_1x = -0.4 + (0.16)(4) = 0.24. \end{aligned}$$

The numerical derivative is $P'(4) = 0.24$ (see Figure 4.7(b)).

(**c**)
$$\begin{aligned} i_4 &= \frac{a_3}{4} = -0.005 \\ i_3 &= \frac{a_2}{3} + i_4x = 0.06666667 + (-0.005)(4) = 0.04666667 \\ i_2 &= \frac{a_1}{2} + i_3x = -0.2 + (0.04666667)(4) = -0.01333333 \\ i_1 &= a_0 + i_2x = 1.28 + (-0.01333333)(4) = 1.22666667 \\ i_0 &= 0 + i_1x = 0 + (1.22666667)(4) = 4.90666667. \end{aligned}$$

Hence $I(4) = 4.90666667$. Similarly, $I(1) = 1.14166667$. Therefore, $\int_1^4 P(x)\,dx = I(4) - I(1) = 3.765$ (see Figure 4.8).
(d) Use Algorithm 4.1(i) with $x = 5.5$.

$$\begin{aligned} b_3 &= a_3 = -0.02 \\ b_2 &= a_2 + b_3x = 0.2 + (-0.02)(5.5) = 0.09 \\ b_1 &= a_1 + b_2x = -0.4 + (0.09)(5.5) = 0.095 \\ b_0 &= a_0 + b_1x = 1.28 + (0.095)(5.5) = 1.8025. \end{aligned}$$

The extrapolated value is $P(5.5) = 1.8025$ (see Figure 4.7(a)).

Table 4.4 Values of the Taylor Polynomial $T(x)$ of Degree 5, the Function $\ln(1+x)$, and the Error $\ln(1+x) - T(x)$ on [0, 1]

x	Taylor polynomial, $T(x)$	Function, $\ln(1+x)$	Error, $\ln(1+x) - T(x)$
0.0	0.00000000	0.00000000	0.00000000
0.2	0.18233067	0.18232156	−0.00000911
0.4	0.33698133	0.33647224	−0.00050909
0.6	0.47515200	0.47000363	−0.00514837
0.8	0.61380267	0.58778666	−0.02601601
1.0	0.78333333	0.69314718	−0.09018615

(e) The methods of Chapter 3 can be used to find the coefficients. Assume that $P(x) = A + Bx + Cx^2 + Dx^3$; then at each value $x = 1, 2, 3$, and 5 we get a linear equation involving A, B, C, and D.

$$\begin{aligned}
&At\, x = 1 : A + 1B + \ \ 1C + \ \ \ 1D = 1.06\\
&At\, x = 2 : A + 2B + \ \ 4C + \ \ \ 8D = 1.12\\
&At\, x = 3 : A + 3B + \ \ 9C + \ 27D = 1.34\\
&At\, x = 5 : A + 5B + 25C + 125D = 1.78
\end{aligned} \tag{4}$$

The solution to (4) is $A = 1.28$, $B = -0.4$, $C = 0.2$, and $D = -0.2$. ■

This method for finding the coefficients is mathematically sound, but sometimes the matrix is difficult to solve accurately. In this chapter we design algorithms specifically for polynomials.

Let us return to the topic of using a polynomial to calculate approximations to a known function. In Section 4.1 we saw that the fifth-degree Taylor polynomial for $f(x) = \ln(1+x)$ is

$$T(x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \frac{x^5}{5}. \tag{5}$$

If $T(x)$ is used to approximate $\ln(1+x)$ on the interval [0, 1], then the error is 0 at $x = 0$ and is largest when $x = 1$ (see Table 4.4). Indeed, the error between $T(1)$ and the correct value ln(2) is 13%. We seek a polynomial of degree 5 that will approximate $\ln(1+x)$ better over the interval [0, 1]. The polynomial $P(x)$ in Example 4.5 is an interpolating polynomial and will approximate $\ln(1+x)$ with an error no bigger than 0.00002385 over the interval [0, 1].

Example 4.5. Consider the function $f(x) = \ln(1+x)$ and the polynomial

$$\begin{aligned}
P(x) = 0.02957206x^5 - 0.12895295x^4 + 0.28249626x^3\\
- 0.48907554x^2 + 0.99910735x
\end{aligned}$$

Exercises for Introduction to Interpolation

1. Consider $P(x) = -0.02x^3 + 0.1x^2 - 0.2x + 1.66$, which passes through the four points $(1, 1.54)$, $(2, 1.5)$, $(3, 1.42)$, and $(5, 0.66)$.

(a) Find $P(4)$.

(b) Find $P'(4)$.

(c) Find the definite integral of $P(x)$ taken over $[1, 4]$.

(d) Find the extrapolated value $P(5.5)$.

(e) Show how to find the coefficients of $P(x)$.

2. Consider $P(x) = -0.04x^3 + 0.14x^2 - 0.16x + 2.08$, which passes through the four points $(0, 2.08)$, $(1, 2.02)$, $(2, 2.00)$, and $(4, 1.12)$.

(a) Find $P(3)$.

(b) Find $P'(3)$.

(c) Find the definite integral of $P(x)$ taken over $[0, 3]$.

(d) Find the extrapolated value $P(4.5)$.

(e) Show how to find the coefficients of $P(x)$.

3. Consider $P(x) = -0.0292166667x^3 + 0.275x^2 - 0.570833333x + 1.375$, which passes through the four points $(1, 1.05)$, $(2, 1.10)$, $(3, 1.35)$, and $(5, 1.75)$.

(a) Show that the ordinates 1.05, 1.10, 1.35, and 1.75 differ from those of Example 4.4 by less than 1.8%, yet the coefficients of x^3 and x differ by more than 42%.

(b) Find $P(4)$ and compare with Example 4.4.

(c) Find $P'(4)$ and compare with Example 4.4.

(d) Find the definite integral of $P(x)$ taken over $[1, 4]$ and compare with Example 4.4.

(e) Find the extrapolated value $P(5.5)$ and compare with Example 4.4.

Remark. Part (a) shows that the computation of the coefficients of an interpolating polynomial is an ill-conditioned problem.

Algorithms and Programs

1. Write a program in MATLAB that will implement Algorithm 4.1. The program should accept the coefficients of the polynomial $P(x) = a_N x^N + a_{N-1}x^{N-1} + \cdots + a_2x^2 + a_1x + a_0$ as an $1 \times N$ matrix: $P = \begin{bmatrix} a_N & a_{N-1} & \cdots & a_2 & a_1 & a_0 \end{bmatrix}$.

2. For each of the given functions, the fifth-degree polynomial $P(x)$ passes through the six points $(0, f(0))$, $(0.2, f(0.2))$, $(0.4, f(0.4))$, $(0.6, f(0.6))$, $(0.8, f(0.8))$, $(1, f(1))$. The six coefficients of $P(x)$ are $a_0, a_1, \ldots, a_5$, where

$$P(x) = a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0.$$

When formula (1) is expanded, the result is a polynomial of degree ≤ 1. Evaluation of $P(x)$ at x_0 and x_1 produces y_0 and y_1, respectively:

$$(2) \qquad \begin{aligned} P(x_0) &= y_0 + (y_1 - y_0)(0) = y_0, \\ P(x_1) &= y_0 + (y_1 - y_0)(1) = y_1. \end{aligned}$$

The French mathematician Joseph Louis Lagrange used a slightly different method to find this polynomial. He noticed that it could be written as

$$(3) \qquad y = P_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}.$$

Each term on the right side of (3) involves a linear factor; hence the sum is a polynomial of degree ≤ 1. The quotients in (3) are denoted by

$$(4) \qquad L_{1,0}(x) = \frac{x - x_1}{x_0 - x_1} \quad \text{and} \quad L_{1,1}(x) = \frac{x - x_0}{x_1 - x_0}.$$

Computation reveals that $L_{1,0}(x_0) = 1$, $L_{1,0}(x_1) = 0$, $L_{1,1}(x_0) = 0$, and $L_{1,1}(x_1) = 1$ so that the polynomial $P_1(x)$ in (3) also passes through the two given points:

$$(5) \qquad P_1(x_0) = y_0 + y_1(0) = y_0 \quad \text{and} \quad P_1(x_1) = y_0(0) + y_1 = y_1.$$

The terms $L_{1,0}(x)$ and $L_{1,1}(x)$ in (4) are called ***Lagrange coefficient polynomials*** based on the nodes x_0 and x_1. Using this notation, (3) can be written in summation form

$$(6) \qquad P_1(x) = \sum_{k=0}^{1} y_k L_{1,k}(x).$$

Suppose that the ordinates y_k are computed with the formula $y_k = f(x_k)$. If $P_1(x)$ is used to approximate $f(x)$ over the interval $[x_0, x_1]$, we call the process ***interpolation***. If $x < x_0$ (or $x_1 < x$), then using $P_1(x)$ is called ***extrapolation***. The next example illustrates these concepts.

Example 4.6. Consider the graph $y = f(x) = \cos(x)$ over $[0.0, 1.2]$.

(a) Use the nodes $x_0 = 0.0$ and $x_1 = 1.2$ to construct a linear interpolation polynomial $P_1(x)$.

(b) Use the nodes $x_0 = 0.2$ and $x_1 = 1.0$ to construct a linear approximating polynomial $Q_1(x)$.

(a) Using (3) with the abscissas $x_0 = 0.0$ and $x_1 = 1.2$ and the ordinates $y_0 = \cos(0.0) = 1.000000$ and $y_1 = \cos(1.2) = 0.362358$ produces

$$\begin{aligned} P_1(x) &= 1.000000 \frac{x - 1.2}{0.0 - 1.2} + 0.362358 \frac{x - 0.0}{1.2 - 0.0} \\ &= -0.833333(x - 1.2) + 0.301965(x - 0.0). \end{aligned}$$

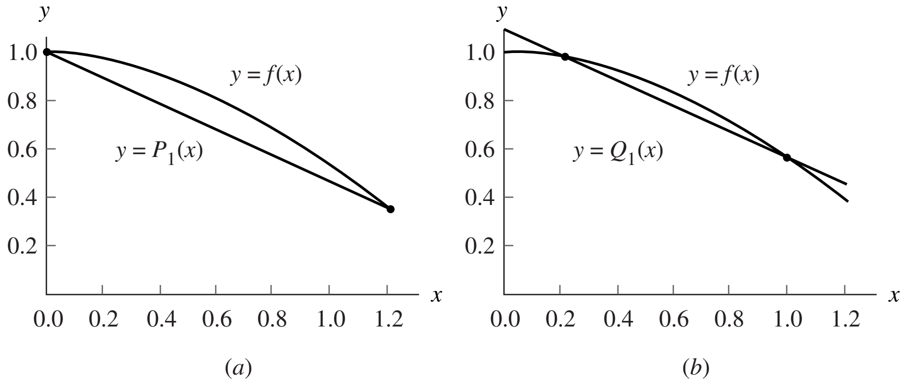


Figure 4.11 (a) The linear approximation $y = P_1(x)$ where the nodes $x_0 = 0.0$ and $x_1 = 1.2$ are the endpoints of the interval $[a, b]$. (b) The linear approximation $y = Q_1(x)$ where the nodes $x_0 = 0.2$ and $x_1 = 1.0$ lie inside the interval $[a, b]$.

(b) When the nodes $x_0 = 0.2$ and $x_1 = 1.0$ with $y_0 = \cos(0.2) = 0.980067$ and $y_1 = \cos(1.0) = 0.540302$ are used, the result is

$$\begin{aligned} Q_1(x) &= 0.980067\frac{x - 1.0}{0.2 - 1.0} + 0.540302\frac{x - 0.2}{1.0 - 0.2} \\ &= -1.225083(x - 1.0) + 0.675378(x - 0.2). \end{aligned}$$

Figure 4.11(a) and (b) show the graph of $y = \cos(x)$ and compare it with $y = P_1(x)$ and $y = Q_1(x)$, respectively. Numerical computations are given in Table 4.6 and reveal that $Q_1(x)$ has less error at the points x_k that satisfy $0.1 \le x_k \le 1.1$. The largest tabulated error, $f(0.6) - P_1(0.6) = 0.144157$, is reduced to $f(0.6) - Q_1(0.6) = 0.065151$ by using $Q_1(x)$. ■

The generalization of (6) is the construction of a polynomial $P_N(x)$ of degree at most N that passes through the $N + 1$ points $(x_0, y_0), (x_1, y_1), \ldots, (x_N, y_N)$ and has the form

$$P_N(x) = \sum_{k=0}^{N} y_k L_{N,k}(x), \tag{7}$$

where $L_{N,k}$ is the Lagrange coefficient polynomial based on these nodes:

$$L_{N,k}(x) = \frac{(x - x_0)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_N)}{(x_k - x_0)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_N)}. \tag{8}$$

It is understood that the terms $(x - x_k)$ and $(x_k - x_k)$ do not appear on the right side of

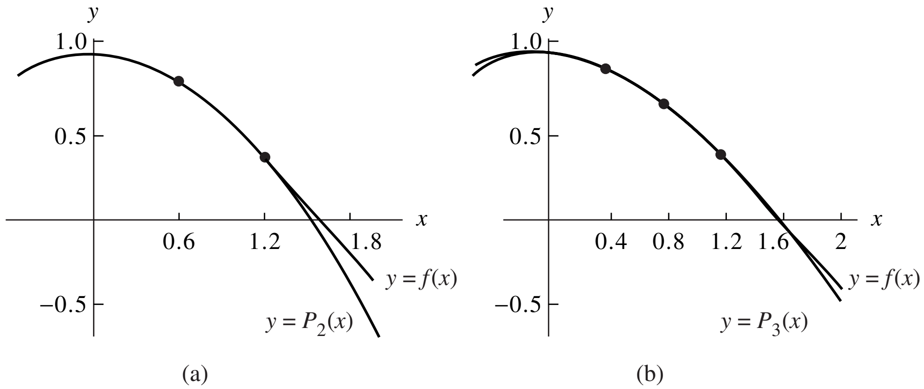


Figure 4.12 (a) The quadratic approximation polynomial $y = P_2(x)$ based on the nodes $x_0 = 0.0$, $x_1 = 0.6$, and $x_2 = 1.2$. (b) The cubic approximation polynomial $y = P_3(x)$ based on the nodes $x_0 = 0.0$, $x_1 = 0.4$, $x_2 = 0.8$, and $x_3 = 1.2$.

$P_N(x) - Q_N(x)$. Observe that the polynomial $T(x)$ has degree $\le N$ and that $T(x_j) = P_N(x_j) - Q_N(x_j) = y_j - y_j = 0$, for $j = 0, 1, \ldots, N$. Therefore, $T(x) \equiv 0$ and it follows that $Q_N(x) = P_N(x)$.

When (7) is expanded, the result is similar to (3). The Lagrange quadratic interpolating polynomial through the three points (x_0, y_0), (x_1, y_1), and (x_2, y_2) is

$$P_2(x) = y_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}. \tag{12}$$

The Lagrange cubic interpolating polynomial through the four points (x_0, y_0), (x_1, y_1), (x_2, y_2), and (x_3, y_3) is

$$\begin{aligned} P_3(x) = {} & y_0\frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + y_1\frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} \\ & + y_2\frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} + y_3\frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}. \end{aligned} \tag{13}$$

Example 4.7. Consider $y = f(x) = \cos(x)$ over [0.0, 1.2].

- **(a)** Use the three nodes $x_0 = 0.0, x_1 = 0.6$, and $x_2 = 1.2$ to construct a quadratic interpolation polynomial $P_2(x)$.
- **(b)** Use the four nodes $x_0 = 0.0$, $x_1 = 0.4$, $x_2 = 0.8$, and $x_3 = 1.2$ to construct a cubic interpolation polynomial $P_3(x)$.

(a) Using $x_0 = 0.0, x_1 = 0.6$, $x_2 = 1.2$ and $y_0 = \cos(0.0) = 1, y_1 = \cos(0.6) =$

0.825336, and $y_2 = \cos(1.2) = 0.362358$ in equation (12) produces

$$\begin{aligned}
P_2(x) &= 1.0\frac{(x-0.6)(x-1.2)}{(0.0-0.6)(0.0-1.2)} + 0.825336\frac{(x-0.0)(x-1.2)}{(0.6-0.0)(0.6-1.2)}\\
&\quad + 0.362358\frac{(x-0.0)(x-0.6)}{(1.2-0.0)(1.2-0.6)}\\
&= 1.388889(x-0.6)(x-1.2) - 2.292599(x-0.0)(x-1.2)\\
&\quad + 0.503275(x-0.0)(x-0.6).
\end{aligned}$$

(b) Using $x_0 = 0.0$, $x_1 = 0.4$, $x_2 = 0.8$, $x_3 = 1.2$ and $y_0 = \cos(0.0) = 1.0$, $y_1 = \cos(0.4) = 0.921061$, $y_2 = \cos(0.8) = 0.696707$, and $y_3 = \cos(1.2) = 0.362358$ in equation (13) produces

$$\begin{aligned}
P_3(x) &= 1.000000\frac{(x-0.4)(x-0.8)(x-1.2)}{(0.0-0.4)(0.0-0.8)(0.0-1.2)}\\
&\quad + 0.921061\frac{(x-0.0)(x-0.8)(x-1.2)}{(0.4-0.0)(0.4-0.8)(0.4-1.2)}\\
&\quad + 0.696707\frac{(x-0.0)(x-0.4)(x-1.2)}{(0.8-0.0)(0.8-0.4)(0.8-1.2)}\\
&\quad + 0.362358\frac{(x-0.0)(x-0.4)(x-0.8)}{(1.2-0.0)(1.2-0.4)(1.2-0.8)}\\
&= -2.604167(x-0.4)(x-0.8)(x-1.2)\\
&\quad + 7.195789(x-0.0)(x-0.8)(x-1.2)\\
&\quad - 5.443021(x-0.0)(x-0.4)(x-1.2)\\
&\quad + 0.943641(x-0.0)(x-0.4)(x-0.8).
\end{aligned}$$

The graphs of $y = \cos(x)$ and the polynomials $y = P_2(x)$ and $y = P_3(x)$ are shown in Figure 4.12(a) and (b), respectively. ■

Error Terms and Error Bounds

It is important to understand the nature of the error term when the Lagrange polynomial is used to approximate a continuous function $f(x)$. It is similar to the error term for the Taylor polynomial, except that the factor $(x - x_0)^{N+1}$ is replaced with the product $(x - x_0)(x - x_1)\cdots(x - x_N)$. This is expected because interpolation is exact at each of the $N + 1$ nodes x_k, where we have $E_N(x_k) = f(x_k) - P_N(x_k) = y_k - y_k = 0$ for $k = 0, 1, 2, \ldots, N$.

Theorem 4.3 (Lagrange Polynomial Approximation). Assume that $f \in C^{N+1}[a, b]$ and that $x_0, x_1, \ldots, x_N \in [a, b]$ are $N + 1$ nodes. If $x \in [a, b]$, then

$$f(x) = P_N(x) + E_N(x), \tag{14}$$

where $P_N(x)$ is a polynomial that can be used to approximate $f(x)$:

$$f(x) \approx P_N(x) = \sum_{k=0}^{N} f(x_k)L_{N,k}(x). \tag{15}$$

The error term $E_N(x)$ has the form

$$E_N(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_N)f^{(N+1)}(c)}{(N+1)!} \tag{16}$$

for some value $c = c(x)$ that lies in the interval $[a, b]$.

Proof. As an example of the general method, we establish (16) when $N = 1$. The general case is discussed in the exercises. Start by defining the special function $g(t)$ as follows:

$$g(t) = f(t) - P_1(t) - E_1(x)\frac{(t-x_0)(t-x_1)}{(x-x_0)(x-x_1)}. \tag{17}$$

Notice that x, x_0, and x_1 are constants with respect to the variable t and that $g(t)$ evaluates to be zero at these three values; that is,

$$\begin{aligned}
g(x) &= f(x) - P_1(x) - E_1(x)\frac{(x-x_0)(x-x_1)}{(x-x_0)(x-x_1)} = f(x) - P_1(x) - E_1(x) = 0,\\
g(x_0) &= f(x_0) - P_1(x_0) - E_1(x)\frac{(x_0-x_0)(x_0-x_1)}{(x-x_0)(x-x_1)} = f(x_0) - P_1(x_0) = 0,\\
g(x_1) &= f(x_1) - P_1(x_1) - E_1(x)\frac{(x_1-x_0)(x_1-x_1)}{(x-x_0)(x-x_1)} = f(x_1) - P_1(x_1) = 0.
\end{aligned}$$

Suppose that x lies in the open interval (x_0, x_1). Applying Rolle's theorem to $g(t)$ on the interval $[x_0, x]$ produces a value d_0, with $x_0 < d_0 < x$, such that

$$g'(d_0) = 0. \tag{18}$$

A second application of Rolle's theorem to $g(t)$ on $[x, x_1]$ will produce a value d_1, with $x < d_1 < x_1$, such that

$$g'(d_1) = 0. \tag{19}$$

Equations (18) and (19) show that the function $g'(t)$ is zero at $t = d_0$ and $t = d_1$. A third use of Rolle's theorem, but this time applied to $g'(t)$ over $[d_0, d_1]$, produces a value c for which

$$g^{(2)}(c) = 0. \tag{20}$$

Now go back to (17) and compute the derivatives $g'(t)$ and $g''(t)$:

$$g'(t) = f'(t) - P_1'(t) - E_1(x)\frac{(t-x_0)+(t-x_1)}{(x-x_0)(x-x_1)}, \tag{21}$$

$$g''(t) = f''(t) - 0 - E_1(x)\frac{2}{(x-x_0)(x-x_1)}. \tag{22}$$

In (22) we have used the fact the $P_1(t)$ is a polynomial of degree $N = 1$; hence its second derivative is $P_1''(t) \equiv 0$. Evaluation of (22) at the point $t = c$ and using (20) yields

$$0 = f''(c) - E_1(x)\frac{2}{(x-x_0)(x-x_1)}. \tag{23}$$

Solving (23) for $E_1(x)$ results in the desired form (16) for the remainder:

$$E_1(x) = \frac{(x-x_0)(x-x_1)f^{(2)}(c)}{2!}, \tag{24}$$

and the proof is complete. •

The next result addresses the special case when the nodes for the Lagrange polynomial are equally spaced $x_k = x_0 + hk$, for $k = 0, 1, \ldots, N$, and the polynomial $P_N(x)$ is used only for interpolation inside the interval $[x_0, x_N]$.

Theorem 4.4 (Error Bounds for Lagrange Interpolation, Equally Spaced Nodes). Assume that $f(x)$ is defined on $[a, b]$, which contains equally spaced nodes $x_k = x_0 + hk$. Additionally, assume that $f(x)$ and the derivatives of $f(x)$, up to the order $N + 1$, are continuous and bounded on the special subintervals $[x_0, x_1]$, $[x_0, x_2]$, and $[x_0, x_3]$, respectively; that is,

$$|f^{(N+1)}(x)| \le M_{N+1} \qquad \text{for } x_0 \le x \le x_N, \tag{25}$$

for $N = 1, 2, 3$. The error terms (16) corresponding to the cases $N = 1, 2$, and 3 have the following useful bounds on their magnitude:

$$|E_1(x)| \le \frac{h^2 M_2}{8} \qquad \text{valid for } x \in [x_0, x_1], \tag{26}$$

$$|E_2(x)| \le \frac{h^3 M_3}{9\sqrt{3}} \qquad \text{valid for } x \in [x_0, x_2], \tag{27}$$

$$|E_3(x)| \le \frac{h^4 M_4}{24} \qquad \text{valid for } x \in [x_0, x_3]. \tag{28}$$

Proof. We establish (26) and leave the others for the reader. Using the change of variables $x - x_0 = t$ and $x - x_1 = t - h$, the error term $E_1(x)$ can be written as

$$E_1(x) = E_1(x_0 + t) = \frac{(t^2 - ht)f^{(2)}(c)}{2!} \qquad \text{for } 0 \le t \le h. \tag{29}$$

The bound for the derivative for this case is

$$(30) \qquad |f^{(2)}(c)| \le M_2 \quad \text{for } x_0 \le c \le x_1.$$

Now determine a bound for the expression $(t^2 - ht)$ in the numerator of (29); call this term $\Phi(t) = t^2 - ht$. Since $\Phi'(t) = 2t - h$, there is one critical point $t = h/2$ that is the solution to $\Phi'(t) = 0$. The extreme values of $\Phi(t)$ over $[0, h]$ occur either at an end point $\Phi(0) = 0$, $\Phi(h) = 0$ or at the critical point $\Phi(h/2) = -h^2/4$. Since the latter value is the largest, we have established the bound

$$(31) \qquad |\Phi(t)| = |t^2 - ht| \le \frac{|-h^2|}{4} = \frac{h^2}{4} \quad \text{for } 0 \le t \le h.$$

Using (30) and (31) to estimate the magnitude of the product in the numerator in (29) results in

$$(32) \qquad |E_1(x)| = \frac{|\Phi(t)||f^{(2)}(c)|}{2!} \le \frac{h^2 M_2}{8},$$

and formula (26) is established. •

Comparison of Accuracy and $O(h^{N+1})$

The significance of Theorem 4.4 is to understand a simple relationship between the size of the error terms for linear, quadratic, and cubic interpolation. In each case the error bound $|E_N(x)|$ depends on h in two ways. First, h^{N+1} is explicitly present so that $|E_N(x)|$ is proportional to h^{N+1}. Second, the values M_{N+1} generally depend on h and tend to $|f^{(N+1)}(x_0)|$ as h goes to zero. Therefore, as h goes to zero, $|E_N(x)|$ converges to zero with the same rapidity that h^{N+1} converges to zero. The notation $\boldsymbol{O}(h^{N+1})$ is used when discussing this behavior. For example, the error bound (26) can be expressed as

$$|E_1(x)| = \boldsymbol{O}(h^2) \quad \text{valid for } x \in [x_0, x_1].$$

The notation $\boldsymbol{O}(h^2)$ stands in place of $h^2 M_2/8$ in relation (26) and is meant to convey the idea that the bound for the error term is approximately a multiple of h^2; that is,

$$|E_1(x)| \le Ch^2 \approx \boldsymbol{O}(h^2).$$

As a consequence, if the derivatives of $f(x)$ are uniformly bounded on the interval $[a, b]$ and $|h| < 1$, then choosing N large will make h^{N+1} small, and the higher-degree approximating polynomial will have less error.

Program 4.1 (Lagrange Approximation). To evaluate the Lagrange polynomial $P(x) = \sum_{k=0}^{N} y_k L_{N,k}(x)$ based on $N + 1$ points (x_k, y_k) for $k = 0, 1, \ldots, N$.

```
function [C,L]=lagran(X,Y)

%Input  - X is a vector that contains a list of abscissas
%       - Y is a vector that contains a list of ordinates
%Output - C is a matrix that contains the coefficients of
%         the Lagrange interpolatory polynomial
%       - L is a matrix that contains the Lagrange
%         coefficient polynomials

w=length(X);
n=w-1;
L=zeros(w,w);

%Form the Lagrange coefficient polynomials
for k=1:n+1
    V=1;
    for j=1:n+1
    if k~=j
    V=conv(V,poly(X(j)))/(X(k)-X(j));
    end
end
    L(k,:)=V;
end

%Determine the coefficients of the Lagrange interpolating
%polynomial

C=Y*L;
```

Exercises for Lagrange Approximation

1. Find Lagrange polynomials that approximate $f(x) = x^3$.

(a) Find the linear interpolation polynomial $P_1(x)$ using the nodes $x_0 = -1$ and $x_1 = 0$.

(b) Find the quadratic interpolation polynomial $P_2(x)$ using the nodes $x_0 = -1$, $x_1 = 0$, and $x_2 = 1$.

(c) Find the cubic interpolation polynomial $P_3(x)$ using the nodes $x_0 = -1$, $x_1 = 0$, $x_2 = 1$, and $x_3 = 2$.

(d) Find the linear interpolation polynomial $P_1(x)$ using the nodes $x_0 = 1$ and $x_1 = 2$.

(e) Find the quadratic interpolation polynomial $P_2(x)$ using the nodes $x_0 = 0$, $x_1 = 1$, and $x_2 = 2$.

2. Let $f(x) = x + 2/x$.

(a) Use quadratic Lagrange interpolation based on the nodes $x_0 = 1$, $x_1 = 2$, and $x_2 = 2.5$ to approximate $f(1.5)$ and $f(1.2)$.

(b) Use cubic Lagrange interpolation based on the nodes $x_0 = 0.5$, $x_1 = 1$, $x_2 = 2$, and $x_3 = 2.5$ to approximate $f(1.5)$ and $f(1.2)$.

3. Let $f(x) = 2\sin(\pi x/6)$, where x is in radians.

(a) Use quadratic Lagrange interpolation based on the nodes $x_0 = 0$, $x_1 = 1$, and $x_2 = 3$ to approximate $f(2)$ and $f(2.4)$.

(b) Use cubic Lagrange interpolation based on the nodes $x_0 = 0$, $x_1 = 1$, $x_2 = 3$, and $x_3 = 5$ to approximate $f(2)$ and $f(2.4)$.

4. Let $f(x) = 2\sin(\pi x/6)$, where x is in radians.

(a) Use quadratic Lagrange interpolation based on the nodes $x_0 = 0$, $x_1 = 1$, and $x_2 = 3$ to approximate $f(4)$ and $f(3.5)$.

(b) Use cubic Lagrange interpolation based on the nodes $x_0 = 0$, $x_1 = 1$, $x_2 = 3$, and $x_3 = 5$ to approximate $f(4)$ and $f(3.5)$.

5. Write down the error term $E_3(x)$ for cubic Lagrange interpolation to $f(x)$, where interpolation is to be exact at the four nodes $x_0 = -1$, $x_1 = 0$, $x_2 = 3$, and $x_4 = 4$ and $f(x)$ is given by

(a) $f(x) = 4x^3 - 3x + 2$

(b) $f(x) = x^4 - 2x^3$

(c) $f(x) = x^5 - 5x^4$

6. Let $f(x) = x^x$.

(a) Find the quadratic Lagrange polynomial $P_2(x)$ using the nodes $x_0 = 1$, $x_1 = 1.25$, and $x_2 = 1.5$.

(b) Use the polynomial from part (a) to estimate the average value of $f(x)$ over the interval $[1, 1.5]$.

(c) Use expression (27) of Theorem 4.4 to obtain a bound on the error in approximating $f(x)$ with $P_2(x)$.

7. Consider the Lagrange coefficient polynomials $L_{2,k}(x)$ that are used for quadratic interpolation at the nodes x_0, x_1, and x_2. Define $g(x) = L_{2,0}(x) + L_{2,1}(x) + L_{2,2}(x) - 1$.

(a) Show that g is a polynomial of degree ≤ 2.

(b) Show that $g(x_k) = 0$ for $k = 0, 1, 2$.

(c) Show that $g(x) = 0$ for all x. *Hint.* Use the fundamental theorem of algebra.

8. Let $L_{N,0}(x)$, $L_{N,1}(x), \ldots,$ and $L_{N,N}(x)$ be the Lagrange coefficient polynomials based on the $N + 1$ nodes $x_0, x_1, \ldots,$ and x_N. Show that $\sum_{k=0}^{N} L_{N,k}(x) = 1$ for any real number x.

9. Let $f(x)$ be a polynomial of degree $\leq N$. Let $P_N(x)$ be the Lagrange polynomial of degree $\leq N$ based on the $N + 1$ nodes $x_0, x_1, \ldots, x_N$. Show that $f(x) = P_N(x)$ for all x. *Hint.* Show that the error term $E_N(x)$ is identically zero.

Table 4.9 Divided-Difference Table Used for Constructing the Newton Polynomial $P_3(x)$ in Example 4.12.

x_k	$f[x_k]$	First divided difference	Second divided difference	Third divided difference	Fourth divided difference	Fifth divided difference
$x_0 = 1$	$\underline{-3}$					
$x_1 = 2$	0	$\underline{3}$				
$x_2 = 3$	15	15	$\underline{6}$			
$x_3 = 4$	48	33	9	$\underline{1}$		
$x_4 = 5$	105	57	12	1	$\underline{0}$	
$x_5 = 6$	192	87	15	1	0	$\underline{0}$

Table 4.10 Divided-Difference Table Used for Constructing the Newton Polynomials $P_k(x)$ in Example 4.13

x_k	$f[x_k]$	$f[\ ,\ ]$	$f[\ ,\ ,\ ]$	$f[\ ,\ ,\ ,\ ]$	$f[\ ,\ ,\ ,\ ,\ ]$
$x_0 = 0.0$	1.0000000				
$x_1 = 1.0$	0.5403023	−0.4596977			
$x_2 = 2.0$	−0.4161468	−0.9564491	−0.2483757		
$x_3 = 3.0$	−0.9899925	−0.5738457	0.1913017	0.1465592	
$x_4 = 4.0$	−0.6536436	0.3363499	0.4550973	0.0879318	−0.0146568

Example 4.12. Let $f(x) = x^3 - 4x$. Construct the divided-difference table based on the nodes $x_0 = 1, x_1 = 2, \ldots, x_5 = 6$, and find the Newton polynomial $P_3(x)$ based on x_0, x_1, x_2, and x_3.

See Table 4.9. ■

The coefficients $a_0 = -3$, $a_1 = 3$, $a_2 = 6$, and $a_3 = 1$ of $P_3(x)$ appear on the diagonal of the divided-difference table. The centers $x_0 = 1$, $x_1 = 2$, and $x_2 = 3$ are the values in the first column. Using formula (3), we write

$$P_3(x) = -3 + 3(x-1) + 6(x-1)(x-2) + (x-1)(x-2)(x-3).$$

Example 4.13. Construct a divided-difference table for $f(x) = \cos(x)$ based on the five points $(k, \cos(k))$, for $k = 0, 1, 2, 3, 4$. Use it to find the coefficients a_k and the four Newton interpolating polynomials $P_k(x)$, for $k = 1, 2, 3, 4$.

For simplicity we round off the values to seven decimal places, which are displayed in Table 4.10. The nodes x_0, x_1, x_2, x_3 and the diagonal elements a_0, a_1, a_2, a_3, a_4 in

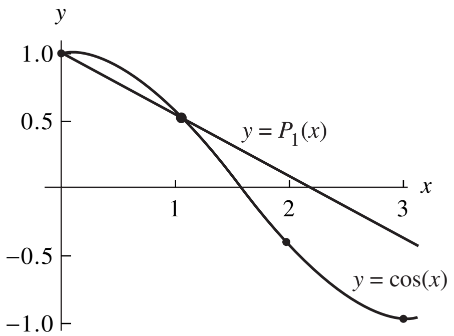


Figure 4.14 (a) The graphs of $y = \cos(x)$ and the linear Newton polynomial $y = P_1(x)$ based on the nodes $x_0 = 0.0$ and $x_1 = 1.0$.

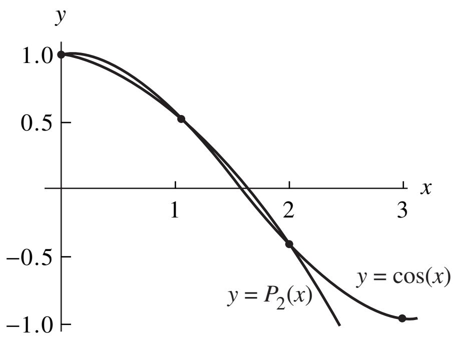


Figure 4.14 (b) The graphs of $y = \cos(x)$ and the quadratic Newton polynomial $y = P_2(x)$ based on the nodes $x_0 = 0.0$, $x_1 = 1.0$, and $x_2 = 2.0$.

Table 4.10 are used in formula (16), and we write down the first four Newton polynomials:

$$\begin{aligned}
P_1(x) &= 1.0000000 - 0.4596977(x - 0.0),\\
P_2(x) &= 1.0000000 - 0.4596977(x - 0.0) - 0.2483757(x - 0.0)(x - 1.0),\\
P_3(x) &= 1.0000000 - 0.4596977(x - 0.0) - 0.2483757(x - 0.0)(x - 1.0)\\
&\quad + 0.1465592(x - 0.0)(x - 1.0)(x - 2.0),\\
P_4(x) &= 1.0000000 - 0.4596977(x - 0.0) - 0.2483757(x - 0.0)(x - 1.0)\\
&\quad + 0.1465592(x - 0.0)(x - 1.0)(x - 2.0)\\
&\quad - 0.0146568(x - 0.0)(x - 1.0)(x - 2.0)(x - 3.0).
\end{aligned}$$

The following sample calculation shows how to find the coefficient a_2.

$$\begin{aligned}
f[x_0, x_1] &= \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{0.5403023 - 10000000}{1.0 - 0.0} = -0.4596977,\\
f[x_1, x_2] &= \frac{f[x_2] - f[x_1]}{x_2 - x_1} = \frac{-0.4161468 - 0.5403023}{2.0 - 1.0} = -0.9564491,\\
a_2 = f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{-0.9564491 + 0.4596977}{2.0 - 0.0} = -0.2483757.
\end{aligned}$$

The graphs of $y = \cos(x)$ and $y = P_1(x)$, $y = P_2(x)$, and $y = P_3(x)$ are shown in Figure 4.14(a), (b), and (c), respectively.

For computational purposes the divided differences in Table 4.8 need to be stored in an array which is chosen to be $D(k, j)$, so that

$$D(k, j) = f[x_{k-j}, x_{k-j+1}, \ldots, x_k] \quad \text{for } j \le k. \tag{19}$$

```
for j=2:n
    for k=j:n
      D(k,j)=(D(k,j-1)-D(k-1,j-1))/(X(k)-X(k-j+1));
   end
end

%Determine the coefficients of the Newton interpolating
%polynomial
C=D(n,n);
for k=(n-1):-1:1
   C=conv(C,poly(X(k)));
   m=length(C);
   C(m)=C(m)+D(k,k);
end
```

Exercises for Newton Polynomials

In Exercises 1 through 4, use the centers x_0, x_1, x_2, and x_3 and the coefficients a_0, a_1, a_2, a_3, and a_4 to find the Newton polynomials $P_1(x)$, $P_2(x)$, $P_3(x)$, and $P_4(x)$, and evaluate them at the value $x = c$. *Hint.* Use equations (1) through (4) and the techniques of Example 4.9.

1. $a_0 = 4$ $a_1 = -1$ $a_2 = 0.4$ $a_3 = 0.01$ $a_4 = -0.002$
$x_0 = 1$ $x_1 = 3$ $x_2 = 4$ $x_3 = 4.5$ $c = 2.5$

2. $a_0 = 5$ $a_1 = -2$ $a_2 = 0.5$ $a_3 = -0.1$ $a_4 = 0.003$
$x_0 = 0$ $x_1 = 1$ $x_2 = 2$ $x_3 = 3$ $c = 2.5$

3. $a_0 = 7$ $a_1 = 3$ $a_2 = 0.1$ $a_3 = 0.05$ $a_4 = -0.04$
$x_0 = -1$ $x_1 = 0$ $x_2 = 1$ $x_3 = 4$ $c = 3$

4. $a_0 = -2$ $a_1 = 4$ $a_2 = -0.04$ $a_3 = 0.06$ $a_4 = 0.005$
$x_0 = -3$ $x_1 = -1$ $x_2 = 1$ $x_3 = 4$ $c = 2$

In Exercises 5 through 8:

(a) Compute the divided-difference table for the tabulated function.

(b) Write down the Newton polynomials $P_1(x)$, $P_2(x)$, $P_3(x)$, and $P_4(x)$.

(c) Evaluate the Newton polynomials in part (b) at the given values of x.

(d) Compare the values in part (c) with the actual function value $f(x)$.

5. $f(x) = x^{1/2}$

$x = 4.5, 7.5$

k	x_k	$f(x_k)$
0	4.0	2.00000
1	5.0	2.23607
2	6.0	2.44949
3	7.0	2.64575
4	8.0	2.82843

6. $f(x) = 3.6/x$

$x = 2.5, 3.5$

k	x_k	$f(x_k)$
0	1.0	3.60
1	2.0	1.80
2	3.0	1.20
3	4.0	0.90
4	5.0	0.72

7. $f(x) = 3\sin^2(\pi x/6)$

$x = 1.5, 3.5$

k	x_k	$f(x_k)$
0	0.0	0.00
1	1.0	0.75
2	2.0	2.25
3	3.0	3.00
4	4.0	2.25

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

$x = 0.5, 1.5$

k	x_k	$f(x_k)$
0	0.0	1.00000
1	1.0	0.36788
2	2.0	0.13534
3	3.0	0.04979
4	4.0	0.01832

9. Consider the $M + 1$ points $(x_0, y_0), \ldots, (x_M, y_M)$.

(a) If the $(N + 1)$st divided differences are zero, then show that the $(N + 2)$nd up to the Mth divided differences are zero.

(b) If the $(N + 1)$st divided differences are zero, then show that there exists a polynomial $P_N(x)$ of degree N such that

$$P_N(x_k) = y_k \qquad \text{for } k = 0,\ 1,\ \ldots,\ M.$$

In Exercises 10 through 12, use the result of Exercise 9 to find the polynomial $P_N(x)$ that goes through the $M + 1$ points $(N < M)$.

10.

x_k	y_k
0	−2
1	2
2	4
3	4
4	2
5	−2

11.

x_k	y_k
1	8
2	17
3	24
4	29
5	32
6	33

12.

x_k	y_k
0	5
1	5
2	3
3	5
4	17
5	45
6	95

5.1 Least-Squares Line

In science and engineering it is often the case that an experiment produces a set of data points $(x_1, y_1), \ldots, (x_N, y_N)$, where the abscissas $\{x_k\}$ are distinct. One goal of numerical methods is to determine a formula $y = f(x)$ that relates these variables. Usually, a class of allowable formulas is chosen and then coefficients must be determined. There are many different possibilities for the type of function that can be used. Often there is an underlying mathematical model, based on the physical situation, that will determine the form of the function. In this section we emphasize the class of linear functions of the form

$$y = f(x) = Ax + B. \tag{1}$$

In Chapter 4 we saw how to construct a polynomial that passes through a set of points. If all the numerical values $\{x_k\}$, $\{y_k\}$ are known to several significant digits of accuracy, then polynomial interpolation can be used successfully; otherwise, it cannot. Some experiments are devised using specialized equipment so that the data points will have at least five digits of accuracy. However, many experiments are done with equipment that is reliable only to three or fewer digits of accuracy. Often, there is an experimental error in the measurements, and although three digits are recorded for the values $\{x_k\}$ and $\{y_k\}$, it is realized that the true value $f(x_k)$ satisfies

$$f(x_k) = y_k + e_k, \tag{2}$$

where e_k is the measurement error.

How do we find the best linear approximation of the form (1) that goes near (not always through) the points? To answer this question, we need to discuss the ***errors*** (also called ***deviations*** or ***residuals***):

$$e_k = f(x_k) - y_k \quad \text{for } 1 \le k \le N. \tag{3}$$

There are several norms that can be used with the residuals in (3) to measure how far the curve $y = f(x)$ lies from the data.

(4) Maximum error: $$E_\infty(f) = \max_{1\le k\le N}\{|f(x_k) - y_k|\},$$

(5) Average error: $$E_1(f) = \frac{1}{N}\sum_{k=1}^{N}|f(x_k) - y_k|,$$

(6) Root-mean-square error: $$E_2(f) = \left(\frac{1}{N}\sum_{k=1}^{N}|f(x_k) - y_k|^2\right)^{1/2}.$$

The next example shows how to apply these norms when a function and a set of points are given.

Table 5.1 Calculations for Finding $E_1(f)$ and $E_2(f)$ for Example 5.1

x_k	y_k	$f(x_k) = 8.6 - 1.6x_k$	$\lvert e_k \rvert$	e_k^2
-1	10.0	10.2	0.2	0.04
0	9.0	8.6	0.4	0.16
1	7.0	7.0	0.0	0.00
2	5.0	5.4	0.4	0.16
3	4.0	3.8	0.2	0.04
4	3.0	2.2	0.8	0.64
5	0.0	0.6	0.6	0.36
6	-1.0	-1.0	0.0	0.00
			2.6	1.40

Example 5.1. Compare the maximum error, average error, and rms error for the linear approximation $y = f(x) = 8.6 - 1.6x$ to the data points $(-1, 10)$, $(0, 9)$, $(1, 7)$, $(2, 5)$, $(3, 4)$, $(4, 3)$, $(5, 0)$, and $(6, -1)$.

The errors are found using the values for $f(x_k)$ and e_k given in Table 5.1.

$$E_\infty(f) = \max\{0.2, 0.4, 0.0, 0.4, 0.2, 0.8, 0.6, 0.0\} = 0.8, \tag{7}$$

$$E_1(f) = \frac{1}{8}(2.6) = 0.325, \tag{8}$$

$$E_2(f) = \left(\frac{1.4}{8}\right)^{1/2} \approx 0.41833. \tag{9}$$

We can see that the maximum error is largest, and if one point is badly in error, its value determines $E_\infty(f)$. The average error $E_1(f)$ simply averages the absolute value of the error at the various points. It is often used because it is easy to compute. The error $E_2(f)$ is often used when the statistical nature of the errors is considered.

A best-fitting line is found by minimizing one of the quantities in equations (4) through (6). Hence there are three best-fitting lines that we could find. The third norm $E_2(f)$ is the traditional choice because it is much easier to minimize computationally. ■

Finding the Least-Squares Line

Let $\{(x_k, y_k)\}_{k=1}^N$ be a set of N points, where the abscissas $\{x_k\}$ are distinct. The ***least-squares line*** $y = f(x) = Ax + B$ is the line that minimizes the root-mean-square error $E_2(f)$.

The quantity $E_2(f)$ will be a minimum if and only if the quantity $N(E_2(f))^2 = \sum_{k=1}^N (Ax_k + B - y_k)^2$ is a minimum. The latter is visualized geometrically by minimizing the sum of the squares of the vertical distances from the points to the line. The next result explains this process.

```
A=sumxy/sumx2;
B=ymean-A*xmean;
```

Exercises for Least-Squares Line

In Exercises 1 and 2, find the least-squares line $y = f(x) = Ax + B$ for the data and calculate $E_2(f)$

1. (a)

x_k	y_k	$f(x_k)$
−2	1	1.2
−1	2	1.9
0	3	2.6
1	3	3.3
2	4	4.0

(b)

x_k	y_k	$f(x_k)$
−6	7	7.0
−2	5	4.6
0	3	3.4
2	2	2.2
6	0	−0.2

(c)

x_k	y_k	$f(x_k)$
−4	−3	−3.0
−1	−1	−0.9
0	0	−0.2
2	1	1.2
3	2	1.9

2. (a)

x_k	y_k	$f(x_k)$
−4	1.2	0.44
−2	2.8	3.34
0	6.2	6.24
2	7.8	9.14
4	13.2	12.04

(b)

x_k	y_k	$f(x_k)$
−6	−5.3	−6.00
−2	−3.5	−2.84
0	−1.7	−1.26
2	0.2	0.32
6	4.0	3.48

(c)

x_k	y_k	$f(x_k)$
−8	6.8	7.32
−2	5.0	3.81
0	2.2	2.64
4	0.5	0.30
6	−1.3	−0.87

3. Find the power fit $y = Ax$, where $M = 1$, which is a line through the origin, for the data and calculate $E_2(f)$.

(a)

x_k	y_k	$f(x_k)$
-4	-3	-2.8
-1	-1	-0.7
0	0	0.0
2	1	1.4
3	2	2.1

(b)

x_k	y_k	$f(x_k)$
3	1.6	1.722
4	2.4	2.296
5	2.9	2.870
6	3.4	3.444
8	4.6	4.592

(c)

x_k	y_k	$f(x_k)$
1	1.6	1.58
2	2.8	3.16
3	4.7	4.74
4	6.4	6.32
5	8.0	7.90

4. Define the means $\overline{x}$ and $\overline{y}$ for the points $\{(x_k, y_k)\}_{k=1}^N$ by

$$\overline{x} = \frac{1}{N}\sum_{k=1}^{N} x_k \quad \text{and} \quad \overline{y} = \frac{1}{N}\sum_{k=1}^{N} y_k.$$

Show that the point $(\overline{x}, \overline{y})$ lies on the least-squares line determined by the given set of points.

5. Show that the solution of the system in (10) is given by

$$A = \frac{1}{D}\left(N\sum_{k=1}^{N} x_k y_k - \sum_{k=1}^{N} x_k \sum_{k=1}^{N} y_k\right),$$

$$B = \frac{1}{D}\left(\sum_{k=1}^{N} x_k^2 \sum_{k=1}^{N} y_k - \sum_{k=1}^{N} x_k \sum_{k=1}^{N} x_k y_k\right),$$

where

$$D = N\sum_{k=1}^{N} x_k^2 - \left(\sum_{k=1}^{N} x_k\right)^2.$$

Hint. Use Gaussian elimination on the system in (10).

6. Show that the value of D in Exercise 5 is nonzero.
Hint. Show that $D = N\sum_{k=1}^{N}(x_k - \overline{x})^2$.

7. Show that the coefficients A and B for the least-squares line can be computed as follows. First compute the means $\overline{x}$ and $\overline{y}$ in Exercise 4, and then perform the calculations:

$$C = \sum_{k=1}^{N}(x_k - \overline{x})^2, \quad A = \frac{1}{C}\sum_{k=1}^{N}(x_k - \overline{x})(y_k - \overline{y}), \quad B = \overline{y} - A\overline{x}.$$

Hint. Use $X_k = x_k - \overline{x}$, $Y_k = y_k - \overline{y}$ and first find the line $Y = AX$.

8. Find the power fits $y = Ax^2$ and $y = Bx^3$ for the following data and use $E_2(f)$ to determine which curve fits best.

(a)

x_k	y_k
2.0	5.1
2.3	7.5
2.6	10.6
2.9	14.4
3.2	19.0

(b)

x_k	y_k
2.0	5.9
2.3	8.3
2.6	10.7
2.9	13.7
3.2	17.0

9. Find the power fits $y = A/x$ and $y = B/x^2$ for the following data and use $E_2(f)$ to determine which curve fits best.

(a)

x_k	y_k
0.5	7.1
0.8	4.4
1.1	3.2
1.8	1.9
4.0	0.9

(b)

x_k	y_k
0.7	8.1
0.9	4.9
1.1	3.3
1.6	1.6
3.0	0.5

10. **(a)** Derive the normal equation for finding the least-squares linear fit through the origin $y = Ax$.

(b) Derive the normal equation for finding the least-squares power fit $y = Ax^2$.

(c) Derive the normal equations for finding the least-squares parabola $y = Ax^2 + B$.

11. Consider the construction of a least-squares line for each of the sets of data points determined by $S_N = \{(k/N, (k/N)^2)\}_{k=1}^{N}$, where $N = 2, 3, 4, \ldots$. Note that for each value of N, the points in S_N all lie on the graph of $f(x) = x^2$ over the closed interval [0, 1]. Let $\overline{x}_N$ and $\overline{y}_N$ be the means for the given data points (see Exercise 4). Let $\widehat{x}$ be the mean of the values of x in the interval [0, 1], and let $\widehat{y}$ be the mean (average) value of $f(x) = x^2$ over the interval [0, 1].

(a) Show $\lim_{N\to\infty} \overline{x}_N = \widehat{x}$.

(b) Show $\lim_{N\to\infty} \overline{y}_N = \widehat{y}$.

12. Consider the construction of a least-squares line for each of the sets of data points:

$$S_N = \left\{((b-a)\frac{k}{N} + a, \; f((b-a)\frac{k}{N} + a))\right\}_{k=1}^{N}$$

for $N = 2, 3, 4, \ldots$. Assume that $y = f(x)$ is an integrable function over the closed interval $[a, b]$. Repeat parts (a) and (b) from Exercise 11.

5.2 Methods of Curve Fitting

Data Linearization Method for $y = Ce^{Ax}$

Suppose that we are given the points $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ and want to fit an exponential curve of the form

$$y = Ce^{Ax}. \tag{1}$$

The first step is to take the logarithm of both sides:

$$\ln(y) = Ax + \ln(C). \tag{2}$$

Then introduce the change of variables:

$$Y = \ln(y), \qquad X = x, \qquad \text{and} \qquad B = \ln(C). \tag{3}$$

This results in a linear relation between the new variables X and Y:

$$Y = AX + B. \tag{4}$$

The original points (x_k, y_k) in the xy-plane are transformed into the points $(X_k, Y_k) = (x_k, \ln(y_k))$ in the XY-plane. This process is called ***data linearization***. Then the least-squares line (4) is fit to the points $\{(X_k, Y_k)\}$. The normal equations for finding A and B are

$$\begin{aligned} \left(\sum_{k=1}^{N} X_k^2\right) A + \left(\sum_{k=1}^{N} X_k\right) B &= \sum_{k=1}^{N} X_k Y_k, \\ \left(\sum_{k=1}^{N} X_k\right) A + \qquad NB \qquad &= \sum_{k=1}^{N} Y_k. \end{aligned} \tag{5}$$

After A and B have been found, the parameter C in equation (1) is computed:

$$C = e^B. \tag{6}$$

Example 5.4. Use the data linearization method and find the exponential fit $y = Ce^{Ax}$ for the five data points (0, 1.5), (1, 2.5), (2, 3.5), (3, 5.0), and (4, 7.5).

Apply the transformation (3) to the original points and obtain

$$\begin{aligned} \{(X_k, Y_k)\} &= \{(0, \ln(1.5), (1, \ln(2.5)), (2, \ln(3.5)), (3, \ln(5.0)), (4, \ln(7.5))\} \\ &= \{(0, 0.40547), (1, 0.91629), (2, 1.25276), (3, 1.60944), (4, 2.01490)\}. \end{aligned} \tag{7}$$

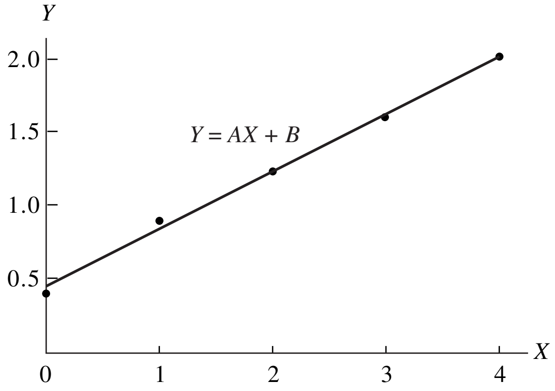


Figure 5.4 The transformed data points $\{(X_k, Y_k)\}$.

Table 5.4 Obtaining Coefficients of the Normal Equations for the Transformed Data Points $\{(X_k, Y_k)\}$

x_k	y_k	X_k	$Y_k = \ln(y_k)$	X_k^2	$X_k Y_k$
0.0	1.5	0.0	0.405465	0.0	0.000000
1.0	2.5	1.0	0.916291	1.0	0.916291
2.0	3.5	2.0	1.252763	4.0	2.505526
3.0	5.0	3.0	1.609438	9.0	4.828314
4.0	7.5	4.0	2.014903	16.0	8.059612
		10.0 $= \sum X_k$	6.198860 $= \sum Y_k$	30.0 $= \sum X_k^2$	16.309743 $= \sum X_k Y_k$

These transformed points are shown in Figure 5.4 and exhibit a linearized form. The equation of the least-squares line $Y = AX + B$ for the points (7) in Figure 5.4 is

$$Y = 0.391202X + 0.457367. \tag{8}$$

Calculation of the coefficients for the normal equations in (5) is shown in Table 5.4.

The resulting linear system (5) for determining A and B is

$$\begin{aligned} 30A + 10B &= 16.309742 \\ 10A + 5B &= 6.198860. \end{aligned} \tag{9}$$

The solution is $A = 0.3912023$ and $B = 0.457367$. Then C is obtained with the calculation $C = e^{0.457367} = 1.579910$, and these values for A and C are substituted into equation (1) to obtain the exponential fit (see Figure 5.5):

$$y = 1.579910e^{0.3912023x} \quad \text{(fit by data linearization).} \tag{10}$$ ■

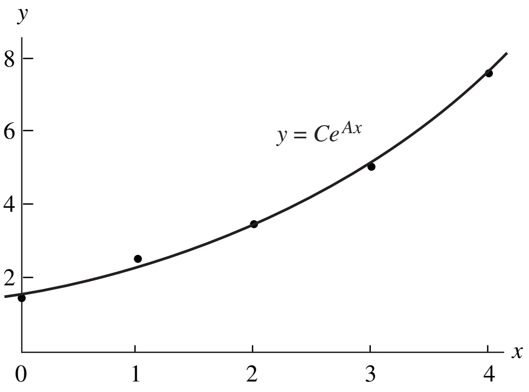


Figure 5.5 The exponential fit $y = 1.579910e^{0.3912023x}$ obtained by using the data linearization method.

Nonlinear Least-Squares Method for $y = Ce^{Ax}$

Suppose that we are given the points $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ and want to fit an exponential curve:

$$y = Ce^{Ax}. \tag{11}$$

The nonlinear least-squares procedure requires that we find a minimum of

$$E(A, C) = \sum_{k=1}^{N} (Ce^{Ax_k} - y_k)^2. \tag{12}$$

The partial derivatives of $E(A, C)$ with respect to A and C are

$$\frac{\partial E}{\partial A} = 2 \sum_{k=1}^{N} (Ce^{Ax_k} - y_k)(Cx_k e^{Ax_k}) \tag{13}$$

and

$$\frac{\partial E}{\partial C} = 2 \sum_{k=1}^{N} (Ce^{Ax_k} - y_k)(e^{Ax_k}). \tag{14}$$

When the partial derivatives in (13) and (14) are set equal to zero and then simplified, the resulting normal equations are

$$\begin{aligned} C \sum_{k=1}^{N} x_k e^{2Ax_k} - \sum_{k=1}^{N} x_k y_k e^{Ax_k} &= 0, \\ C \sum_{k=1}^{N} e^{Ax_k} - \sum_{k=1}^{N} y_k e^{Ax_k} &= 0. \end{aligned} \tag{15}$$

The equations in (15) are nonlinear in the unknowns A and C and can be solved using Newton's method. This is a time-consuming computation and the iteration involved requires good starting values for A and C. Many software packages have a built-in minimization subroutine for functions of several variables that can be used to minimize $E(A, C)$ directly, For example, the Nelder-Mead simplex algorithm can be used to minimize (12) directly and bypass the need for equations (13) through (15).

Example 5.5. Use the least-squares method and determine the exponential fit $y = Ce^{Ax}$ for the five data points (0, 1.5), (1, 2.5), (2, 3.5), (3, 5.0), and (4, 7.5).

For this solution we must minimize the quantity $E(A, C)$, which is

$$\begin{aligned}E(A, C) &= (C - 1.5)^2 + (Ce^A - 2.5)^2 + (Ce^{2A} - 3.5)^2 \\ &\quad + (Ce^{3A} - 5.0)^2 + (Ce^{4A} - 7.5)^2.\end{aligned} \tag{16}$$

We use the `fmins` command in MATLAB to approximate the values of A and C that minimize $E(A, C)$. First we define $E(A, C)$ as an M-file in MATLAB.

```
function z=E(u)
A=u(1);
C=u(2);
z=(C-1.5).^2+(C.*exp(A)-2.5).^2+(C.*exp(2*A)-3.5).^2+...
   (C.*exp(3*A)-5.0).^2+(C.*exp(4*A)-7.5).^2;
```

Using the `fmins` command in the MATLAB Command Window and the initial values $A = 1.0$ and $C = 1.0$, we find

```
>>fmins('E',[1 1])
ans =
   0.38357046980073 1.61089952247928
```

Thus the exponential fit to the five data points is

$$y = 1.6108995e^{0.3835705} \quad \text{(fit by nonlinear least squares).} \tag{17}$$

A comparison of the solutions using data linearization and nonlinear least squares is given in Table 5.5. There is a slight difference in the coefficients. For the purpose of interpolation it can be seen that the approximations differ by no more than 2% over the interval [0, 4] (see Table 5.5 and Figure 5.6). If there is a normal distribution of the errors in the data, (17) is usually the preferred choice. When extrapolation is made beyond the range of the data, the two solutions will diverge and the discrepancy increases to about 6% when $x = 10$. ■

Transformations for Data Linearization

The technique of data linearization has been used by scientists to fit curves such as $y = Ce^{(Ax)}$, $y = A\ln(x) + B$, and $y = A/x + B$. Once the curve has been chosen, a suitable transformation of the variables must be found so that a linear relation is

Table 5.5 Comparison of the Two Exponential Fits

x_k	y_k	$1.5799e^{0.39120x}$	$1.6109e^{0.38357x}$
0.0	1.5	1.5799	1.6109
1.0	2.5	2.3363	2.3640
2.0	3.5	3.4548	3.4692
3.0	5.0	5.1088	5.0911
4.0	7.5	7.5548	7.4713
5.0		11.1716	10.9644
6.0		16.5202	16.0904
7.0		24.4293	23.6130
8.0		36.1250	34.6527
9.0		53.4202	50.8535
10.0		78.9955	74.6287

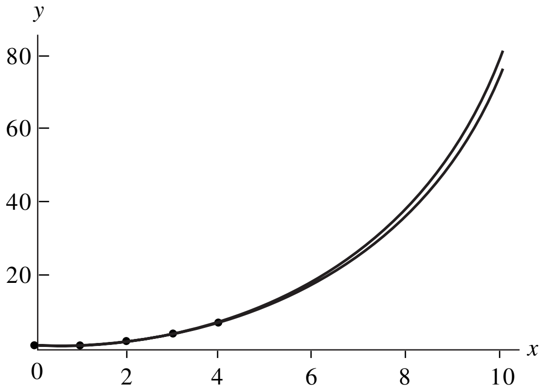


Figure 5.6 A graphical comparison of the two exponential curves.

obtained. For example, the reader can verify that $y = D/(x + C)$ is transformed into a linear problem $Y = AX + B$ by using the change of variables (and constants) $X = xy, Y = y, C = -1/A$, and $D = -B/A$. Graphs of several cases of the possibilities for the curves are shown in Figure 5.7, and other useful transformations are given in Table 5.6.

Linear Least Squares

The linear least-squares problem is stated as follows. Suppose that N data points $\{(x_k, y_k)\}$ and a set of M linear independent functions $\{f_j(x)\}$ are given. We want

Table 5.6 Change of Variable(s) for Data Linearization

Function, $y = f(x)$	Linearized form, $Y = AX + B$	Change of variable(s) and constants
$y = \frac{A}{x} + B$	$y = A\frac{1}{x} + B$	$X = \frac{1}{x}$, $Y = y$
$y = \frac{D}{x + C}$	$y + \frac{-1}{C}(xy) + \frac{D}{C}$	$X = xy$, $Y = y$
		$C = \frac{-1}{A}$, $D = \frac{-B}{A}$
$y = \frac{1}{Ax + B}$	$\frac{1}{y} = Ax + B$	$X = x$, $Y = \frac{1}{y}$
$y = \frac{x}{Ax + B}$	$\frac{1}{y} = A\frac{1}{x} + B$	$X = \frac{1}{x}$, $Y = \frac{1}{y}$
$y = A\ln(x) + B$	$y = A\ln(x) + B$	$X = \ln(x)$, $Y = y$
$y = Ce^{Ax}$	$\ln(y) = Ax + \ln(C)$	$X = x$, $Y = \ln(y)$
		$C = e^B$
$y = Cx^A$	$\ln(y) = A\ln(x) + \ln(C)$	$X = \ln(x)$, $Y = \ln(y)$
		$C = e^B$
$y = (Ax + B)^{-2}$	$y^{-1/2} = Ax + B$	$X = x$, $Y = y^{-1/2}$
$y = Cxe^{-Dx}$	$\ln\left(\frac{y}{x}\right) = -Dx + \ln(C)$	$X = x$, $Y = \ln\left(\frac{y}{x}\right)$
		$C = e^B$, $D = -A$
$y = \frac{L}{1 + Ce^{Ax}}$	$\ln\left(\frac{L}{y} - 1\right) = Ax + \ln(C)$	$X = x$, $Y = \ln\left(\frac{L}{y} - 1\right)$
		$C = e^B$ and L is a constant that must be given

to find M coefficients $\{c_j\}$ so that the function $f(x)$ given by the linear combination

$$f(x) = \sum_{j=1}^{M} c_j f_j(x) \tag{18}$$

will minimize the sum of the squares of the errors:

$$E(c_1, c_2, \ldots, c_M) = \sum_{k=1}^{N} (f(x_k) - y_k)^2 = \sum_{k=1}^{N} \left(\left(\sum_{j=1}^{M} c_j f_j(x_k) \right) - y_k \right)^2. \tag{19}$$

Table 5.7 Obtaining the Coefficients for the Least-Squares Parabola of Example 5.6

x_k	y_k	x_k^2	x_k^3	x_k^4	$x_k y_k$	$x_k^2 y_k$
−3	3	9	−27	81	−9	27
0	1	0	0	0	0	0
2	1	4	8	16	2	4
4	3	16	64	256	12	48
3	8	29	45	353	5	79

Proof. The coefficients A, B, and C will minimize the quantity:

$$E(A, B, C) = \sum_{k=1}^{N} (Ax_k^2 + Bx_k + C - y_k)^2. \tag{29}$$

The partial derivatives $\partial E/\partial A$, $\partial E/\partial B$, and $\partial E/\partial C$ must all be zero. This results in

$$\begin{aligned} 0 &= \frac{\partial E(A, B, C)}{\partial A} = 2\sum_{k=1}^{N} (Ax_k^2 + Bx_k + C - y_k)^1 (x_k^2), \\ 0 &= \frac{\partial E(A, B, C)}{\partial B} = 2\sum_{k=1}^{N} (Ax_k^2 + Bx_k + C - y_k)^1 (x_k), \\ 0 &= \frac{\partial E(A, B, C)}{\partial C} = 2\sum_{k=1}^{N} (Ax_k^2 + Bx_k + C - y_k)^1 (1). \end{aligned} \tag{30}$$

Using the distributive property of addition, we can move the values A, B, and C outside the summations in (30) to obtain the normal equations that are given in (28). •

Example 5.6. Find the least-squares parabola for the four points (−3, 3), (0, 1), (2, 1), and (4, 3).

The entries in Table 5.7 are used to compute the summations required in the linear system (28).

The linear system (28) for finding A, B, and C becomes

$$\begin{aligned} 353A + 45B + 29C &= 79 \\ 45A + 29B + \ \ 3C &= \ \ 5 \\ 29A + \ \ 3B + \ \ 4C &= \ \ 8. \end{aligned}$$

The solution to the linear system is $A = 585/3278$, $B = -631/3278$, and $C = 1394/1639$, and the desired parabola is (see Figure 5.8)

$$y = \frac{585}{3278}x^2 - \frac{631}{3278}x + \frac{1394}{1639} = 0.178462x^2 - 0.192495x + 0.850519. \qquad \blacksquare$$

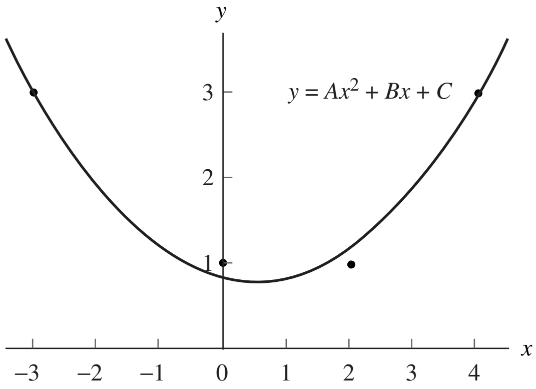


Figure 5.8 The least-squares parabola for Example 5.6.

Polynomial Wiggle

It is tempting to use a least-squares polynomial to fit data that are nonlinear. But if the data do not exhibit a polynomial nature, the resulting curve may exhibit large oscillations. This phenomenon, called ***polynomial wiggle***, becomes more pronounced with higher-degree polynomials. For this reason we seldom use a polynomial of degree 6 or above unless it is known that the true function we are working with is a polynomial.

For example, let $f(x) = 1.44/x^2 + 0.24x$ be used to generate the six data points $(0.25, 23.1)$, $(1.0, 1.68)$, $(1.5, 1.0)$, $(2.0, 0.84)$, $(2.4, 0.826)$, and $(5.0, 1.2576)$. The result of curve fitting with the least-squares polynomials

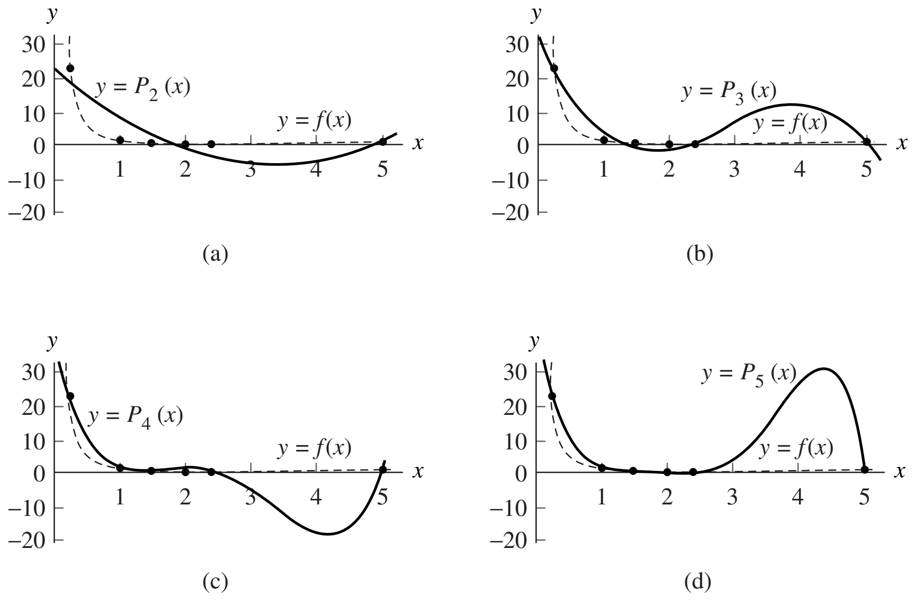
$$\begin{aligned}
P_2(x) &= 22.93 - 16.96x + 2.553x^2,\\
P_3(x) &= 33.04 - 46.51x + 19.51x^2 - 2.296x^3,\\
P_4(x) &= 39.92 - 80.93x + 58.39x^2 - 17.15x^3 + 1.680x^4,
\end{aligned}$$

and

$$P_5(x) = 46.02 - 118.1x + 119.4x^2 - 57.51x^3 + 13.03x^4 - 1.085x^5$$

is shown in Figure 5.9(a) through (d). Notice that $P_3(x)$, $P_4(x)$, and $P_5(x)$ exhibit a large wiggle in the interval $[2, 5]$. Even though $P_5(x)$ goes through the six points, it produces the worst fit. If we must fit a polynomial to these data, $P_2(x)$ should be the choice.

The following program uses the matrix $\boldsymbol{F}$ with entries $f_j(x) = x_k^{j-1}$ from equation (18).



(a)

(b)

(c)

(d)

Figure 5.9 (a) Using $P_2(x)$ to fit data. (b) Using $P_3(x)$ to fit data. (c) Using $P_4(x)$ to fit data. (d) Using $P_5(x)$ to fit data.

Program 5.2 (Least-Squares Polynomial). To construct the least-squares polynomial of degree M of the form

$$P_M(x) = c_1 + c_2x + c_3x^2 + \cdots + c_Mx^{M-1} + c_{M+1}x^M$$

that fits the N data points $\{(x_k, y_k)\}_{k=1}^{N}$.

```
function C = lspoly(X,Y,M)
%Input   - X is the 1xn abscissa vector
%        - Y is the 1xn ordinate vector
%        - M is the degree of the least-squares polynomial
% Output - C is the coefficient list for the polynomial
n=length(X);
B=zeros(1:M+1);
F=zeros(n,M+1);
%Fill the columns of F with the powers of X
for k=1:M+1
   F(:,k)=X'.^(k-1);
end
%Solve the linear system from (25)
```

```
A=F'*F;
B=F'*Y';
C=A\B;
C=flipud(C);
```

Exercises for Methods of Curve Fitting

1. Find the least-squares parabola $f(x) = Ax^2 + Bx + C$ for each set of data.

(a)

x_k	y_k
−3	15
−1	5
1	1
3	5

(b)

x_k	y_k
−3	−1
−1	25
1	25
3	1

2. Find the least-squares parabola $f(x) = Ax^2 + Bx + C$ for each set of data.

(a)

x_k	y_k
−2	−5.8
−1	1.1
0	3.8
1	3.3
2	−1.5

(b)

x_k	y_k
−2	2.8
−1	2.1
0	3.25
1	6.0
2	11.5

(c)

x_k	y_k
−2	10
−1	1
0	0
1	2
2	9

3. For the given set of data, find the least-squares curve:

(a) $f(x) = Ce^{Ax}$, by using the change of variables $X = x$, $Y = \ln(y)$, and $C = e^B$, from Table 5.6, to linearize the data points.

(b) $f(x) = Cx^A$, by using the change of variables $X = \ln(x)$, $Y = \ln(y)$, and $C = e^B$, from Table 5.6, to linearize the data points.

(c) Use $E_2(f)$ to determine which curve gives the best fit.

x_k	y_k
1	0.6
2	1.9
3	4.3
4	7.6
5	12.6

(a) Assume that $L = 8 \times 10^8$.

Year	t_k	P_k
1800	-10	5.3
1850	-5	23.2
1900	0	76.1
1950	5	152.3

(b) Assume that $L = 8 \times 10^8$.

Year	t_k	P_k
1900	0	76.1
1920	2	106.5
1940	4	132.6
1960	6	180.7
1980	8	226.5

In Exercises 8 through 15, carry out the indicated change of variables in Table 5.6, and derive the linearized form for each of the following functions.

8. $y = \dfrac{A}{x} + B$

9. $y = \dfrac{D}{x + C}$

10. $y = \dfrac{1}{Ax + B}$

11. $y = \dfrac{x}{A + Bx}$

12. $y = A\ln(x) + B$

13. $y = Cx^A$

14. $y = (Ax + B)^{-2}$

15. $y = Cxe^{-Dx}$

16. **(a)** Follow the procedure outlined in the proof of Theorem 5.3 and derive the normal equations for the least-squares curve $f(x) = A\cos(x) + B\sin(x)$.

(b) Use the results from part (a) to find the least-squares curve $f(x) = A\cos(x) + B\sin(x)$ for the following data:

x_k	y_k
-3.0	-0.1385
-1.5	-2.1587
0.0	0.8330
1.5	2.2774
3.0	-0.5110

17. The least-squares plane $z = Ax + By + C$ for the N points $(x_1, y_1, z_1), \ldots, (x_N, y_N, z_N)$ is obtained by minimizing

$$E(A, B, C) = \sum_{k=1}^{N} (Ax_k + By_k + C - z_k)^2.$$

(b) Determine $E_2(f)$.

(c) Plot the data and the least-squares curve from part (a) on the same coordinate system.

Time, p.m.	Degrees	Time, a.m.	Degrees
1	66	1	58
2	66	2	58
3	65	3	58
4	64	4	58
5	63	5	57
6	63	6	57
7	62	7	57
8	61	8	58
9	60	9	60
10	60	10	64
11	59	11	67
Midnight	58	Noon	68

5.3 Interpolation by Spline Functions

Polynomial interpolation for a set of $N+1$ points $\{(x_k, y_k)\}_{k=0}^{N}$ is frequently unsatisfactory. As discussed in Section 5.2, a polynomial of degree N can have $N-1$ relative maxima and minima, and the graph can wiggle in order to pass through the points. Another method is to piece together the graphs of lower-degree polynomials $S_k(x)$ and interpolate between the successive nodes (x_k, y_k) and (x_{k+1}, y_{k+1}) (see Figure 5.10).

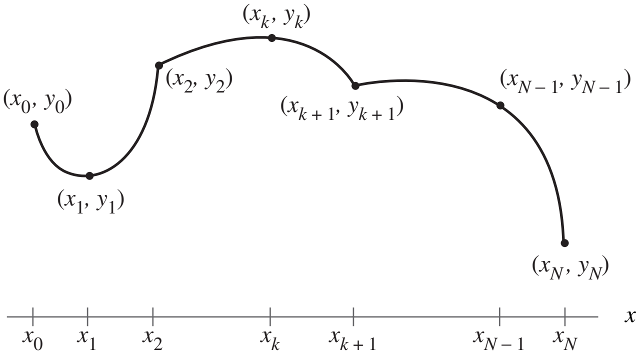


Figure 5.10 Piecewise polynomial interpolation.

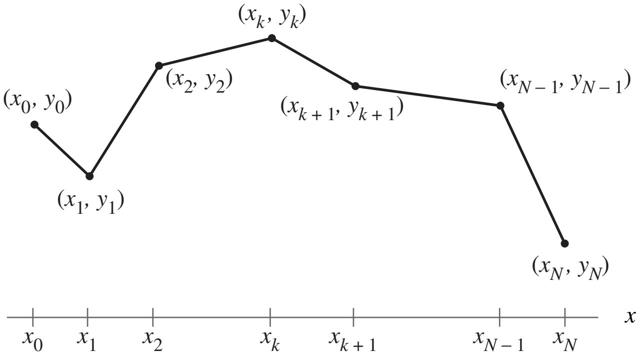


Figure 5.11 Piecewise linear interpolation (a linear spline).

The two adjacent portions of the curve $y = S_k(x)$ and $y = S_{k+1}(x)$, which lie above $[x_k, x_{k+1}]$ and $[x_{k+1}, x_{k+2}]$, respectively, pass through the common ***knot*** (x_{k+1}, y_{k+1}). The two portions of the graph are tied together at the knot (x_{k+1}, y_{k+1}), and the set of functions $\{S_k(x)\}$ forms a piecewise polynomial curve, which is denoted by $S(x)$.

Piecewise Linear Interpolation

The simplest polynomial to use, a polynomial of degree 1, produces a polygonal path that consists of line segments that pass through the points. The Lagrange polynomial from Section 4.3 is used to represent this piecewise linear curve:

$$(1) \qquad S_k(x) = y_k \frac{x - x_{k+1}}{x_k - x_{k+1}} + y_{k+1} \frac{x - x_k}{x_{k+1} - x_k} \qquad \text{for } x_k \le x \le x_{k+1}.$$

The resulting curve looks like a broken line (see Figure 5.11).

An equivalent expression can be obtained if we use the point-slope formula for a line segment:

$$S_k(x) = y_k + d_k(x - x_k),$$

where $d_k = (y_{k+1} - y_k)/(x_{k+1} - x_k)$. The resulting ***linear spline*** function can be written in the form

$$(2) \qquad S(x) = \begin{cases} y_0 + d_0(x - x_0) & \text{for } x \text{ in } [x_0, x_1], \\ y_1 + d_1(x - x_1) & \text{for } x \text{ in } [x_1, x_2], \\ \vdots & \vdots \\ y_k + d_k(x - x_k) & \text{for } x \text{ in } [x_k, x_{k+1}], \\ \vdots & \vdots \\ y_{N-1} + d_{N-1}(x - x_{N-1}) & \text{for } x \text{ in } [x_{N-1}, x_N]. \end{cases}$$

The form of equation (2) is better than equation (1) for the explicit calculation of $S(x)$. It is assumed that the abscissas are ordered $x_0 < x_1 < \cdots < x_{N-1} < x_N$. For a fixed value of x, the interval $[x_k, x_{k+1}]$ containing x can be found by successively computing the differences $x - x_1, \ldots, x - x_k, x - x_{k+1}$ until $k + 1$ is the smallest integer such that $x - x_{k+1} < 0$. Hence we have found k so that $x_k \le x \le x_{k+1}$, and the value of the spline function $S(x)$ is

$$(3) \qquad S(x) = S_k(x) = y_k + d_k(x - x_k) \qquad \text{for } x_k \le x \le x_{k+1}.$$

These techniques can be extended to higher-order polynomials. For example, if an odd number of nodes $x_0, x_1, \ldots, x_{2M}$ is given, then a piecewise quadratic polynomial can be constructed on each subinterval $[x_{2k}, x_{2k+2}]$, for $k = 0, 1, \ldots, M - 1$. A shortcoming of the resulting quadratic spline is that the curvature at the even nodes x_{2k} changes abruptly, and this can cause an undesired bend or distortion in the graph. The second derivative of a quadratic spline is discontinuous at the even nodes. If we use piecewise cubic polynomials, then both the first and second derivatives can be made continuous.

Piecewise Cubic Splines

The fitting of a polynomial curve to a set of data points has applications in CAD (computer-assisted design), CAM (computer-assisted manufacturing), and computer graphics systems. An operator wants to draw a smooth curve through data points that are not subject to error. Traditionally, it was common to use a french curve or an architect's spline and subjectively draw a curve that looks smooth when viewed by the eye. Mathematically, it is possible to construct cubic functions $S_k(x)$ on each interval $[x_k, x_{k+1}]$ so that the resulting piecewise curve $y = S(x)$ and its first and second derivatives are all continuous on the larger interval $[x_0, x_N]$. The continuity of $S'(x)$ means that the graph $y = S(x)$ will not have sharp corners. The continuity of $S''(x)$ means that the *radius of curvature* is defined at each point.

Definition 5.1. Suppose that $\{(x_k, y_k)\}_{k=0}^{N}$ are $N + 1$ points, where $a = x_0 < x_1 < \cdots < x_N = b$. The function $S(x)$ is called a ***cubic spline*** if there exist N cubic polynomials $S_k(x)$ with coefficients $s_{k,0}$, $s_{k,1}$, $s_{k,2}$, and $s_{k,3}$ that satisfy the following properties:

I. $S(x) = S_k(x) = s_{k,0} + s_{k,1}(x - x_k) + s_{k,2}(x - x_k)^2 + s_{k,3}(x - x_k)^3$
for $x \in [x_k, x_{k+1}]$ and $k = 0, 1, \ldots, N - 1$.

II. $S(x_k) = y_k$ for $k = 0, 1, \ldots, N$.

III. $S_k(x_{k+1}) = S_{k+1}(x_{k+1})$ for $k = 0, 1, \ldots, N - 2$.

IV. $S'_k(x_{k+1}) = S'_{k+1}(x_{k+1})$ for $k = 0, 1, \ldots, N - 2$.

V. $S''_k(x_{k+1}) = S''_{k+1}(x_{k+1})$ for $k = 0, 1, \ldots, N - 2$. ▲

Property I states that $S(x)$ consists of piecewise cubics. Property II states that the piecewise cubics interpolate the given set of data points. Properties III and IV require that the piecewise cubics represent a smooth continuous function. Property V states that the second derivative of the resulting function is also continuous.

Existence of Cubic Splines

Let us try to determine if it is possible to construct a cubic spline that satisfies properties I through V. Each cubic polynomial $S_k(x)$ has four unknown constants ($s_{k,0}$, $s_{k,1}$, $s_{k,2}$, and $s_{k,3}$); hence there are $4N$ coefficients to be determined. Loosely speaking, we have $4N$ degrees of freedom or conditions that must be specified. The data points supply $N + 1$ conditions, and properties III, IV, and V each supply $N - 1$ conditions. Hence, $N + 1 + 3(N - 1) = 4N - 2$ conditions are specified. This leaves us two additional degrees of freedom. We will call them ***endpoint constraints***: they will involve either $S'(x)$ or $S''(x)$ at x_0 and x_N and will be discussed later. We now proceed with the construction.

Since $S(x)$ is piecewise cubic, its second derivative $S''(x)$ is piecewise linear on $[x_0, x_N]$. The linear Lagrange interpolation formula gives the following representation for $S''(x) = S''_k(x)$:

$$S''_k(x) = S''(x_k)\frac{x - x_{k+1}}{x_k - x_{k+1}} + S''(x_{k+1})\frac{x - x_k}{x_{k+1} - x_k}. \tag{4}$$

Use $m_k = S''(x_k)$, $m_{k+1} = S''(x_{k+1})$, and $h_k = x_{k+1} - x_k$ in (4) to get

$$S''_k(x) = \frac{m_k}{h_k}(x_{k+1} - x) + \frac{m_{k+1}}{h_k}(x - x_k) \tag{5}$$

for $x_k \le x \le x_{k+1}$ and $k = 0, 1, \ldots, N - 1$. Integrating (5) twice will introduce two constants of integration, and the result can be manipulated so that it has the form

$$S_k(x) = \frac{m_k}{6h_k}(x_{k+1} - x)^3 + \frac{m_{k+1}}{6h_k}(x - x_k)^3 \quad + p_k(x_{k+1} - x) + q_k(x - x_k). \tag{6}$$

Substituting x_k and x_{k+1} into equation (6) and using the values $y_k = S_k(x_k)$ and $y_{k+1} = S_k(x_{k+1})$ yields the following equations that involve p_k and q_k, respectively:

$$y_k = \frac{m_k}{6}h_k^2 + p_k h_k \qquad \text{and} \qquad y_{k+1} = \frac{m_{k+1}}{6}h_k^2 + q_k h_k. \tag{7}$$

These two equations are easily solved for p_k and q_k, and when these values are substituted into equation (6), the result is the following expression for the cubic function $S_k(x)$:

$$\begin{aligned} S_k(x) = &-\frac{m_k}{6h_k}(x_{k+1} - x)^3 + \frac{m_{k+1}}{6h_k}(x - x_k)^3 \\ &+ \left(\frac{y_k}{h_k} - \frac{m_k h_k}{6}\right)(x_{k+1} - x) + \left(\frac{y_{k+1}}{h_k} - \frac{m_{k+1} h_k}{6}\right)(x - x_k). \end{aligned} \tag{8}$$

Notice that the representation (8) has been reduced to a form that involves only the unknown coefficients $\{m_k\}$. To find these values, we must use the derivative of (8), which is

$$\begin{aligned} S_k'(x) &= -\frac{m_k}{2h_k}(x_{k+1} - x)^2 + \frac{m_{k+1}}{2h_k}(x - x_k)^2 \\ &\quad - \left(\frac{y_k}{h_k} - \frac{m_k h_k}{6}\right) + \frac{y_{k+1}}{h_k} - \frac{m_{k+1}h_k}{h_k}. \end{aligned} \tag{9}$$

Evaluating (9) at x_k and simplifying the result yield

$$S_k'(x_k) = -\frac{m_k}{3}h_k - \frac{m_{k+1}}{6}h_k + d_k, \qquad \text{where } d_k = \frac{y_{k+1} - y_k}{h_k}. \tag{10}$$

Similarly, we can replace k by $k - 1$ in (9) to get the expression for $S_{k-1}'(x)$ and evaluate it at x_k to obtain

$$S_{k-1}'(x_k) = \frac{m_k}{3}h_{k-1} + \frac{m_{k-1}}{6}h_{k-1} + d_{k-1}. \tag{11}$$

Now use property IV and equations (10) and (11) to obtain an important relation involving m_{k-1}, m_k, and m_{k+1}:

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = u_k, \tag{12}$$

where $u_k = 6\,(d_k - d_{k-1})$ for $k = 1, 2, \ldots, N - 1$.

Construction of Cubic Splines

Observe that the unknowns in (12) are the desired values $\{m_k\}$, and the other terms are constants obtained by performing simple arithmetic with the data points $\{(x_k, y_k)\}$. Therefore, in reality, system (12) is an underdetermined system of $N - 1$ linear equations involving $N + 1$ unknowns. Hence two additional equations must be supplied. They are used to eliminate m_0 from the first equation and m_N from the $(N - 1)$st equation in system (12). The standard strategies for the endpoint constraints are summarized in Table 5.8.

Consider strategy (v) in Table 5.8. If m_0 is given, then $h_0 m_0$ can be computed, and the first equation (when $k = 1$) of (12) is

$$2(h_0 + h_1)m_1 + h_1 m_2 = u_1 - h_0 m_0. \tag{13}$$

Table 5.8 Endpoint Constraints for a Cubic Spline

	Description of the strategy	Equations involving m_0 and m_N
(i)	*Clamped cubic spline:* specify $S'(x_0)$, $S'(x_n)$ (the "best choice" if the derivatives are known)	$m_0 = \dfrac{3}{h_0}(d_0 - S'(x_0)) - \dfrac{m_1}{2}$ $m_N = \dfrac{3}{h_{N-1}}(S'(x_N) - d_{N-1}) - \dfrac{m_{N-1}}{2}$
(ii)	*Natural cubic spline* (a "relaxed curve")	$m_0 = 0, m_N = 0$
(iii)	Extrapolate $S''(x)$ to the endpoints	$m_0 = m_1 - \dfrac{h_0(m_2 - m_1)}{h_1},$ $m_N = m_{N-1} + \dfrac{h_{N-1}(m_{N-1} - m_{N-2})}{h_{N-2}}$
(iv)	$S''(x)$ is constant near the endpoints	$m_0 = m_1, m_N = m_{N-1}$
(v)	Specify $S''(X)$ at each endpoint	$m_0 = S''(x_0), m_N = S''(x_N)$

Similarly, if m_N is given, then $h_{N-1}m_N$ can be computed, and the last equation (when $k = N - 1$) of (12) is

$$h_{N-2}m_{N-2} + 2(h_{N-2} + h_{N-1})m_{N-1} = u_{N-1} - h_{N-1}m_N. \tag{14}$$

Equations (13) and (14) with (12) used for $k = 2, 3, \ldots, N - 2$ form $N - 1$ linear equations involving the coefficients $m_1, m_2, \ldots, m_{N-1}$.

Regardless of the particular strategy chosen in Table 5.8, we can rewrite equations 1 and $N - 1$ in (12) and obtain a tridiagonal linear system of the form $\boldsymbol{HM} = \boldsymbol{V}$, which involves $m_1, m_2, \ldots, m_{N-1}$:

$$\begin{bmatrix} b_1 & c_1 & & & & \\ a_1 & b_2 & c_2 & & & \\ & & \ddots & & & \\ & & & a_{N-3} & b_{N-2} & c_{N-2} \\ & & & & a_{N-2} & b_{N-1} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{N-2} \\ m_{N-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N-2} \\ v_{N-1} \end{bmatrix}. \tag{15}$$

The linear system in (15) is strictly diagonally dominant and has a unique solution (see Chapter 3 for details). After the coefficients $\{m_k\}$ are determined, the spline coefficients $\{s_{k,j}\}$ for $S_k(x)$ are computed using the formulas

$$\begin{aligned} s_{k,0} &= y_k, & s_{k,1} &= d_k - \frac{h_k(2m_k + m_{k+1})}{6}, \\ s_{k,2} &= \frac{m_k}{2}, & s_{k,3} &= \frac{m_{k+1} - m_k}{6h_k}. \end{aligned} \tag{16}$$

Each cubic polynomial $S_k(x)$ can be written in nested multiplication form for efficient computation:

$$S_k(x) = ((s_{k,3}w + s_{k,2})w + s_{k,1})w + y_k, \qquad \text{where } w = x - x_k \tag{17}$$

and $S_k(x)$ is used on the interval $x_k \le x \le x_{k+1}$.

Equations (12) together with a strategy from Table 5.8 can be used to construct a cubic spline with distinctive properties at the endpoints. Specifically, the values for m_0 and m_N in Table 5.8 are used to customize the first and last equations in (12) and form the system of $N - 1$ equations given in (15). Then the tridiagonal system is solved for the remaining coefficients $m_1, m_2, \ldots, m_{N-1}$. Finally, the formulas in (16) are used to determine the spline coefficients. For reference, we now state how the equations must be prepared for each different type of spline.

Endpoint Constraints

The following five lemmas show the form of the tridiagonal linear system that must be solved for each of the different endpoint constraints in Table 5.8.

Lemma 5.1 (Clamped Spline). There exists a unique cubic spline with the first derivative boundary conditions $S'(a) = d_0$ and $S'(b) = d_N$.

Proof. Solve the linear system

$$\begin{aligned} \left(\frac{3}{2}h_0 + 2h_1\right)m_1 + h_1 m_2 &= u_1 - 3(d_0 - S'(x_0)) \\ h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} &= u_k \qquad \text{for } k = 2,\ 3,\ \ldots,\ N-2 \\ h_{N-2}m_{N-2} + \left(2h_{N-2} + \frac{3}{2}h_{N-1}\right)m_{N-1} &= u_{N-1} - 3(S'(x_N) - d_{N-1}). \end{aligned}$$

•

Remark. The clamped spline involves slope at the ends. This spline can be visualized as the curve obtained when a flexible elastic rod is forced to pass through the data points, and the rod is clamped at each end with a fixed slope. This spline would be useful to a draftsman for drawing a smooth curve through several points.

Lemma 5.2 (Natural Spline). There exists a unique cubic spline with the free boundary conditions $S''(a) = 0$ and $S''(b) = 0$.

Proof. Solve the linear system

$$\begin{aligned} 2(h_0 + h_1)m_1 + h_1 m_2 &= u_1 \\ h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} &= u_k \qquad \text{for } k = 2,\ 3,\ \ldots,\ N-2. \\ h_{N-2}m_{N-2} + 2(h_{N-2} + h_{N-1})m_{N-1} &= u_{N-1}. \end{aligned}$$ •

Remark. The natural spline is the curve obtained by forcing a flexible elastic rod through the data points but letting the slope at the ends be free to equilibrate to the position that minimizes the oscillatory behavior of the curve. It is useful for fitting a curve to experimental data that are significant to several significant digits.

Lemma 5.3 (Extrapolated Spline). There exists a unique cubic spline that uses extrapolation from the interior nodes at x_1 and x_2 to determine $S''(a)$ and extrapolation from the nodes at x_{N-1} and x_{N-2} to determine $S''(b)$.

Proof. Solve the linear system

$$\begin{aligned} &\left(3h_0 + 2h_1 + \frac{h_0^2}{h_1}\right)m_1 + \left(h_1 - \frac{h_0^2}{h_1}\right)m_2 = u_1 \\ &h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = u_k \qquad \text{for } k = 2,\ 3,\ \ldots,\ N-2 \\ &\left(h_{N-2} - \frac{h_{N-1}^2}{h_{N-2}}\right)m_{N-2} + \left(2h_{N-2} + 3h_{N-1} + \frac{h_{N-1}^2}{h_{N-2}}\right)m_{N-1} = u_{N-1}. \end{aligned}$$ •

Remark. The extrapolated spline is equivalent to assuming that the end cubic is an extension of the adjacent cubic; that is, the spline forms a single cubic curve over the interval $[x_0, x_2]$ and another single cubic over the interval $[x_{N-2}, x_N]$.

Lemma 5.4 (Parabolically Terminated Spline). There exists a unique cubic spline that uses $S'''(x) \equiv 0$ on the interval $[x_0, x_1]$ and $S'''(x) \equiv 0$ on $[x_{N-1}, x_N]$.

Proof. Solve the linear system

$$\begin{aligned} (3h_0 + 2h_1)m_1 + h_1 m_2 &= u_1 \\ h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} &= u_k \qquad \text{for } k = 2,\ 3,\ \ldots,\ N-2 \\ h_{N-2}m_{N-2} + (2h_{N-2} + 3h_{N-1})m_{N-1} &= u_{N-1}. \end{aligned}$$ •

Remark. The assumption that $S''(x) \equiv 0$ on the interval $[x_0, x_1]$ forces the cubic to degenerate to a quadratic over $[x_0, x_1]$, and a similar situation occurs over $[x_{N-1}, x_N]$.

Lemma 5.5 (Endpoint Curvature-Adjusted Spline). There exists a unique cubic spline with the second derivative boundary conditions $S''(a)$ and $S''(b)$ specified.

Proof. Solve the linear system

$$\begin{aligned} 2(h_0 + h_1)m_1 + h_1 m_2 &= u_1 - h_0 S''(x_0) \\ h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} &= u_k \qquad \text{for } k = 2,\ 3,\ \ldots,\ N-2 \\ h_{N-2}m_{N-2} + 2(h_{N-2} + h_{N-1})m_{N-1} &= u_{N-1} - h_{N-1}S''(x_N). \end{aligned}$$

•

Remark. Imposing values for $S''(a)$ and $S''(b)$ permits the practitioner to adjust the curvature at each endpoint.

The next five examples illustrate the behavior of the various splines. It is possible to mix the end conditions to obtain an even wider variety of possibilities, but we leave these variations to the reader to investigate.

Example 5.7. Find the clamped cubic spline that passes through (0, 0), (1, 0.5), (2, 2.0), and (3, 1.5) with the first derivative boundary conditions $S'(0) = 0.2$ and $S'(3) = -1$.

First, compute the quantities

$$\begin{aligned} &h_0 = h_1 = h_2 = 1 \\ &d_0 = (y_1 - y_0)/h_0 = (0.5 - 0.0)/1 = 0.5 \\ &d_1 = (y_2 - y_1)/h_1 = (2.0 - 0.5)/1 = 1.5 \\ &d_2 = (y_3 - y_2)/h_2 = (1.5 - 2.0)/1 = -0.5 \\ &u_1 = 6(d_1 - d_0) = 6(1.5 - 0.5) = 6.0 \\ &u_2 = 6(d_2 - d_1) = 6(-0.5 - 1.5) = -12.0. \end{aligned}$$

Then use Lemma 5.1 and obtain the equations

$$\begin{aligned} \left(\frac{3}{2} + 2\right) m_1 + m_2 &= 6.0 - 3(0.5 - 0.2) = 5.1, \\ m_1 + \left(2 + \frac{3}{2}\right) m_2 &= -12.0 - 3(-1.0 - (-0.5)) = -10.5. \end{aligned}$$

When these equations are simplified and put in matrix notation, we have

$$\begin{bmatrix} 3.5 & 1.0 \\ 1.0 & 3.5 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 5.1 \\ -10.5 \end{bmatrix}.$$

It is a straightforward task to compute the solution $m_1 = 2.25$ and $m_2 = -3.72$. Now apply the equations in (i) of Table 5.8 to determine the coefficients m_0 and m_3:

$$\begin{aligned} m_0 &= 3(0.5 - 0.2) - \frac{2.52}{2} = -0.36, \\ m_3 &= 3(-1.0 + 0.5) - \frac{-3.72}{2} = 0.36. \end{aligned}$$

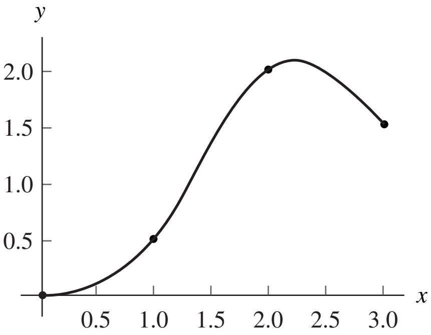


Figure 5.12 The clamped cubic spline with derivative boundary conditions: $S'(0) = 0.2$ and $S'(3) = -1$.

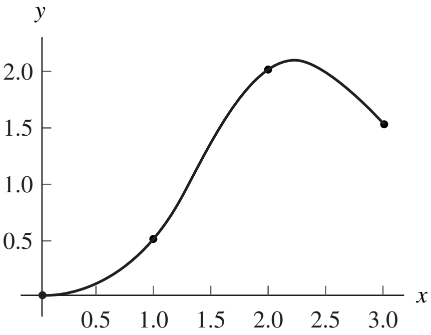


Figure 5.13 The natural cubic spline with $S''(0) = 0$ and $S''(3) = 0$.

Next, the values $m_0 = -0.36$, $m_1 = 2.25$, $m_2 = -3.72$, and $m_3 = 0.36$ are substituted into equations (16) to find the spline coefficients. The solution is

$$\begin{aligned}
S_0(x) &= 0.48x^3 - 0.18x^2 + 0.2x && \text{for } 0 \le x \le 1,\\
S_1(x) &= -1.04(x-1)^3 + 1.26(x-1)^2 &&\\
&\quad + 1.28(x-1) + 0.5 && \text{for } 1 \le x \le 2,\\
S_2(x) &= 0.68(x-2)^3 - 1.86(x-2)^2 &&\\
&\quad + 0.68(x-2) + 2.0 && \text{for } 2 \le x \le 3.
\end{aligned} \tag{18}$$

This clamped cubic spline is shown in Figure 5.12. ■

Example 5.8. Find the natural cubic spline that passes through (0, 0.0), (1, 0.5), (2, 2.0), and (3, 1.5) with the free boundary conditions $S''(x) = 0$ and $S''(3) = 0$.

Use the same values $\{h_k\}$, $\{d_k\}$, and $\{u_k\}$ that were computed in Example 5.7. Then use Lemma 5.2 and obtain the equations

$$\begin{aligned}
2(1+1)m_1 + m_2 &= 6.0,\\
m_1 + 2(1+1)m_2 &= -12.0.
\end{aligned}$$

The matrix form of this linear system is

$$\begin{bmatrix} 4.0 & 1.0 \\ 1.0 & 4.0 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 6.0 \\ -12.0 \end{bmatrix}.$$

It is easy to find the solution $m_1 = 2.4$ and $m_2 = -3.6$. Since $m_0 = S''(0) = 0$ and

$m_3 = S''(3) = 0$, when equations (16) are used to find the spline coefficients, the result is

$$\begin{aligned} S_0(x) &= 0.4x^3 + 0.1x && \text{for } 0 \le x \le 1, \\ S_1(x) &= -(x-1)^3 + 1.2(x-1)^2 \\ &\qquad + 1.3(x-1) + 0.5 && \text{for } 1 \le x \le 2, \\ S_2(x) &= 0.6(x-2)^3 - 1.8(x-2)^2 \\ &\qquad + 0.7(x-2) + 2.0 && \text{for } 2 \le x \le 3. \end{aligned} \tag{19}$$

This natural cubic spline is shown in Figure 5.13. ■

Example 5.9. Find the extrapolated cubic spline through (0, 0.0), (1, 0.5), (2, 2.0), and (3, 1.5).

Use the values $\{h_k\}$, $\{d_k\}$, and $\{u_k\}$ from Example 5.7 with Lemma 5.3 and obtain the linear system

$$\begin{aligned} (3+2+1)m_1 + (1-1)m_2 &= 6.0, \\ (1-1)m_1 + (2+3+1)m_2 &= -12.0. \end{aligned}$$

The matrix form is

$$\begin{bmatrix} 6.0 & 0.0 \\ 0.0 & 6.0 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 6.0 \\ -12.0 \end{bmatrix},$$

and it is trivial to obtain $m_1 = 1.0$ and $m_2 = -2.0$. Now apply the equations in (iii) of Table 5.8 to compute m_0 and m_3:

$$\begin{aligned} m_0 &= 1.0 - (-2.0 - 1.0) = 4.0, \\ m_3 &= -2.0 + (-2.0 - 1.0) = -5.0. \end{aligned}$$

Finally, the values for $\{m_k\}$ are substituted in equations (16) to find the spline coefficients. The solution is

$$\begin{aligned} S_0(x) &= -0.5x^3 + 2.0x^2 - x && \text{for } 0 \le x \le 1, \\ S_1(x) &= -0.5(x-1)^3 + 0.5(x-1)^2 \\ &\qquad + 1.5(x-1) + 0.5 && \text{for } 1 \le x \le 2, \\ S_2(x) &= -0.5(x-2)^3 - (x-2)^2 \\ &\qquad + (x-2) + 2.0 && \text{for } 2 \le x \le 3. \end{aligned} \tag{20}$$

The extrapolated cubic spline is shown in Figure 5.14. ■

Example 5.10. Find the parabolically terminated cubic spline through (0, 0.0), (1, 0.5), (2, 2.0), and (3, 1.5).

Use $\{h_k\}$, $\{d_k\}$, and $\{u_k\}$ from Example 5.7 and then apply Lemma 5.4 to obtain

$$\begin{aligned} (3+2)m_1 + m_2 &= 6.0, \\ m_1 + (2+3)m_2 &= -12.0. \end{aligned}$$

3. Determine which of the following functions are cubic splines. *Hint.* Which, if any, of the five parts of Definition 5.1 does a given function $f(x)$ not satisfy?

(a) $$f(x) = \begin{cases} \frac{19}{2} - \frac{81}{4}x + 15x^2 - \frac{13}{4}x^3 & \text{for } 1 \le x \le 2 \\ \frac{-77}{2} + \frac{207}{4}x - 21x^2 + \frac{11}{4}x^3 & \text{for } 2 \le x \le 3 \end{cases}$$

(b) $$f(x) = \begin{cases} 11 - 24x + 18x^2 - 4x^3 & \text{for } 1 \le x \le 2 \\ -54 + 72x - 30x^2 + 4x^3 & \text{for } 2 \le x \le 3 \end{cases}$$

(c) $$f(x) = \begin{cases} 18 - \frac{75}{2}x + 26x^2 - \frac{11}{2}x^3 & \text{for } 1 \le x \le 2 \\ -70 + \frac{189}{2}x - 40x^2 + \frac{11}{2}x^3 & \text{for } 2 \le x \le 3 \end{cases}$$

(d) $$f(x) = \begin{cases} 13 - 31x + 23x^2 - 5x^3 & \text{for } 1 \le x \le 2 \\ -35 + 51x - 22x^2 + 3x^3 & \text{for } 2 \le x \le 3 \end{cases}$$

4. Find the clamped cubic spline that passes through the points $(-3, 2)$, $(-2, 0)$, $(1, 3)$, and $(4, 1)$ with the first derivative boundary conditions $S'(-3) = -1$ and $S'(4) = 1$.

5. Find the natural cubic spline that passes through the points $(-3, 2)$, $(-2, 0)$, $(1, 3)$, and $(4, 1)$ with the free boundary conditions $S''(-3) = 0$ and $S''(4) = 0$.

6. Find the extrapolated cubic spline that passes through the points $(-3, 2)$, $(-2, 0)$, $(1, 3)$, and $(4, 1)$.

7. Find the parabolically terminated cubic spline that passes through the points $(-3, 2)$, $(-2, 0)$, $(1, 3)$, and $(4, 1)$.

8. Find the curvature-adjusted cubic spline that passes through the points $(-3, 2)$, $(-2, 0)$, $(1, 3)$, and $(4, 1)$ with the second derivative boundary conditions $S''(-3) = -1$ and $S''(4) = 2$.

9. (a) Find the clamped cubic spline that passes through the points $\{(x_k, f(x_k))\}_{k=0}^{3}$, on the graph of $f(x) = x + 2/x$, using the nodes $x_0 = 1/2, x_1 = 1, x_2 = 3/2$, and $x_3 = 2$. Use the first derivative boundary conditions $S'(x_0) = f'(x_0)$ and $S'(x_3) = f'(x_3)$. Graph f and the clamped cubic spline interpolant on the same coordinate system.

(b) Find the natural cubic spline that passes through the points $\{(x_k, f(x_k))\}_{k=0}^{3}$, on the graph of $f(x) = x + 2/x$, using the nodes $x_0 = 1/2, x_1 = 1, x_2 = 3/2$, and $x_3 = 2$. Use the free boundary conditions $S''(x_0) = 0$ and $S''(x_3) = 0$. Graph f and the natural cubic spline interpolant on the same coordinate system.

10. (a) Find the clamped cubic spline that passes through the points $\{(x_k, f(x_k))\}_{k=0}^{3}$, on the graph of $f(x) = \cos(x^2)$, using the nodes $x_0 = 0, x_1 = \sqrt{\pi/2}, x_2 = \sqrt{3\pi/2}$, and $x_3 = \sqrt{5\pi/2}$. Use the first derivative boundary conditions $S'(x_0) = f'(x_0)$ and $S'(x_3) = f'(x_3)$. Graph f and the clamped cubic spline interpolant on the same coordinate system.

(b) Find the natural cubic spline that passes through the points $\{(x_k, f(x_k))\}_{k=0}^{3}$, on the graph of $f(x) = \cos(x^2)$, using the nodes $x_0 = 0, x_1 = \sqrt{\pi/2}, x_2 = \sqrt{3\pi/2}$, and $x_3 = \sqrt{5\pi/2}$. Use the free boundary conditions $S''(x_0) = 0$ and

In Example 6.1 the mathematical value for the limit is $f'(1) \approx 2.718281828$. Observe that the value $h_5 = 10^{-5}$ gives the best approximation, $D_5 = 2.7183$.

Example 6.1 shows that it is not easy to find numerically the limit in equation (2). The sequence starts to converge to e, and D_5 is the closest; then the terms move away from e. In Program 6.1 it is suggested that terms in the sequence $\{D_k\}$ should be computed until $|D_{N+1} - D_N| \geq |D_N - D_{N-1}|$. This is an attempt to determine the best approximation before the terms start to move away from the limit. When this criterion is applied to Example 6.1, we have $0.0007 = |D_6 - D_5| > |D_5 - D_4| = 0.00012$; hence D_5 is the answer we choose. We now proceed to develop formulas that give a reasonable amount of accuracy for larger values of h.

Central-Difference Formulas

If the function $f(x)$ can be evaluated at values that lie to the left and right of x, then the best two-point formula will involve abscissas that are chosen symmetrically on both sides of x.

Theorem 6.1 (Centered Formula of Order $\boldsymbol{O(h^2)}$). Assume that $f \in C^3[a, b]$ and that $x - h, x, x + h \in [a, b]$. Then

$$(3) \qquad f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

Furthermore, there exists a number $c = c(x) \in [a, b]$ such that

$$(4) \qquad f'(x) = \frac{f(x+h) - f(x-h)}{2h} + E_{\text{trunc}}(f, h),$$

where

$$E_{\text{trunc}}(f, h) = -\frac{h^2 f^{(3)}(c)}{6} = \boldsymbol{O}(h^2).$$

The term $E(f, h)$ is called the ***truncation error***.

Proof. Start with the second-degree Taylor expansions $f(x) = P_2(x) + E_2(x)$, about x, for $f(x+h)$ and $f(x-h)$:

$$(5) \qquad f(x+h) = f(x) + f'(x)h + \frac{f^{(2)}(x)h^2}{2!} + \frac{f^{(3)}(c_1)h^3}{3!}$$

and

$$(6) \qquad f(x-h) = f(x) - f'(x)h + \frac{f^{(2)}(x)h^2}{2!} - \frac{f^{(3)}(c_2)h^3}{3!}.$$

After (6) is subtracted from (5), the result is

$$(7) \qquad f(x+h) - f(x-h) = 2f'(x)h + \frac{((f^{(3)}(c_1) + f^{(3)}(c_2))h^3}{3!}.$$

Since $f^{(3)}(x)$ is continuous, the intermediate value theorem can be used to find a value c so that

$$\frac{f^{(3)}(c_1) + f^{(3)}(c_2)}{2} = f^{(3)}(c). \tag{8}$$

This can be substituted into (7) and the terms rearranged to yield

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f^{(3)}(c)h^2}{3!}. \tag{9}$$

The first term on the right side of (9) is the central-difference formula (3), the second term is the truncation error, and the proof is complete. •

Suppose that the value of the third derivative $f^{(3)}(c)$ does not change too rapidly; then the truncation error in (4) goes to zero in the same manner as h^2, which is expressed by using the notation $\boldsymbol{O}(h^2)$. When computer calculations are used, it is not desirable to choose h too small. For this reason it is useful to have a formula for approximating $f'(x)$ that has a truncation error term of the order $\boldsymbol{O}(h^4)$.

Theorem 6.2 (Centered Formula of Order $\boldsymbol{O(h^4)}$). Assume that $f \in C^5[a, b]$ and that $x - 2h, x - h, x, x + h, x + 2h \in [a, b]$. Then

$$f'(x) \approx \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h}. \tag{10}$$

Furthermore, there exists a number $c = c(x) \in [a, b]$ such that

$$f'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} + E_{\text{trunc}}(f, h), \tag{11}$$

where

$$E_{\text{trunc}}(f, h) = \frac{h^4 f^{(5)}(c)}{30} = \boldsymbol{O}(h^4).$$

Proof. One way to derive formula (10) is as follows. Start with the difference between the fourth-degree Taylor expansions $f(x) = P_4(x) + E_4(x)$, about x, of $f(x+h)$ and $f(x-h)$:

$$f(x+h) - f(x-h) = 2f'(x)h + \frac{2f^{(3)}(x)h^3}{3!} + \frac{2f^{(5)}(c_1)h^5}{5!}. \tag{12}$$

Then use the step size $2h$, instead of h, and write down the following approximation:

$$f(x+2h) - f(x-2h) = 4f'(x)h + \frac{16f^{(3)}(x)h^3}{3!} + \frac{64f^{(5)}(c_2)h^5}{5!}. \tag{13}$$

Next multiply the terms in equation (12) by 8 and subtract (13) from it. The terms involving $f^{(3)}(x)$ will be eliminated and we get

$$(14)\quad \begin{aligned} &-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)\\ &\qquad = 12f'(x)h + \frac{(16f^{(5)}(c_1)-64f^{(5)}(c_2))h^5}{120}. \end{aligned}$$

If $f^{(5)}(x)$ has one sign and if its magnitude does not change rapidly, we can find a value c that lies in $[x-2h, x+2h]$ so that

$$(15)\qquad 16f^{(5)}(c_1)-64f^{(5)}(c_2) = -48f^{(5)}(c).$$

After (15) is substituted into (14) and the result is solved for $f'(x)$, we obtain

$$(16)\quad f'(x) = \frac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)}{12h} + \frac{f^{(5)}(c)h^4}{30}.$$

The first term on the right side of (16) is the central-difference formula (10) and the second term is the truncation error; the theorem is proved. •

Suppose that $|f^{(5)}(c)|$ is bounded for $c \in [a, b]$; then the truncation error in (11) goes to zero in the same manner as h^4, which is expressed with the notation $\boldsymbol{O}(h^4)$. Now we can make a comparison of the two formulas (3) and (10). Suppose that $f(x)$ has five continuous derivatives and that $|f^{(3)}(c)|$ and $|f^{(5)}(c)|$ are about the same. Then the truncation error for the fourth-order formula (10) is $\boldsymbol{O}(h^4)$ and will go to zero faster than the truncation error $\boldsymbol{O}(h^2)$ for the second-order formula (3). This permits the use of a larger step size.

Example 6.2. Let $f(x) = \cos(x)$.

- **(a)** Use formulas (3) and (10) with step sizes $h = 0.1, 0.01, 0.001$, and 0.0001, and calculate approximations for $f'(0.8)$. Carry nine decimal places in all the calculations.
- **(b)** Compare with the true value $f'(0.8) = -\sin(0.8)$.

(a) Using formula (3) with $h = 0.01$, we get

$$f'(0.8) \approx \frac{f(0.81)-f(0.79)}{0.02} \approx \frac{0.689498433-0.703845316}{0.02} \approx -0.717344150.$$

Using formula (10) with $h = 0.01$, we get

$$\begin{aligned} f'(0.8) &\approx \frac{-f(0.82)+8f(0.81)-8f(0.79)+f(0.78)}{0.12}\\ &\approx \frac{-0.682221207+8(0.689498433)-8(0.703845316)+0.710913538}{0.12}\\ &\approx -0.717356108. \end{aligned}$$

(b) The error in approximation for formulas (3) and (10) turns out to be -0.000011941 and 0.000000017, respectively. In this example, formula (10) gives a better approximation to $f'(0.8)$ than formula (3) when $h = 0.01$. The error analysis will illuminate this example and show why this happened . The other calculations are summarized in Table 6.2. ■

Table 6.2 Numerical Differentiation Using Formulas (3) and (10)

Step size	Approximation by formula (3)	Error using formula (3)	Approximation by formula (10)	Error using formula (10)
0.1	−0.716161095	−0.001194996	−0.717353703	−0.000002389
0.01	−0.717344150	−0.000011941	−0.717356108	0.000000017
0.001	−0.717356000	−0.000000091	−0.717356167	0.000000076
0.0001	−0.717360000	−0.000003909	−0.717360833	0.000004742

Error Analysis and Optimum Step Size

An important topic in the study of numerical differentiation is the effect of the computer's round-off error. Let us examine the formulas more closely. Assume that a computer is used to make numerical computations and that

$$f(x_0 - h) = y_{-1} + e_{-1} \quad \text{and} \quad f(x_0 + h) = y_1 + e_1,$$

where $f(x_0 - h)$ and $f(x_0 + h)$ are approximated by the numerical values y_{-1} and y_1, and e_{-1} and e_1 are the associated round-off errors, respectively. The following result indicates the complex nature of error analysis for numerical differentiation.

Corollary 6.1(a). Assume that f satisfies the hypotheses of Theorem 6.1 and use the ***computational formula***

$$f'(x_0) \approx \frac{y_1 - y_{-1}}{2h}. \tag{17}$$

The error analysis is explained by the following equations:

$$f'(x_0) = \frac{y_1 - y_{-1}}{2h} + E(f, h), \tag{18}$$

where

$$\begin{aligned} E(f, h) &= E_{\text{round}}(f, h) + E_{\text{trunc}}(f, h) \\ &= \frac{e_1 - e_{-1}}{2h} - \frac{h^2 f^{(3)}(c)}{6}, \end{aligned} \tag{19}$$

where the ***total error term*** $E(f, h)$ has a part due to round-off error plus a part due to truncation error.

Corollary 6.1(b). Assume that f satisfies the hypotheses of Theorem 6.1 and that numerical computations are made. If $|e_{-1}| \le \epsilon$, $|e_1| \le \epsilon$, and $M = \max_{a \le x \le b}\{|f^{(3)}(x)|\}$, then

$$|E(f, h)| \le \frac{\epsilon}{h} + \frac{Mh^2}{6}, \tag{20}$$

and the value of h that minimizes the right-hand side of (20) is

$$h = \left(\frac{3\epsilon}{M}\right)^{1/3}. \tag{21}$$

When h is small, the portion of (19) involving $(e_1 - e_{-1})/2h$ can be relatively large. In Example 6.2, when $h = 0.0001$, this difficulty was encountered. The round-off errors are

$$\begin{aligned} f(0.8001) &= 0.696634970 + e_1 && \text{where } e_1 \approx -0.0000000003 \\ f(0.7999) &= 0.696778442 + e_{-1} && \text{where } e_{-1} \approx 0.0000000005. \end{aligned}$$

The truncation error term is

$$\frac{-h^2 f^{(3)}(c)}{6} \approx -(0.0001)^2\left(\frac{\sin(0.8)}{6}\right) \approx 0.000000001.$$

The error term $E(f, h)$ in (19) can now be estimated:

$$\begin{aligned} E(f,h) &\approx \frac{-0.0000000003 - 0.0000000005}{0.0002} - 0.000000001 \\ &= -0.000004001. \end{aligned}$$

Indeed, the computed numerical approximation for the derivative using $h = 0.0001$ is found by the calculation

$$\begin{aligned} f'(0.8) &\approx \frac{f(0.8001) - f(0.7999)}{0.0002} = \frac{0.696634970 - 0.696778442}{0.0002} \\ &= -0.717360000, \end{aligned}$$

and a loss of about four significant digits is evident. The error is -0.000003909 and this is close to the predicted error, -0.000004001.

When formula (21) is applied to Example 6.2, we can use the bound $|f^{(3)}(x)| \le |\sin(x)| \le 1 = M$ and the value $\epsilon = 0.5 \times 10^{-9}$ for the magnitude of the round-off error. The optimal value for h is easily calculated: $h = (1.5 \times 10^{-9}/1)^{1/3} = 0.001144714$. The step size $h = 0.001$ was closest to the optimal value 0.001144714 and it gave the best approximation to $f'(0.8)$ among the four choices involving formula (3) (see Table 6.2 and Figure 6.3).

An error analysis of formula (10) is similar. Assume that a computer is used to make numerical computations and that $f(x_0 + kh) = y_k + e_k$.

Corollary 6.2(a). Assume that f satisfies the hypotheses of Theorem 6.2 and use the ***computational formula***

$$f'(x_0) \approx \frac{-y_2 + 8y_1 - 8y_{-1} + y_{-2}}{12h}. \tag{22}$$

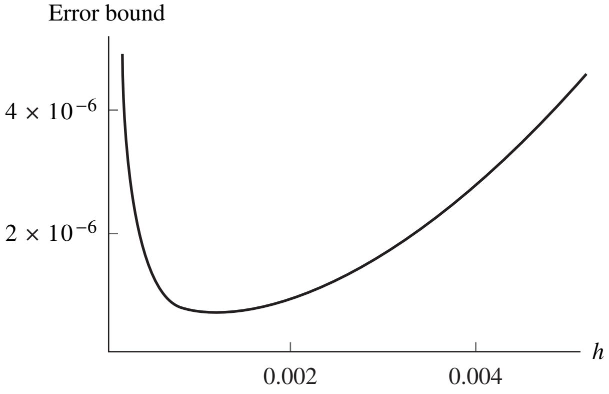


Figure 6.3 Finding the optimal step size $h = 0.001144714$ when formula (21) is applied to $f(x) = \cos(x)$ in Example 6.2.

The error analysis is explained by the following equations:

$$f'(x_0) = \frac{-y_2 + 8y_1 - 8y_{-1} + y_{-2}}{12h} + E(f, h), \tag{23}$$

where

$$\begin{aligned} E(f, h) &= E_{\text{round}}(f, h) + E_{\text{trunc}}(f, h) \\ &= \frac{-e_2 + 8e_1 - 8e_{-1} + e_{-2}}{12h} + \frac{h^4 f^{(5)}(c)}{30}, \end{aligned} \tag{24}$$

where the total error term $E(f, h)$ has a part due to round-off error plus a part due to truncation error.

Corollary 6.2(b). Assume that f satisfies the hypotheses of Theorem 6.2 and that numerical computations are made. If $|e_k| \le \epsilon$ and $M = \max_{a \le x \le b}\{|f^{(5)}(x)|\}$, then

$$|E(f, h)| \le \frac{3\epsilon}{2h} + \frac{Mh^4}{30}, \tag{25}$$

and the value of h that minimizes the right-hand side of (25) is

$$h = \left(\frac{45\epsilon}{4M}\right)^{1/5}. \tag{26}$$

When formula (25) is applied to Example 6.2, we can use the bound $|f^{(5)}(x)| \le |\sin(x)| \le 1 = M$ and the value $\epsilon = 0.5 \times 10^{-9}$ for the magnitude of the round-off error. The optimal value for h is easily calculated: $h = (22.5 \times 10^{-9}/4)^{1/5} = 0.022388475$. The step size $h = 0.01$ was closest to the optimal value 0.022388475, and it gave the best approximation to $f'(0.8)$ among the four choices involving formula (10) (see Table 6.2 and Figure 6.4).

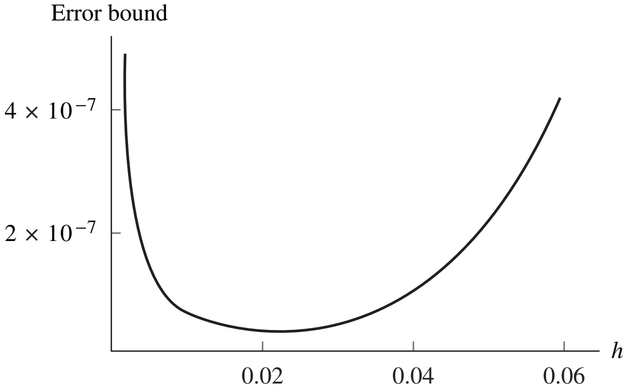


Figure 6.4 Finding the optimal step size $h = 0.022388475$ when formula (26) is applied to $f(x) = \cos(x)$ in Example 6.2.

We should not end the discussion of Example 6.2 without mentioning that numerical differentiation formulas can be obtained by an alternative derivation. They can be derived by differentiation of an interpolation polynomial. For example, the Lagrange form of the quadratic polynomial $p_2(x)$ that passes through the three points $(0.7, \cos(0.7))$, $(0.8, \cos(0.8))$, and $(0.9, \cos(0.9))$ is

$$\begin{aligned} p_2(x) = {}& 38.2421094(x - 0.8)(x - 0.9) - 69.6706709(x - 0.7)(x - 0.9) \\ & + 31.0804984(x - 0.7)(x - 0.8). \end{aligned}$$

This polynomial can be expanded to obtain the usual form:

$$p_2(x) = 1.046875165 - 0.159260044x - 0.348063157x^2.$$

A similar computation can be used to obtain the quartic polynomial $p_4(x)$ that passes through the points $(0.6, \cos(0.6))$, $(0.7, \cos(0.7))$, $(0.8, \cos(0.8))$, $(0.9, \cos(0.9))$, and $(1.0, \cos(1.0))$:

$$\begin{aligned} p_4(x) = {}& 0.998452927 + 0.009638391x - 0.523291341x^2 \\ & + 0.026521229x^3 + 0.028981100x^4. \end{aligned}$$

When these polynomials are differentiated, they produce $p_2'(0.8) = -0.716161095$ and $p_4'(0.8) = -0.717353703$, which agree with the values listed under $h = 0.1$ in Table 6.2. The graphs of $p_2(x)$ and $p_4(x)$ and their tangent lines at $(0.8, \cos(0.8))$ are shown in Figure 6.5(a) and (b), respectively.

```
%       - relerr is the relative error bound
%       - n is the coordinate of the ''best approximation''
err=1;
relerr=1;
h=1;
j=1;
D(1,1)=(feval(f,x+h)-feval(f,x-h))/(2*h);
while relerr>toler & err>delta &j<12
   h=h/2;
   D(j+1,1)=(feval(f,x+h)-feval(f,x-h))/(2*h);
   for k=1:j
      D(j+1,k+1)=D(j+1,k)+(D(j+1,k)-D(j,k))/((4^k)-1);
   end
   err=abs(D(j+1,j+1)-D(j,j));
   relerr=2*err/(abs(D(j+1,j+1))+abs(D(j,j))+eps);
   j=j+1;
end
[n,n]=size(D);
```

Exercises for Approximating the Derivative

1. Let $f(x) = \sin(x)$, where x is measured in radians.

(a) Calculate approximations to $f'(0.8)$ using formula (3) with $h = 0.1$, $h = 0.01$, and $h = 0.001$. Carry eight or nine decimal places.

(b) Compare with the value $f'(0.8) = \cos(0.8)$.

(c) Compute bounds for the truncation error (4). Use

$$|f^{(3)}(c)| \leq \cos(0.7) \approx 0.764842187$$

for all cases.

2. Let $f(x) = e^x$.

(a) Calculate approximations to $f'(2.3)$ using formula (3) with $h = 0.1$, $h = 0.01$, and $h = 0.001$. Carry eight or nine decimal places.

(b) Compare with the value $f'(2.3) = e^{2.3}$.

(c) Compute bounds for the truncation error (4). Use

$$|f^{(3)}(c)| \leq e^{2.4} \approx 11.02317638$$

for all cases.

3. Let $f(x) = \sin(x)$, where x is measured in radians.

(a) Calculate approximations to $f'(0.8)$ using formula (10) with $h = 0.1$ and $h = 0.01$, and compare with $f'(0.8) = \cos(0.8)$.

(b) Use the extrapolation formula in (29) to compute the approximations to $f'(0.8)$ in part (a).

(c) Compute bounds for the truncation error (11). Use

$$|f^{(5)}(c)| \le \cos(0.6) \approx 0.825335615$$

for both cases.

4. Let $f(x) = e^x$.

(a) Calculate approximations to $f'(2.3)$ using formula (10) with $h = 0.1$ and $h = 0.01$, and compare with $f'(2.3) = e^{2.3}$.

(b) Use the extrapolation formula in (29) to compute the approximations to $f'(2.3)$ in part (a).

(c) Compute bounds for the truncation error (11). Use

$$|f^{(5)}(c)| \le e^{2.5} \approx 12.18249396$$

for both cases.

5. Compare the numerical differentiation formulas (3) and (10). Let $f(x) = x^3$ and find approximations for $f'(2)$.

(a) Use formula (3) with $h = 0.05$.

(b) Use formula (10) with $h = 0.05$.

(c) Compute bounds for the truncation errors (4) and (11).

6. **(a)** Use Taylor's theorem to show that

$$f(x+h) = f(x) + hf'(x) + \frac{h^2 f^{(2)}(c)}{2}, \quad \text{where } |c - x| < h.$$

(b) Use part (a) to show that the difference quotient in equation (2) has error of order $\boldsymbol{O}(h) = -hf^{(2)}(c)/2$.

(c) Why is formula (3) better to use than formula (2)?

7. *Partial differentiation formulas.* The partial derivative $f_x(x, y)$ of $f(x, y)$ with respect to x is obtained by holding y fixed and differentiating with respect to x. Similarly, $f_y(x, y)$ is found by holding x fixed and differentiating with respect to y. Formula (3) can be adapted to partial derivatives

$$\begin{aligned} f_x(x, y) &= \frac{f(x+h, y) - f(x-h, y)}{2h} + \boldsymbol{O}(h^2), \\ f_y(x, y) &= \frac{f(x, y+h) - f(x, y-h)}{2h} + \boldsymbol{O}(h^2). \end{aligned} \tag{1}$$

(a) Let $f(x, y) = xy/(x + y)$. Calculate approximations to $f_x(2, 3)$ and $f_y(2, 3)$ using the formulas in (1) with $h = 0.1$, 0.01, and 0.001. Compare with the values obtained by differentiating $f(x, y)$ partially.

6.2 Numerical Differentiation Formulas

More Central-Difference Formulas

The formulas for $f'(x_0)$ in the preceding section required that the function can be computed at abscissas that lie on both sides of x, and they were referred to as central-difference formulas. Taylor series can be used to obtain central-difference formulas for the higher derivatives. The popular choices are those of order $\boldsymbol{O}(h^2)$ and $\boldsymbol{O}(h^4)$ and are given in Tables 6.3 and 6.4. In these tables we use the convention that $f_k = f(x_0+kh)$ for $k = -3, -2, -1, 0, 1, 2, 3$.

For illustration, we will derive the formula for $f''(x)$ of order $\boldsymbol{O}(h^2)$ in Table 6.3. Start with the Taylor expansions

$$(1) \qquad f(x+h) = f(x) + hf'(x) + \frac{h^2 f''(x)}{2} + \frac{h^3 f^{(3)}(x)}{6} + \frac{h^4 f^{(4)}(x)}{24} + \cdots$$

Table 6.3 Central-Difference Formulas of Order $\boldsymbol{O}(h^2)$

$f'(x_0) \approx \dfrac{f_1 - f_{-1}}{2h}$
$f''(x_0) \approx \dfrac{f_1 - 2f_0 + f_{-1}}{h^2}$
$f^{(3)}(x_0) \approx \dfrac{f_2 - 2f_1 + 2f_{-1} - f_{-2}}{2h^3}$
$f^{(4)}(x_0) \approx \dfrac{f_2 - 4f_1 + 6f_0 - 4f_{-1} + f_{-2}}{h^4}$

Table 6.4 Central-Difference Formulas of Order $\boldsymbol{O}(h^4)$

$f'(x_0) \approx \dfrac{-f_2 + 8f_1 - 8f_{-1} + f_{-2}}{12h}$
$f''(x_0) \approx \dfrac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2}$
$f^{(3)}(x_0) \approx \dfrac{-f_3 + 8f_2 - 13f_1 + 13f_{-1} - 8f_{-2} + f_{-3}}{8h^3}$
$f^{(4)}(x_0) \approx \dfrac{-f_3 + 12f_2 - 39f_1 + 56f_0 - 39f_{-1} + 12f_{-2} - f_{-3}}{6h^4}$

and

$$(2) \qquad f(x-h) = f(x) - hf'(x) + \frac{h^2 f''(x)}{2} - \frac{h^3 f^{(3)}(x)}{6} + \frac{h^4 f^{(4)}(x)}{24} - \cdots .$$

Adding equations (1) and (2) will eliminate the terms involving the odd derivatives $f'(x), f^{(3)}(x), f^{(5)}(x), \ldots$:

$$(3) \qquad f(x+h) + f(x-h) = 2f(x) + \frac{2h^2 f''(x)}{2} + \frac{2h^4 f^{(4)}(x)}{24} + \cdots .$$

Solving equation (3) for $f''(x)$ yields

$$(4) \qquad \begin{aligned} f''(x) = {} & \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{2h^2 f^{(4)}(x)}{4!} \\ & - \frac{2h^4 f^{(6)}(x)}{6!} - \cdots - \frac{2h^{2k-2} f^{(2k)}(x)}{(2k)!} - \cdots . \end{aligned}$$

If the series in (4) is truncated at the fourth derivative, there exists a value c that lies in $[x-h, x+h]$, so that

$$(5) \qquad f''(x_0) = \frac{f_1 - 2f_0 + f_{-1}}{h^2} - \frac{h^2 f^{(4)}(c)}{12}.$$

This gives us the desired formula for approximating $f''(x)$:

$$(6) \qquad f''(x_0) \approx \frac{f_1 - 2f_0 + f_{-1}}{h^2}.$$

Example 6.4. Let $f(x) = \cos(x)$.

- **(a)** Use formula (6) with $h = 0.1$, 0.01, and 0.001 and find approximations to $f''(0.8)$. Carry nine decimal places in all calculations.
- **(b)** Compare with the true value $f''(0.8) = -\cos(0.8)$.

(a) The calculation for $h = 0.01$ is

$$\begin{aligned} f''(0.8) &\approx \frac{f(0.81) - 2f(0.80) + f(0.79)}{0.0001} \\ &\approx \frac{0.689498433 - 2(0.696706709) + 0.703845316}{0.0001} \\ &\approx -0.696690000. \end{aligned}$$

(b) The error in this approximation is -0.000016709. The other calculations are summarized in Table 6.5. The error analysis will illuminate this example and show why $h = 0.01$ was best. ■

Table 6.5 Numerical Approximations to $f''(x)$ for Example 6.4

Step size	Approximation by formula (6)	Error using formula (6)
$h = 0.1$	-0.696126300	-0.000580409
$h = 0.01$	-0.696690000	-0.000016709
$h = 0.001$	-0.696000000	-0.000706709

Error Analysis

Let $f_k = y_k + e_k$, where e_k is the error in computing $f(x_k)$, including noise in measurement and round-off error. Then formula (6) can be written

$$f''(x_0) = \frac{y_1 - 2y_0 + y_{-1}}{h^2} + E(f, h). \tag{7}$$

The error term $E(h, f)$ for the numerical derivative (7) will have a part due to round-off error and a part due to truncation error:

$$E(f, h) = \frac{e_1 - 2e_0 + e_{-1}}{h^2} - \frac{h^2 f^{(4)}(c)}{12}. \tag{8}$$

If it is assumed that each error e_k is of the magnitude ϵ, with signs that accumulate errors, and that $|f^{(4)}(x)| \le M$, then we get the following error bound:

$$|E(f, h)| \le \frac{4\epsilon}{h^2} + \frac{Mh^2}{12}. \tag{9}$$

If h is small, then the contribution $4\epsilon/h^2$ due to round-off error is large. When h is large, the contribution $Mh^2/12$ is large. The optimal step size will minimize the quantity

$$g(h) = \frac{4\epsilon}{h^2} + \frac{Mh^2}{12}. \tag{10}$$

Setting $g'(h) = 0$ results in $-8\epsilon/h^3 + Mh/6 = 0$, which yields the equation $h^4 = 48\epsilon/M$, from which we obtain the optimal value:

$$h = \left(\frac{48\epsilon}{M}\right)^{1/4}. \tag{11}$$

When formula (11) is applied to Example 6.4, use the bound $|f^{(4)}(x)| \le |\cos(x)| \le 1 = M$ and the value $\epsilon = 0.5 \times 10^{-9}$. The optimal step size is $h = (24 \times 10^{-9}/1)^{1/4} = 0.01244666$, and we see that $h = 0.01$ was closest to the optimal value.

Since the portion of the error due to round off is inversely proportional to the square of h, this term grows when h gets small. This is sometimes referred to as the ***step-size dilemma***. One partial solution to this problem is to use a formula of higher order so that a larger value of h will produce the desired accuracy. The formula for $f''(x_0)$ of order $\boldsymbol{O}(h^4)$ in Table 6.4 is

$$f''(x_0) = \frac{-f_2 + 16f_1 - 30f_0 + 16f_{-1} - f_{-2}}{12h^2} + E(f,h). \tag{12}$$

The error term for (12) has the form

$$E(f,h) = \frac{16\epsilon}{3h^2} + \frac{h^4 f^{(6)}(c)}{90}, \tag{13}$$

where c lies in the interval $[x - 2h, x + 2h]$. A bound for $|E(f,h)|$ is

$$|E(f,h)| \le \frac{16\epsilon}{3h^2} + \frac{h^4 M}{90}, \tag{14}$$

where $|f^{(6)}(x)| \le M$. The optimal value for h is given by the formula

$$h = \left(\frac{240\epsilon}{M}\right)^{1/6}. \tag{15}$$

Example 6.5. Let $f(x) = \cos(x)$.

(a) Use formula (12) with $h = 1.0, 0.1$, and 0.01 and find approximations to $f''(0.8)$. Carry nine decimal places in all the calculations.

(b) Compare with the true value $f''(0.8) = -\cos(0.8)$.

(c) Determine the optimal step size.

(a) The calculation for $h = 0.1$ is

$$\begin{aligned} &f''(0.8) \\ &\quad\approx \frac{-f(1.0) + 16f(0.9) - 30f(0.8) + 16f(0.7) - f(0.6)}{0.12} \\ &\quad\approx \frac{-0.540302306 + 9.945759488 - 20.90120127 + 12.23747499 - 0.825335615}{0.12} \\ &\quad\approx -0.696705958. \end{aligned}$$

(b) The error in this approximation is -0.000000751. The other calculations are summarized in Table 6.6.

(c) When formula (15) is applied, we can use the bound $|f^{(6)}(x)| \le |\cos(x)| \le 1 = M$ and the value $\epsilon = 0.5 \times 10^{-9}$. These values give the optimal step size $h = (120 \times 10^{-9}/1)^{1/6} = 0.070231219$. ■

Table 6.6 Numerical Approximations to $f''(x)$ for Example 6.5

Step size	Approximation by formula (12)	Error using formula (12)
$h = 1.0$	-0.689625413	-0.007081296
$h = 0.1$	-0.696705958	-0.000000751
$h = 0.01$	-0.696690000	-0.000016709

Table 6.7 Forward- and Backward-Difference Formulas of Order $\boldsymbol{O}(h^2)$

$$f'(x_0) \approx \frac{-3f_0 + 4f_1 - f_2}{2h} \qquad \left(\begin{array}{l}\text{forward}\\ \text{difference}\end{array}\right)$$

$$f'(x_0) \approx \frac{3f_0 - 4f_{-1} + f_{-2}}{2h} \qquad \left(\begin{array}{l}\text{backward}\\ \text{difference}\end{array}\right)$$

$$f''(x_0) \approx \frac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2} \qquad \left(\begin{array}{l}\text{forward}\\ \text{difference}\end{array}\right)$$

$$f''(x_0) \approx \frac{2f_0 - 5f_{-1} + 4f_{-2} - f_{-3}}{h^2} \qquad \left(\begin{array}{l}\text{backward}\\ \text{difference}\end{array}\right)$$

$$f^{(3)}(x_0) \approx \frac{-5f_0 + 18f_1 - 24f_2 + 14f_3 - 3f_4}{2h^3}$$

$$f^{(3)}(x_0) \approx \frac{5f_0 - 18f_{-1} + 24f_{-2} - 14f_{-3} + 3f_{-4}}{2h^3}$$

$$f^{(4)}(x_0) \approx \frac{3f_0 - 14f_1 + 26f_2 - 24f_3 + 11f_4 - 2f_5}{h^4}$$

$$f^{(4)}(x_0) \approx \frac{3f_0 - 14f_{-1} + 26f_{-2} - 24f_{-3} + 11f_{-4} - 2f_{-5}}{h^4}$$

Generally, if numerical differentiation is performed, only about half the accuracy of which the computer is capable is obtained. This severe loss of significant digits will almost always occur unless we are fortunate to find a step size that is optimal. Hence we must always proceed with caution when numerical differentiation is performed. The difficulties are more pronounced when working with experimental data, where the function values have been rounded to only a few digits. If a numerical derivative must be obtained from data, we should consider curve fitting, by using least-squares techniques, and differentiate the formula for the curve.

Exercises for Numerical Differentiation Formulas

1. Let $f(x) = \ln(x)$ and carry eight or nine decimal places.

(a) Use formula (6) with $h = 0.05$ to approximate $f''(5)$.

(b) Use formula (6) with $h = 0.01$ to approximate $f''(5)$.

(c) Use formula (12) with $h = 0.1$ to approximate $f''(5)$.

(d) Which answer, (a), (b), or (c), is most accurate?

2. Let $f(x) = \cos(x)$ and carry eight or nine decimal places.

(a) Use formula (6) with $h = 0.05$ to approximate $f''(1)$.

(b) Use formula (6) with $h = 0.01$ to approximate $f''(1)$.

(c) Use formula (12) with $h = 0.1$ to approximate $f''(1)$.

(d) Which answer, (a), (b), or (c), is most accurate?

3. Consider the table for $f(x) = \ln(x)$ rounded to four decimal places.

x	$f(x) = \ln(x)$
4.90	1.5892
4.95	1.5994
5.00	1.6094
5.05	1.6194
5.10	1.6292

(a) Use formula (6) with $h = 0.05$ to approximate $f''(5)$.

(b) Use formula (6) with $h = 0.01$ to approximate $f''(5)$.

(c) Use formula (12) with $h = 0.05$ to approximate $f''(5)$.

(d) Which answer, (a), (b), or (c), is most accurate?

4. Consider the table for $f(x) = \cos(x)$ rounded to four decimal places.

x	$f(x) = \cos(x)$
0.90	0.6216
0.95	0.5817
1.00	0.5403
1.05	0.4976
1.10	0.4536

(a) Use formula (6) with $h = 0.05$ to approximate $f''(1)$.

(b) Use formula (6) with $h = 0.01$ to approximate $f''(1)$.

(c) Use formula (12) with $h = 0.05$ to approximate $f''(1)$.

(d) Which answer, (a), (b), or (c), is most accurate?

5. Use the numerical differentiation formula (6) and $h = 0.01$ to approximate $f''(1)$ for the functions

(a) $f(x) = x^2$ **(b)** $f(x) = x^4$

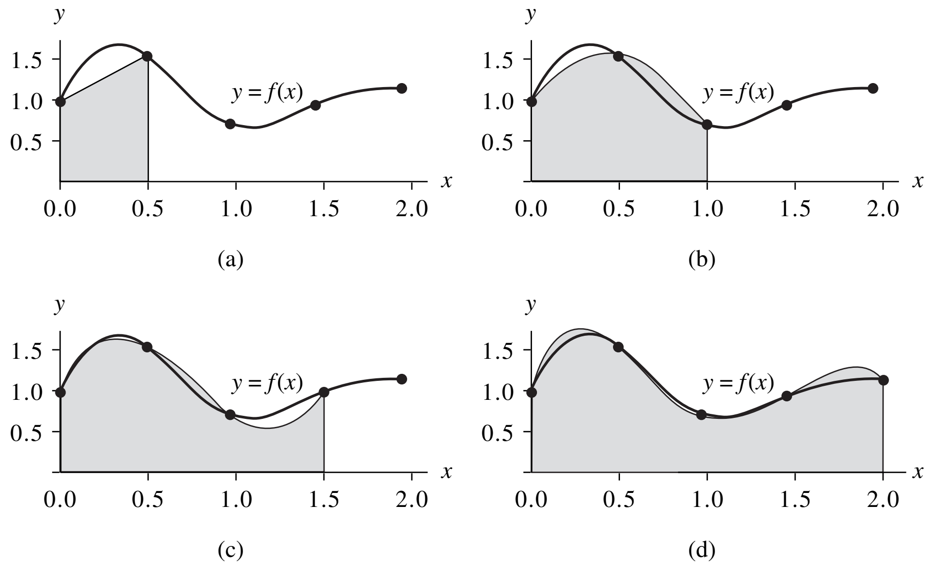


Figure 7.2 (a) The trapezoidal rule integrates $y = P_1(x)$ over $[x_0, x_1] = [0.0, 0.5]$. (b) Simpson's rule integrates $y = P_2(x)$ over $[x_0, x_1] = [0.0, 1.0]$. (c) Simpson's $\frac{3}{8}$ rule integrates $y = P_3(x)$ over $[x_0, x_3] = [0.0, 1.5]$. (d) Boole's rule integrates $y = P_4(x)$ over $[x_0, x_4] = [0.0, 2.0]$.

quadrature formulas are

$$\int_{x_0}^{x_1} f(x)\,dx \approx \frac{h}{2}(f_0 + f_1) \qquad \text{(trapezoidal rule)}, \tag{4}$$

$$\int_{x_0}^{x_2} f(x)\,dx \approx \frac{h}{3}(f_0 + 4f_1 + f_2) \qquad \text{(Simpson's rule)}, \tag{5}$$

$$\int_{x_0}^{x_3} f(x)\,dx \approx \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) \qquad \text{(Simpson's } \tfrac{3}{8} \text{ rule)}, \tag{6}$$

$$\int_{x_0}^{x_4} f(x)\,dx \approx \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) \tag{7}$$

(Boole's rule).

Corollary 7.1 (Newton-Cotes Precision). Assume that $f(x)$ is sufficiently differentiable; then $E[f]$ for Newton-Cotes quadrature involves an appropriate higher derivative. The trapezoidal rule has degree of precision $n = 1$. If $f \in C^2[a, b]$, then

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}(f_0 + f_1) - \frac{h^3}{12} f^{(2)}(c). \tag{8}$$

integration over a fixed interval $[a, b]$ using exactly five function evaluations $f_k = f(x_k)$, for $k = 0, 1, \ldots, 4$ for each method. When the trapezoidal rule is applied on the four subintervals $[x_0, x_1]$, $[x_1, x_2]$, $[x_2, x_3]$, and $[x_3, x_4]$, it is called a ***composite trapezoidal rule***:

$$\begin{aligned}\int_{x_0}^{x_4} f(x)\,dx &= \int_{x_0}^{x_1} f(x)\,dx + \int_{x_1}^{x_2} f(x)\,dx + \int_{x_2}^{x_3} f(x)\,dx + \int_{x_3}^{x_4} f(x)\,dx \\ &\approx \frac{h}{2}(f_0 + f_1) + \frac{h}{2}(f_1 + f_2) + \frac{h}{2}(f_2 + f_3) + \frac{h}{2}(f_3 + f_4) \qquad (17) \\ &= \frac{h}{2}(f_0 + 2f_1 + 2f_2 + 2f_3 + f_4).\end{aligned}$$

Simpson's rule can also be used in this manner. When Simpson's rule is applied on the two subintervals $[x_0, x_2]$ and $[x_2, x_4]$, it is called a ***composite Simpson's rule***:

$$\begin{aligned}\int_{x_0}^{x_4} f(x)\,dx &= \int_{x_0}^{x_2} f(x)\,dx + \int_{x_2}^{x_4} f(x)\,dx \\ &\approx \frac{h}{3}(f_0 + 4f_1 + f_2) + \frac{h}{3}(f_2 + 4f_3 + f_4) \qquad (18) \\ &= \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + f_4).\end{aligned}$$

The next example compares the values obtained with (17), (18), and (7).

Example 7.3. Consider the integration of the function $f(x) = 1 + e^{-x}\sin(4x)$ over $[a, b] = [0, 1]$. Use exactly five function evaluations and compare the results from the composite trapezoidal rule, composite Simpson rule, and Boole's rule.

The uniform step size is $h = 1/4$. The composite trapezoidal rule (17) produces

$$\begin{aligned}\int_0^1 f(x)\,dx &\approx \frac{1/4}{2}(f(0) + 2f(\tfrac{1}{4}) + 2f(\tfrac{1}{2}) + 2f(\tfrac{3}{4}) + f(1)) \\ &= \frac{1}{8}(1.00000 + 2(1.65534) + 2(1.55152) + 2(1.06666) + 0.72159) \\ &= 1.28358.\end{aligned}$$

Using the composite Simpson's rule (18), we get

$$\begin{aligned}\int_0^1 f(x)\,dx &\approx \frac{1/4}{3}(f(0) + 4f(\tfrac{1}{4}) + 2f(\tfrac{1}{2}) + 4f(\tfrac{3}{4}) + f(1)) \\ &= \frac{1}{12}(1.00000 + 4(1.65534) + 2(1.55152) + 4(1.06666) + 0.72159) \\ &= 1.30938.\end{aligned}$$

We have already seen the result of Boole's rule in Example 7.2:

$$\begin{aligned}\int_0^1 f(x)\,dx &\approx \frac{2(1/4)}{45}(7f(0) + 32f(\tfrac{1}{4}) + 12f(\tfrac{1}{2}) + 32f(\tfrac{3}{4}) + 7f(1)) \\ &= 1.30859.\end{aligned}$$

Exercises for Introduction to Quadrature

1. Consider integration of $f(x)$ over the fixed interval $[a, b] = [0, 1]$. Apply the various quadrature formulas (4) through (7). The step sizes are $h = 1$, $h = \frac{1}{2}$, $h = \frac{1}{3}$, and $h = \frac{1}{4}$ for the trapezoidal rule, Simpson's rule, Simpson's $\frac{3}{8}$ rule, and Boole's rule, respectively.

(a) $f(x) = \sin(\pi x)$

(b) $f(x) = 1 + e^{-x}\cos(4x)$

(c) $f(x) = \sin(\sqrt{x})$

Remark. The true values of the definite integrals are (a) $2/\pi = 0.636619772367\ldots$, (b) $(18e - \cos(4) + 4\sin(4))/(17e) = 1.007459631397\ldots$, and (c) $2(\sin(1) - \cos(1)) = 0.602337357879\ldots$. Graphs of the functions are shown in Figure 7.5(a) through (c), respectively.

2. Consider integration of $f(x)$ over the fixed interval $[a, b] = [0, 1]$. Apply the various quadrature formulas: the composite trapezoidal rule (17), the composite Simpson rule (18), and Boole's rule (7). Use five function evaluations at equally spaced nodes. The uniform step size is $h = \frac{1}{4}$.

(a) $f(x) = \sin(\pi x)$

(b) $f(x) = 1 + e^{-x}\cos(4x)$

(c) $f(x) = \sin(\sqrt{x})$

3. Consider a general interval $[a, b]$. Show that Simpson's rule produces exact results for the functions $f(x) = x^2$ and $f(x) = x^3$; that is,

(a) $\int_a^b x^2\,dx = \dfrac{b^3}{3} - \dfrac{a^3}{3}$ **(b)** $\int_a^b x^3\,dx = \dfrac{b^4}{4} - \dfrac{a^4}{4}$

4. Integrate the Lagrange interpolation polynomial

$$P_1(x) = f_0\frac{x - x_1}{x_0 - x_1} + f_1\frac{x - x_0}{x_1 - x_0}$$

over the interval $[x_0, x_1]$ and establish the trapezoidal rule.

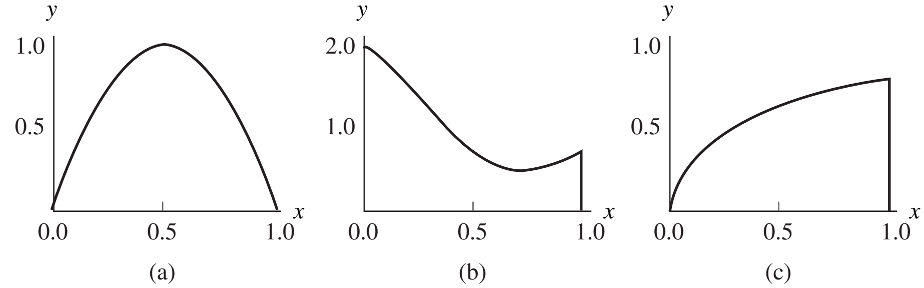


Figure 7.5 (a) $y = \sin(\pi x)$, (b) $y = 1 + e^{-x}\cos(4x)$, (c) $y = \sin(\sqrt{x})$.

can be used to estimate the number of subintervals required to achieve a specified accuracy.

Corollary 7.2 (Trapezoidal Rule: Error Analysis). Suppose that $[a, b]$ is subdivided into M subintervals $[x_k, x_{k+1}]$ of width $h = (b - a)/M$. The composite trapezoidal rule

$$T(f, h) = \frac{h}{2}(f(a) + f(b)) + h \sum_{k=1}^{M-1} f(x_k) \tag{7}$$

is an approximation to the integral

$$\int_a^b f(x)\,dx = T(f, h) + E_T(f, h). \tag{8}$$

Furthermore, if $f \in C^2[a, b]$, there exists a value c with $a < c < b$ so that the error term $E_T(f, h)$ has the form

$$E_T(f, h) = \frac{-(b - a) f^{(2)}(c) h^2}{12} = \boldsymbol{O}(h^2). \tag{9}$$

Proof. We first determine the error term when the rule is applied over $[x_0, x_1]$. Integrating the Lagrange polynomial $P_1(x)$ and its remainder yields

$$\int_{x_0}^{x_1} f(x)\,dx = \int_{x_0}^{x_1} P_1(x)\,dx + \int_{x_0}^{x_1} \frac{(x - x_0)(x - x_1) f^{(2)}(c(x))}{2!}\,dx. \tag{10}$$

The term $(x - x_0)(x - x_1)$ does not change sign on $[x_0, x_1]$, and $f^{(2)}(c(x))$ is continuous. Hence the second mean value theorem for integrals implies that there exists a value c_1 so that

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}(f_0 + f_1) + f^{(2)}(c_1) \int_{x_0}^{x_1} \frac{(x - x_0)(x - x_1)}{2!}\,dx. \tag{11}$$

Use the change of variable $x = x_0 + ht$ in the integral on the right side of (11):

$$\begin{aligned}
\int_{x_0}^{x_1} f(x)\,dx &= \frac{h}{2}(f_0 + f_1) + \frac{f^{(2)}(c_1)}{2} \int_0^1 h(t - 0)h(t - 1)h\,dt \\
&= \frac{h}{2}(f_0 + f_1) + \frac{f^{(2)}(c_1) h^3}{2} \int_0^1 (t^2 - t)\,dt \\
&= \frac{h}{2}(f_0 + f_1) - \frac{f^{(2)}(c_1) h^3}{12}.
\end{aligned} \tag{12}$$

Now we are ready to add up the error terms for all of the intervals $[x_k, x_{k+1}]$:

$$(13)\qquad \begin{aligned}\int_a^b f(x)\,dx &= \sum_{k=1}^{M} \int_{x_{k-1}}^{x_k} f(x)\,dx \\ &= \sum_{k=1}^{M} \frac{h}{2}(f(x_{k-1}) + f(x_k)) - \frac{h^3}{12} \sum_{k=1}^{M} f^{(2)}(c_k).\end{aligned}$$

The first sum is the composite trapezoidal rule $T(f, h)$. In the second term, one factor of h is replaced with its equivalent $h = (b - a)/M$, and the result is

$$\int_a^b f(x)\,dx = T(f, h) - \frac{(b-a)h^2}{12}\left(\frac{1}{M}\sum_{k=1}^{M} f^{(2)}(c_k)\right).$$

The term in parentheses can be recognized as an average of values for the second derivative and hence is replaced by $f^{(2)}(c)$. Therefore, we have established that

$$\int_a^b f(x)\,dx = T(f, h) - \frac{(b-a)f^{(2)}(c)h^2}{12},$$

and the proof of Corollary 7.2 is complete. •

Corollary 7.3 (Simpson's Rule: Error Analysis). Suppose that $[a, b]$ is subdivided into $2M$ subintervals $[x_k, x_{k+1}]$ of equal width $h = (b - a)/(2M)$. The composite Simpson rule

$$(14)\qquad S(f, h) = \frac{h}{3}(f(a) + f(b)) + \frac{2h}{3}\sum_{k=1}^{M-1} f(x_{2k}) + \frac{4h}{3}\sum_{k=1}^{M} f(x_{2k-1})$$

is an approximation to the integral

$$(15)\qquad \int_a^b f(x)\,dx = S(f, h) + E_S(f, h).$$

Furthermore, if $f \in C^4[a, b]$, there exists a value c with $a < c < b$ so that the error term $E_S(f, h)$ has the form

$$(16)\qquad E_S(f, h) = \frac{-(b-a)f^{(4)}(c)h^4}{180} = \boldsymbol{O}(h^4).$$

Example 7.7. Consider $f(x) = 2 + \sin(2\sqrt{x})$. Investigate the error when the composite trapezoidal rule is used over $[1, 6]$ and the number of subintervals is 10, 20, 40, 80, and 160.

Table 7.2 Composite Trapezoidal Rule for $f(x) = 2 + \sin(2\sqrt{x})$ over [1, 6]

M	h	$T(f, h)$	$E_T(f, h) = \boldsymbol{O}(h^2)$
10	0.5	8.19385457	−0.01037540
20	0.25	8.18604926	−0.00257006
40	0.125	8.18412019	−0.00064098
80	0.0625	8.18363936	−0.00016015
160	0.03125	8.18351924	−0.00004003

Table 7.2 shows the approximations $T(f, h)$. The antiderivative of $f(x)$ is

$$F(x) = 2x - \sqrt{x}\cos(2\sqrt{x}) + \frac{\sin(2\sqrt{x})}{2},$$

and the true value of the definite integral is

$$\int_1^6 f(x)\,dx = F(x)\Big|_{x=1}^{x=6} = 8.1834792077.$$

This value was used to compute the values $E_T(f, h) = 8.1834792077 - T(f, h)$ in Table 7.2. It is important to observe that when h is reduced by a factor of $\frac{1}{2}$ the successive errors $E_T(f, h)$ are diminished by approximately $\frac{1}{4}$. This confirms that the order is $\boldsymbol{O}(h^2)$. ∎

Example 7.8. Consider $f(x) = 2 + \sin(2\sqrt{x})$. Investigate the error when the composite Simpson rule is used over [1, 6] and the number of subintervals is 10, 20, 40, 80, and 160.

Table 7.3 shows the approximations $S(f, h)$. The true value of the integral is 8.1834792077, which was used to compute the values $E_S(f, h) = 8.1834792077 - S(f, h)$ in Table 7.3. It is important to observe that when h is reduced by a factor of $\frac{1}{2}$, the successive errors $E_S(f, h)$ are diminished by approximately $\frac{1}{16}$. This confirms that the order is $\boldsymbol{O}(h^4)$. ∎

Example 7.9. Find the number M and the step size h so that the error $E_T(f, h)$ for the composite trapezoidal rule is less than 5×10^{-9} for the approximation $\int_2^7 dx/x \approx T(f, h)$.

The integrand is $f(x) = 1/x$ and its first two derivatives are $f'(x) = -1/x^2$ and $f^{(2)}(x) = 2/x^3$. The maximum value of $|f^{(2)}(x)|$ taken over [2, 7] occurs at the endpoint $x = 2$, and thus we have the bound $|f^{(2)}(c)| \le |f^{(2)}(2)| = \frac{1}{4}$, for $2 \le c \le 7$. This is used with formula (9) to obtain

$$(17) \qquad |E_T(f, h)| = \frac{|-(b-a)f^{(2)}(c)h^2|}{12} \le \frac{(7-2)\frac{1}{4}h^2}{12} = \frac{5h^2}{48}.$$

Table 7.3 Composite Simpson Rule for $f(x) = 2 + \sin(2\sqrt{x})$ over $[1, 6]$

M	h	$S(f, h)$	$E_S(f, h) = \boldsymbol{O}(h^4)$
5	0.5	8.18301549	0.00046371
10	0.25	8.18344750	0.00003171
20	0.125	8.18347717	0.00000204
40	0.0625	8.18347908	0.00000013
80	0.03125	8.18347920	0.00000001

The step size h and number M satisfy the relation $h = 5/M$, and this is used in (17) to get the relation

$$|E_T(f, h)| \le \frac{125}{48M^2} \le 5 \times 10^{-9}. \tag{18}$$

Now rewrite (18) so that it is easier to solve for M:

$$\frac{25}{48} \times 10^9 \le M^2. \tag{19}$$

Solving (19), we find that $22821.77 \le M$. Since M must be an integer, we choose $M = 22{,}822$, and the corresponding step size is $h = 5/22{,}822 = 0.000219086846$. When the composite trapezoidal rule is implemented with this many function evaluations, there is a possibility that the rounded-off function evaluations will produce a significant amount of error. When the computation was performed, the result was

$$T\left(f, \frac{5}{22{,}822}\right) = 1.252762969,$$

which compares favorably with the true value $\int_2^7 dx/x = \ln(x)|_{x=2}^{x=7} = 1.252762968$. The error is smaller than predicted because the bound $\frac{1}{4}$ for $|f^{(2)}(c)|$ was used. Experimentation shows that it takes about 10,001 function evaluations to achieve the desired accuracy of 5×10^{-9}, and when the calculation is performed with $M = 10{,}000$, the result is

$$T\left(f, \frac{5}{10{,}000}\right) = 1.252762973. \qquad \blacksquare$$

The composite trapezoidal rule usually requires a large number of function evaluations to achieve an accurate answer. This is contrasted in the next example with Simpson's rule, which will require significantly fewer evaluations.

Example 7.10. Find the number M and the step size h so that the error $E_S(f, h)$ for the composite Simpson rule is less than 5×10^{-9} for the approximation $\int_2^7 dx/x \approx S(f, h)$.

The integrand is $f(x) = 1/x$, and $f^{(4)}(x) = 24/x^5$. The maximum value of $|f^{(4)}(c)|$ taken over [2, 7] occurs at the endpoint $x = 2$, and thus we have the bound

$$|f^{(4)}(c)| \leq |f^{(4)}(2)| = \frac{3}{4}$$

for $2 \leq c \leq 7$. This is used with formula (16) to obtain

$$(20) \qquad |E_S(f, h)| = \frac{|-(b-a)f^{(4)}(c)h^4|}{180} \leq \frac{(7-2)\frac{3}{4}h^4}{180} = \frac{h^4}{48}.$$

The step size h and number M satisfy the relation $h = 5/(2M)$, and this is used in (20) to get the relation

$$(21) \qquad |E_S(f, h)| \leq \frac{625}{768M^4} \leq 5 \times 10^{-9}.$$

Now rewrite (21) so that it is easier to solve for M:

$$(22) \qquad \frac{125}{768} \times 10^9 \leq M^4.$$

Solving (22), we find that $112.95 \leq M$. Since M must be an integer, we chose $M = 113$, and the corresponding step size is $h = 5/226 = 0.02212389381$. When the composite Simpson rule was performed, the result was

$$S\left(f, \frac{5}{226}\right) = 1.252762969,$$

which agrees with $\int_2^7 dx/x = \ln(x)|_{x=2}^{x=7} = 1.252762968$. Experimentation shows that it takes about 129 function evaluations to achieve the desired accuracy of 5×10^{-9}, and when the calculation is performed with $M = 64$, the result is

$$S\left(f, \frac{5}{128}\right) = 1.252762973. \qquad \blacksquare$$

So we see that the composite Simpson rule using 229 evaluations of $f(x)$ and the composite trapezoidal rule using 22,823 evaluations of $f(x)$ achieve the same accuracy. In Example 7.10, Simpson's rule required about $\frac{1}{100}$ the number of function evaluations.

```
   x=a+h*2*k;
   s2=s2+feval(f,x);
end
s=h*(feval(f,a)+feval(f,b)+4*s1+2*s2)/3;
```

Exercises for Composite Trapezoidal and Simpson's Rule

1. **(i)** Approximate each integral using the composite trapezoidal rule with $M = 10$.

(ii) Approximate each integral using the composite Simpson rule with $M = 5$.

(a) $\int_{-1}^{1}(1+x^2)^{-1}\,dx$ **(b)** $\int_0^1 (2+\sin(2\sqrt{x}))\,dx$ **(c)** $\int_{0.25}^{4} dx/\sqrt{x}$

(d) $\int_0^4 x^2e^{-x}\,dx$ **(e)** $\int_0^2 2x\cos(x)\,dx$ **(f)** $\int_0^{\pi}\sin(2x)e^{-x}\,dx$

2. *Length of a curve.* The arc length of the curve $y = f(x)$ over the interval $a \le x \le b$ is

$$\text{length} = \int_a^b \sqrt{1 + (f'(x)^2)}\,dx.$$

(i) Approximate the arc length of each function using the composite trapezoidal rule with $M = 10$.

(ii) Approximate the arc length of each function using the composite Simpson rule with $M = 5$.

(a) $f(x) = x^3$ for $0 \le x \le 1$

(b) $f(x) = \sin(x)$ for $0 \le x \le \pi/4$

(c) $f(x) = e^{-x}$ for $0 \le x \le 1$

3. *Surface area.* The solid of revolution obtained by rotating the region under the curve $y = f(x)$, where $a \le x \le b$, about the x-axis has surface area given by

$$\text{area} = 2\pi\int_a^b f(x)\sqrt{1 + (f'(x))^2}\,dx.$$

(i) Approximate the surface area using the composite trapezoidal rule with $M = 10$.

(ii) Approximate the surface area using the composite Simpson rule with $M = 5$.

(a) $f(x) = x^3$ for $0 \le x \le 1$

(b) $f(x) = \sin(x)$ for $0 \le x \le \pi/4$

(c) $f(x) = e^{-x}$ for $0 \le x \le 1$

4. **(a)** Verify that the trapezoidal rule ($M = 1$, $h = 1$) is exact for polynomials of degree ≤ 1 of the form $f(x) = c_1x + c_0$ over [0, 1].

(b) Use the integrand $f(x) = c_2x^2$ and verify that the error term for the trapezoidal rule ($M = 1$, $h = 1$) over the interval [0, 1] is

$$E_T(f, h) = \frac{-(b-a)f^{(2)}(c)h^2}{12}.$$

5. **(a)** Verify that Simpson's rule ($M = 1$, $h = 1$) is exact for polynomials of degree ≤ 3 of the form $f(x) = c_3x^3 + c_2x^2 + c_1x + c_0$ over [0, 2].

(b) Use the integrand $f(x) = c_4x^4$ and verify that the error term for Simpson's rule ($M = 1$, $h = 1$) over the interval [0, 2] is

$$E_S(f, h) = \frac{-(b-a)f^{(4)}(c)h^4}{180}.$$

6. Derive the trapezoidal rule ($M = 1$, $h = 1$) by using the method of undetermined coefficients.

(a) Find the constants w_0 and w_1 so that $\int_0^1 g(t)\,dt = w_0g(0) + w_1g(1)$ is exact for the two functions $g(t) = 1$ and $g(t) = t$.

(b) Use the relation $f(x_0 + ht) = g(t)$ and the change of variable $x = x_0 + ht$ and $dx = h\,dt$ to translate the trapezoidal rule over [0, 1] to the interval $[x_0, x_1]$.

Hint for part (a). You will get a linear system involving the two unknowns w_0 and w_1.

7. Derive Simpson's rule ($M = 1$, $h = 1$) by using the method of undetermined coefficients.

(a) Find the constants w_0, w_1, and w_2 so that $\int_0^2 g(t)\,dt = w_0g(0) + w_1g(1) + w_2g(2)$ is exact for the three functions $g(t) = 1$, $g(t) = t$, and $g(t) = t^2$.

(b) Use the relation $f(x_0 + ht) = g(t)$ and the change of variable $x = x_0 + ht$ and $dx = h\,dt$ to translate the trapezoidal rule over [0, 2] to the interval $[x_0, x_2]$.

Hint for part (a). You will get a linear system involving the three unknowns w_0, w_1, and w_2.

8. Determine the number M and the interval width h so that the composite trapezoidal rule for M subintervals can be used to compute the given integral with an accuracy of 5×10^{-9}.

(a) $\displaystyle\int_{-\pi/6}^{\pi/6} \cos(x)\,dx$ **(b)** $\displaystyle\int_2^3 \frac{1}{5-x}\,dx$ **(c)** $\displaystyle\int_0^2 xe^{-x}\,dx$

Hint for part (c). $f^{(2)}(x) = (x-2)e^{-x}$.

9. Determine the number M and the interval width h so that the composite Simpson rule for $2M$ subintervals can be used to compute the given integral with an accuracy of 5×10^{-9}.

(a) $\displaystyle\int_{-\pi/6}^{\pi/6} \cos(x)\,dx$ **(b)** $\displaystyle\int_2^3 \frac{1}{5-x}\,dx$ **(c)** $\displaystyle\int_0^2 xe^{-x}\,dx$

Hint for part (c). $f^{(4)}(x) = (x-4)e^{-x}$.

10. Consider the definite integral $\int_{-0.1}^{0.1} \cos(x)\,dx = 2\sin(0.1) = 0.1996668333$. The following table gives approximations using the composite trapezoidal rule. Calculate $E_T(f,h) = 0.199668 - T(f,h)$ and confirm that the order is $\boldsymbol{O}(h^2)$.

M	h	$S(f,h)$	$E_T(f,h) = \boldsymbol{O}(h^2)$
1	0.2	0.1990008	
2	0.1	0.1995004	
4	0.05	0.1996252	
8	0.025	0.1996564	
16	0.0125	0.1996642	

11. Consider the definite integral $\int_{-0.75}^{0.75} \cos(x)\,dx = 2\sin(0.75) = 1.363277520$. The following table gives approximations using the composite Simpson rule. Calculate $E_S(f,h) = 1.3632775 - S(f,h)$ and confirm that the order is $\boldsymbol{O}(h^4)$.

M	h	$S(f,h)$	$E_S(f,h) = \boldsymbol{O}(h^4)$
1	0.75	1.3658444	
2	0.375	1.3634298	
4	0.1875	1.3632869	
8	0.09375	1.3632781	

12. *Midpoint rule.* The midpoint rule on $[x_0, x_1]$ is

$$\int_{x_0}^{x_1} f(x)\,dx = 2hf(x_0+h) + \frac{h^3}{3}f''(c), \qquad \text{where } h = \frac{x_1 - x_0}{2}.$$

(a) Expand $F(x)$, the antiderivative of $f(x)$, in a Taylor series about $x_0 + h$ and establish the midpoint rule on $[x_0, x_1]$.

(b) Use part (a) and show that the composite midpoint rule for approximating the integral of $f(x)$ over $[a, b]$ is

$$M(f,h) = h\sum_{k=1}^{N} f\left(a + \left(k - \frac{1}{2}\right)h\right), \qquad \text{where } h = \frac{b-a}{N}.$$

This is an approximation to the integral of $f(x)$ over $[a, b]$, and we write

$$\int_a^b f(x)\,dx \approx M(f,h).$$

(c) Show that the error term $E_M(f,h)$ for part (b) is

$$E_M(f,h) = \frac{h^3}{3}\sum_{k=1}^{N} f^{(2)}(c_k) = \frac{(b-a)f^{(2)}(c)h^2}{3} = \boldsymbol{O}(h^2).$$

13. Use the midpoint rule with $M = 10$ to approximate the integrals in Exercise 1.

14. Prove Corollary 7.3.

Table 7.5 Romberg Integration Tableau

J	$R(J, 0)$ Trapezoidal rule	$R(J, 1)$ Simpson's rule	$R(J, 2)$ Boole's rule	$R(J, 3)$ Third improvement	$R(J, 4)$ Fourth improvement
0	$R(0, 0)$				
1	$R(1, 0)$	$R(1, 1)$			
2	$R(2, 0)$	$R(2, 1)$	$R(2, 2)$		
3	$R(3, 0)$	$R(3, 1)$	$R(3, 2)$	$R(3, 3)$	
4	$R(4, 0)$	$R(4, 1)$	$R(4, 2)$	$R(4, 3)$	$R(4, 4)$

Table 7.6 Romberg Integration Tableau for Example 7.14

J	$R(J, 0)$ Trapezoidal rule	$R(J, 1)$ Simpson's rule	$R(J, 2)$ Boole's rule	$R(J, 3)$ Third improvement
0	0.785398163397			
1	1.726812656758	2.040617487878		
2	1.960534166564	2.038441336499	2.038296259740	
3	2.018793948078	2.038213875249	2.038198711166	2.038197162776
4	2.033347341805	2.038198473047	2.038197446234	2.038197426156
5	2.036984954990	2.038197492719	2.038197427363	2.038197427064

For computational purposes, the values $R(J, K)$ are arranged in the Romberg integration tableau given in Table 7.5.

Example 7.14. Use Romberg integration to find approximations for the definite integral

$$\int_0^{\pi/2} (x^2 + x + 1)\cos(x)\,dx = -2 + \frac{\pi}{2} + \frac{\pi^2}{4} = 2.038197427067\ldots.$$

The computations are given in Table 7.6. In each column the numbers are converging to the value 2.038197427067 .... The values in the Simpson's rule column converge faster than the values in the trapezoidal rule column. For this example, convergence in columns to the right is faster than the adjacent column to the left.

Convergence of the Romberg values in Table 7.6 is easier to see if we look at the error terms $E(J, K) = -2+\pi/2+\pi^2/4-R(J, K)$. Suppose that the interval width is $h = b-a$ and that the higher derivatives of $f(x)$ are of the same magnitude. The error in column K of the Romberg table diminishes by about a factor of $1/2^{2K+2} = 1/4^{K+1}$ as one progresses down its rows. The errors $E(J, 0)$ diminish by a factor of $1/4$, the errors $E(J, 1)$ diminish by a factor of $1/16$, and so on. This can be observed by inspecting the entries $\{E(J, K)\}$ in Table 7.7. ■

Table 7.7 Romberg Error Tableau for Example 7.14

J	h	$E(J,0)=\boldsymbol{O}(h^2)$	$E(J,1)=\boldsymbol{O}(h^4)$	$E(J,2)=\boldsymbol{O}(h^6)$	$E(J,3)=\boldsymbol{O}(h^8)$
0	$b-a$	−1.252799263670			
1	$\dfrac{b-a}{2}$	−0.311384770309	0.002420060811		
2	$\dfrac{b-a}{4}$	−0.077663260503	0.000243909432	0.000098832673	
3	$\dfrac{b-a}{8}$	−0.019403478989	0.000016448182	0.000001284099	−0.000000264291
4	$\dfrac{b-a}{16}$	−0.004850085262	0.000001045980	0.000000019167	−0.000000000912
5	$\dfrac{b-a}{32}$	−0.001212472077	0.000000065651	0.000000000296	−0.000000000003

Theorem 7.7 (Precision of Romberg Integration). Assume that $f \in C^{2K+2}[a,b]$. Then the truncation error term for the Romberg approximation is given in the formula

$$\begin{aligned}\int_a^b f(x)\,dx &= R(J,K) + b_K h^{2K+2} f^{(2K+2)}(c_{J,K}) \\ &= R(J,K) + \boldsymbol{O}(h^{2K+2}),\end{aligned} \tag{34}$$

where $h=(b-a)/2^J$, b_K is a constant that depends on K, and $c_{J,K} \in [a,b]$.

Example 7.15. Apply Theorem 7.7 and show that

$$\int_0^2 10x^9\,dx = 1024 \equiv R(4,4).$$

The integrand is $f(x)=10x^9$, and $f^{(10)}(x) \equiv 0$. Thus the value $K=4$ will make the error term identically zero. A numerical computation will produce $R(4,4)=1024$. ■

Program 7.3 (Recursive Trapezoidal Rule). To approximate

$$\int_a^b f(x)\,dx \approx \frac{h}{2}\sum_{k=1}^{2^J}(f(x_{k-1})+f(x_k))$$

by using the trapezoidal rule and successively increasing the number of subintervals of $[a,b]$. The Jth iteration samples $f(x)$ at 2^J+1 equally spaced points.

```
function T=rctrap(f,a,b,n)
%Input  - f is the integrand input as a string 'f'
%       - a and b are upper and lower limits of integration
%       - n is the number of times for recursion
```

```
h=b-a;
err=1;
J=0;
R=zeros(4,4);
R(1,1)=h*(feval(f,a)+feval(f,b))/2;
while((err>tol)&(J<n))|(J<4)
   J=J+1;
   h=h/2;
   s=0;
   for p=1:M
      x=a+h*(2*p-1);
      s=s+feval(f,x);
   end
   R(J+1,1)=R(J,1)/2+h*s;
   M=2*M;
   for K=1:J
      R(J+1,K+1)=R(J+1,K)+(R(J+1,K)-R(J,K))/(4^K-1);
   end
   err=abs(R(J,J)-R(J+1,K+1));
end
quad=R(J+1,J+1);
```

Exercises for Recursive Rules and Romberg Integration

1. For each of the following definite integrals, construct (by hand) a Romberg table (Table 7.5) with three rows.

(a) $\displaystyle\int_0^3 \frac{\sin(2x)}{1+x^2}\,dx = 0.4761463020\ldots$

(b) $\displaystyle\int_0^3 \sin(4x)e^{-2x}\,dx = 0.1997146621\ldots$

(c) $\displaystyle\int_{0.04}^1 \frac{1}{\sqrt{x}}\,dx = 1.6$

(d) $\displaystyle\int_0^2 \frac{1}{x^2+\frac{1}{10}}\,dx = 4.4713993943\ldots$

(e) $\displaystyle\int_{1/(2\pi)}^2 \sin\left(\frac{1}{x}\right)dx = 1.1140744942\ldots$

(f) $\displaystyle\int_0^2 \sqrt{4-x^2}\,dx = \pi = 3.1415926535\ldots$

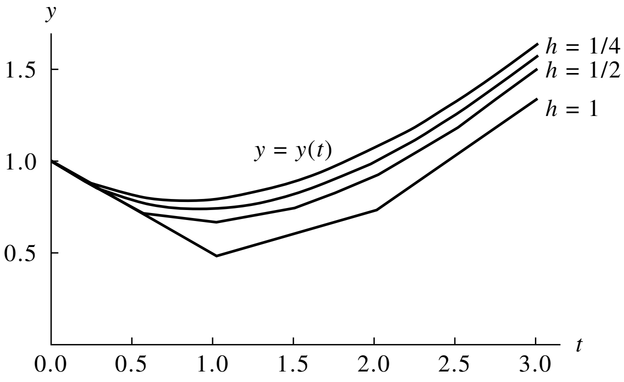


Figure 9.6 Comparison of Euler solutions with different step sizes for $y' = (t - y)/2$ over $[0, 3]$ with the initial condition $y(0) = 1$.

Example 9.5. Compare the F.G.E. when Euler's method is used to solve the I.V.P.

$$y' = \frac{t-y}{2} \qquad \text{over } [0, 3] \text{ with } y(0) = 1,$$

using step sizes $1, \frac{1}{2}, \ldots, \frac{1}{64}$.

Table 9.3 gives the F.G.E. for several step sizes and shows that the error in the approximation to $y(3)$ decreases by about $\frac{1}{2}$ when the step size is reduced by a factor of $\frac{1}{2}$. For the smaller step sizes the conclusion of Theorem 9.3 is easy to see:

$$E(y(3), h) = y(3) - y_M = \boldsymbol{O}(h^1) \approx Ch, \qquad \text{where} \qquad C = 0.256. \qquad \blacksquare$$

Program 9.1 (Euler's Method). To approximate the solution of the initial value problem $y' = f(t, y)$ with $y(a) = y_0$ over $[a, b]$ by computing

$$y_{k+1} = y_k + hf(t_k, y_k) \qquad \text{for } k = 0, 1, \ldots, M - 1.$$

```
function E=euler(f,a,b,ya,M)

%Input  - f is the function entered as a string 'f'
%       - a and b are the left and right endpoints
%       - ya is the initial condition y(a)
%       - M is the number of steps
%Output - E=[T' Y'] where T is the vector of abscissas and
%         Y is the vector of ordinates

h=(b-a)/M;
T=zeros(1,M+1);
Y=zeros(1,M+1);
```

Table 9.2 Comparison of Euler Solutions with Different Step Sizes for $y' = (t - y)/2$ over [0, 3] with $y(0) = 1$

	y_k				
t_k	$h = 1$	$h = \frac{1}{2}$	$h = \frac{1}{4}$	$h = \frac{1}{8}$	$y(t_k)$ Exact
0	1.0	1.0	1.0	1.0	1.0
0.125				0.9375	0.943239
0.25			0.875	0.886719	0.897491
0.375				0.846924	0.862087
0.50		0.75	0.796875	0.817429	0.836402
0.75			0.759766	0.786802	0.811868
1.00	0.5	0.6875	0.758545	0.790158	0.819592
1.50		0.765625	0.846386	0.882855	0.917100
2.00	0.75	0.949219	1.030827	1.068222	1.103638
2.50		1.211914	1.289227	1.325176	1.359514
3.00	1.375	1.533936	1.604252	1.637429	1.669390

Table 9.3 Relation between Step Size and F.G.E. for Euler Solutions to $y' = (t - y)/2$ over [0, 3] with $y(0) = 1$

Step size, h	Number of steps, M	Approximation to $y(3)$, y_M	F.G.E. Error at $t = 3$, $y(3) - y_M$	$\boldsymbol{O}(h) \approx Ch$ where $C = 0.256$
1	3	1.375	0.294390	0.256
$\frac{1}{2}$	6	1.533936	0.135454	0.128
$\frac{1}{4}$	12	1.604252	0.065138	0.064
$\frac{1}{8}$	24	1.637429	0.031961	0.032
$\frac{1}{16}$	48	1.653557	0.015833	0.016
$\frac{1}{32}$	96	1.661510	0.007880	0.008
$\frac{1}{64}$	192	1.665459	0.003931	0.004

```
T=a:h:b;
Y(1)=ya;
for j=1:M
   Y(j+1)=Y(j)+h*feval(f,T(j),Y(j));
end
E=[T' Y'];
```

Exercises for Euler's Method

In Exercises 1 through 5 solve the differential equations by the Euler method.

(a) Let $h = 0.2$ and do two steps by hand calculation. Then let $h = 0.1$ and do four steps by hand calculation.

(b) Compare the exact solution $y(0.4)$ with the two approximations in part (a).

(c) Does the F.G.E. in part (a) behave as expected when h is halved?

1. $y' = t^2 - y$ with $y(0) = 1$, $y(t) = -e^{-t} + t^2 - 2t + 2$

2. $y' = 3y + 3t$ with $y(0) = 1$, $y(t) = \frac{4}{3}e^{3t} - t - \frac{1}{3}$

3. $y' = -ty$ with $y(0) = 1$, $y(t) = e^{-t^2/2}$

4. $y' = e^{-2t} - 2y$ with $y(0) = \frac{1}{10}$, $y(t) = \frac{1}{10}e^{-2t} + te^{-2t}$

5. $y' = 2ty^2$ with $y(0) = 1$, $y(t) = 1/(1 - t^2)$

6. *Logistic population growth.* The population curve $P(t)$ for the United States is assumed to obey the differential equation for a logistic curve $P' = aP - bP^2$. Let t denote the year past 1900, and let the step size be $h = 10$. The values $a = 0.02$ and $b = 0.00004$ produce a model for the population. Using hand calculations, find the Euler approximations to $P(t)$ and fill in the following table. Round off each value P_k to the nearest tenth.

Year	t_k	$P(t_k)$ Actual	P_k Euler approximation
1900	0.0	76.1	76.1
1910	10.0	92.4	89.0
1920	20.0	106.5	______
1930	30.0	123.1	______
1940	40.0	132.6	138.2
1950	50.0	152.3	______
1960	60.0	180.7	______
1970	70.0	204.9	202.8
1980	80.0	226.5	______

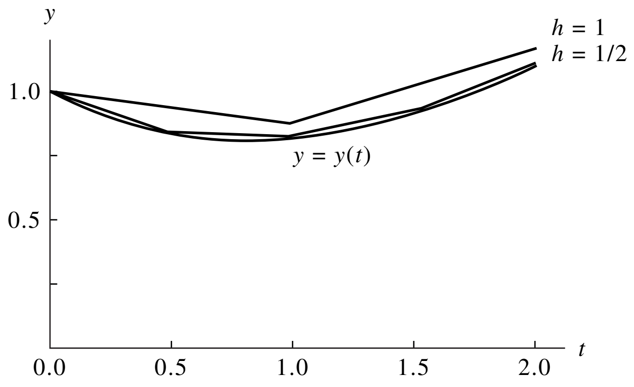


Figure 9.8 Comparison of Heun solutions with different step sizes for $y' = (t - y)/2$ over $[0, 2]$ with the initial condition $y(0) = 1$.

Table 9.4 Comparison of Heun Solutions with Different Step Sizes for $y' = (t - y)/2$ over $[0, 3]$ with $y(0) = 1$

	y_k				
t_k	$h = 1$	$h = \frac{1}{2}$	$h = \frac{1}{4}$	$h = \frac{1}{8}$	$y(t_k)$ Exact
0	1.0	1.0	1.0	1.0	1.0
0.125				0.943359	0.943239
0.25			0.898438	0.897717	0.897491
0.375				0.862406	0.862087
0.50		0.84375	0.838074	0.836801	0.836402
0.75			0.814081	0.812395	0.811868
1.00	0.875	0.831055	0.822196	0.820213	0.819592
1.50		0.930511	0.920143	0.917825	0.917100
2.00	1.171875	1.117587	1.106800	1.104392	1.103638
2.50		1.373115	1.362593	1.360248	1.359514
3.00	1.732422	1.682121	1.672269	1.670076	1.669390

Example 9.7. Compare the F.G.E. when Heun's method is used to solve

$$y' = \frac{t - y}{2} \qquad \text{over } [0, 3] \text{ with } y(0) = 1,$$

using step sizes $1, \frac{1}{2}, \ldots, \frac{1}{64}$.

Table 9.5 Relation between Step Size and F.G.E. for Heun Solutions to $y' = (t - y)/2$ over $[0, 3]$ with $y(0) = 1$

Step size, h	Number of steps, M	Approximation to $y(3)$, y_M	F.G.E. Error at $t = 3$, $y(3) - y_M$	$\boldsymbol{O}(h^2) \approx Ch^2$ where $C = -0.0432$
1	3	1.732422	−0.063032	−0.043200
$\frac{1}{2}$	6	1.682121	−0.012731	−0.010800
$\frac{1}{4}$	12	1.672269	−0.002879	−0.002700
$\frac{1}{8}$	24	1.670076	−0.000686	−0.000675
$\frac{1}{16}$	48	1.669558	−0.000168	−0.000169
$\frac{1}{32}$	96	1.669432	−0.000042	−0.000042
$\frac{1}{64}$	192	1.669401	−0.000011	−0.000011

Table 9.5 gives the F.G.E. and shows that the error in the approximation to $y(3)$ decreases by about $\frac{1}{4}$ when the step size is reduced by a factor of $\frac{1}{2}$:

$$E(y(3), h) = y(3) - y_M = \boldsymbol{O}(h^2) \approx Ch^2, \qquad \text{where } C = -0.0432. \qquad \blacksquare$$

Program 9.2 (Heun’s Method). To approximate the solution of the initial value problem $y' = f(t, y)$ with $y(a) = y_0$ over $[a, b]$ by computing

$$y_{k+1} = y_k + \frac{h}{2}(f(t_k, y_k) + f(t_{k+1}, y_k + hf(t_k, y_k)))$$

for $k = 0, 1, \ldots, M - 1$.

```
function H=heun(f,a,b,ya,M)
%Input  - f is the function entered as a string ’f’
%       - a and b are the left and right endpoints
%       - ya is the initial condition y(a)
%       - M is the number of steps
%Output - H=[T’Y’] where T is the vector of abscissas and
%         Y is the vector of ordinates
h=(b-a)/M;
T=zeros(1,M+1);
Y=zeros(1,M+1);
```

```
T=a:h:b;
Y(1)=ya;
for j=1:M
    k1=feval(f,T(j),Y(j));
    k2=feval(f,T(j+1),Y(j)+h*k1);
    Y(j+1)=Y(j)+(h/2)*(k1+k2);
end
H=[T'Y'];
```

Exercises for Heun's Method

In Exercises 1 through 5, solve the differential equations by Heun's method.

(a) Let $h = 0.2$ and do two steps by hand calculation. Then let $h = 0.1$ and do four steps by hand calculation.

(b) Compare the exact solution $y(0.4)$ with the two approximations in part (a).

(c) Does the F.G.E. in part (a) behave as expected when h is halved?

1. $y' = t^2 - y$ with $y(0) = 1$, $y(t) = -e^{-t} + t^2 - 2t + 2$

2. $y' = 3y + 3t$ with $y(0) = 1$, $y(t) = \frac{4}{3}e^{3t} - t - \frac{1}{3}$

3. $y' = -ty$ with $y(0) = 1$, $y(t) = e^{-t^2/2}$

4. $y' = e^{-2t} - 2y$ with $y(0) = \frac{1}{10}$, $y(t) = \frac{1}{10}e^{-2t} + te^{-2t}$

5. $y' = 2ty^2$ with $y(0) = 1$, $y(t) = 1/(1 - t^2)$
Notice that Heun's method will generate an approximation to $y(1)$ even though the solution curve is not defined at $t = 1$.

6. Show that when Heun's method is used to solve the I.V.P. $y' = f(t)$ over $[a, b]$ with $y(a) = y_0 = 0$ the result is

$$y(b) = \frac{h}{2}\sum_{k=0}^{M-1}(f(t_k) + f(t_{k+1})),$$

which is the trapezoidal rule approximation for the definite integral of $f(t)$ taken over the interval $[a, b]$.

7. The Richardson improvement method discussed in Lemma 7.1 (Section 7.3) can be used in conjunction with Heun's method. If Heun's method is used with step size h, then we have

$$y(b) \approx y_h + Ch^2.$$

If Heun's method is used with step size $2h$, we have

$$y(b) \approx y_{2h} + 4Ch^2.$$

Table 9.11 RK4 Solution to $y' = 1 + y^2$, $y(0) = 0$

k	t_k	RK4 approximation y_k	True solution, $y(t_k) = \tan(t_k)$	Error $y(t_k) - y_k$
0	0.0	0.0000000	0.0000000	0.0000000
1	0.1	0.1003346	0.1003347	0.0000001
2	0.2	0.2027099	0.2027100	0.0000001
3	0.3	0.3093360	0.3093362	0.0000002
4	0.4	0.4227930	0.4227932	0.0000002
5	0.5	0.5463023	0.5463025	0.0000002
6	0.6	0.6841368	0.6841368	0.0000000
7	0.7	0.8422886	0.8422884	−0.0000002
8	0.8	1.0296391	1.0296386	−0.0000005
9	0.9	1.2601588	1.2601582	−0.0000006
10	1.0	1.5574064	1.5574077	0.0000013
11	1.1	1.9647466	1.9647597	0.0000131
12	1.2	2.5720718	2.5721516	0.0000798
13	1.3	3.6015634	3.6021024	0.0005390
14	1.4	5.7919748	5.7978837	0.0059089

Program 9.4 (Runge-Kutta Method of Order 4). To approximate the solution of the initial value problem $y' = f(t, y)$ with $y(a) = y_0$ over $[a, b]$ by using the formula

$$y_{k+1} = y_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

```
function R=rk4(f,a,b,ya,M)

%Input  - f is the function entered as a string 'f'
%       - a and b are the left and right endpoints
%       - ya is the initial condition y(a)
%       - M is the number of steps
%Output - R=[T' Y'] where T is the vector of abscissas
%         and Y is the vector of ordinates

h=(b-a)/M;
T=zeros(1,M+1);
Y=zeros(1,M+1);
T=a:h:b;
Y(1)=ya;
for j=1:M
   k1=h*feval(f,T(j),Y(j));
   k2=h*feval(f,T(j)+h/2,Y(j)+k1/2);
   k3=h*feval(f,T(j)+h/2,Y(j)+k2/2);
   k4=h*feval(f,T(j)+h,Y(j)+k3);
   Y(j+1)=Y(j)+(k1+2*k2+2*k3+k4)/6;
end
R=[T’ Y’];
```

The following program implements the Runge-Kutta-Fehlberg Method (RKF45) described in (28) through (31).

Program 9.5 (Runge-Kutta-Fehlberg Method (RKF45)). To approximate the solution of the initial value problem $y' = f(t, y)$ with $y(a) = y_0$ over $[a, b]$ with an error control and step-size method.

```
function R=rkf45(f,a,b,ya,M,tol)
%Input  - f is the function entered as a string ’f’
%       - a and b are the left and right endpoints
%       - ya is the initial condition y(a)
%       - M is the number of steps
%       - tol is the tolerance
%Output - R=[T’ Y’] where T is the vector of abscissas
%         and Y is the vector of ordinates

%Enter the coefficients necessary to calculate the
%values in (28) and (29)
a2=1/4;b2=1/4;a3=3/8;b3=3/32;c3=9/32;a4=12/13;
b4=1932/2197;c4=-7200/2197;d4=7296/2197;a5=1;
b5=439/216;c5=-8;d5=3680/513;e5=-845/4104;a6=1/2;
b6=-8/27;c6=2;d6=-3544/2565;e6=1859/4104;
f6=-11/40;r1=1/360;r3=-128/4275;r4=-2197/75240;r5=1/50;
r6=2/55;n1=25/216;n3=1408/2565;n4=2197/4104;n5=-1/5;

big=1e15;
h=(b-a)/M;
hmin=h/64;
hmax=64*h;
max1=200;
Y(1)=ya;
T(1)=a;
j=1;
br=b-0.00001*abs(b);

while (T(j)<b)
   if ((T(j)+h)>br)
      h=b-T(j);
   end

   %Calculation of values in (28) and (29)
   k1=h*feval(f,T(j),Y(j));
   y2=Y(j)+b2*k1;
   end
end
R=[T' Y'];
```

Exercises for Runge-Kutta Methods

In Exercises 1 through 5, solve the differential equations by the Runge-Kutta method of order $N = 4$.

(a) Let $h = 0.2$ and do two steps by hand calculation. Then let $h = 0.1$ and do four steps by hand calculation.

(b) Compare the exact solution $y(0.4)$ with the two approximations in part (a).

(c) Does the F.G.E. in part (a) behave as expected when h is halved?

1. $y' = t^2 - y$ with $y(0) = 1$, $y(t) = -e^{-t} + t^2 - 2t + 2$

2. $y' = 3y + 3t$ with $y(0) = 1$, $y(t) = \frac{4}{3}e^{3t} - t - \frac{1}{3}$

3. $y' = -ty$ with $y(0) = 1$, $y(t) = e^{-t^2/2}$

4. $y' = e^{-2t} - 2y$ with $y(0) = \frac{1}{10}$, $y(t) = \frac{1}{10}e^{-2t} + te^{-2t}$

5. $y' = 2ty^2$ with $y(0) = 1$, $y(t) = 1/(1 - t^2)$

6. Show that when the Runge-Kutta method of order $N = 4$ is used to solve the I.V.P. $y' = f(t)$ over $[a, b]$ with $y(a) = 0$ the result is

$$y(b) \approx \frac{h}{6} \sum_{k=0}^{M-1} (f(t_k) + 4f(t_{k+1/2}) + f(t_{k+1})),$$

where $h = (b - a)/M$, and $t_k = a + kh$, and $t_{k+1/2} = a + \left(k + \frac{1}{2}\right)h$, which is Simpson's approximation (with step size $h/2$) for the definite integral of $f(t)$ taken over the interval $[a, b]$.

7. The Richardson improvement method discussed in Lemma 7.1 (Section 7.3) can be used in conjunction with the Runge-Kutta method. If the Runge-Kutta method of order $N = 4$ is used with step size h, we have

$$y(b) \approx y_h + Ch^4.$$

If the Runge-Kutta method of order $N = 4$ is used with step size $2h$, we have

$$y(b) \approx y_{2h} + 16Ch^4.$$

The terms involving Ch^4 can be eliminated to obtain an improved approximation for $y(b)$, and the result is

$$y(b) \approx \frac{16y_h - y_{2h}}{15}.$$

This improvement scheme can be used with the values in Example 9.11 to obtain better approximations to $y(3)$. Find the missing entries in the following table.

h	y_h	$(16y_h - y_{2h})/15$
1	1.6701860	____________
$\frac{1}{2}$	1.6694308	____________
$\frac{1}{4}$	1.6693928	____________
$\frac{1}{8}$	1.6693906	____________

For Exercises 8 and 9, the Taylor polynomial of degree $N = 2$ for a function $f(t, y)$ of two variables t and y expanded about the point (a, b) is

$$\begin{aligned} P_2(t, y) &= f(a, b) + f_t(a, b)(t - a) + f_y(a, b)(y - b) \\ &\quad + \frac{f_{tt}(a, b)(t - a)^2}{2} + f_{ty}(a, b)(t - a)(y - b) + \frac{f_{yy}(a, b)(y - b)^2}{2}. \end{aligned}$$

8. **(a)** Find the Taylor polynomial of degree $N = 2$ for $f(t, y) = y/t$ expanded about $(1, 1)$.

(b) Find $P_2(1.05, 1.1)$ and compare with $f(1.05, 1.1)$.

9. **(a)** Find the Taylor polynomial of degree $N = 2$ for $f(t, y) = (1 + t - y)^{1/2}$ expanded about $(0, 0)$.

(b) Find $P_2(0.04, 0.08)$ and compare with $f(0.04, 0.08)$.

Algorithms and Programs

In Problems 1 through 5, solve the differential equations by the Runge-Kutta method of order $N = 4$.

(a) Let $h = 0.1$ and do 20 steps with Program 9.4. Then let $h = 0.05$ and do 40 steps with Program 9.4.

(b) Compare the exact solution $y(2)$ with the two approximations in part (a).

(c) Does the F.G.E. in part (a) behave as expected when h is halved?

(d) Plot the two approximations and the exact solution on the same coordinate system. *Hint.* The output matrix `R` from Program 9.4 contains the x- and y-coordinates of the approximations. The command `plot(R(:,1),R(:,2))` will produce a graph analogous to Figure 9.6.

1. $y' = t^2 - y$ with $y(0) = 1$, $y(t) = -e^{-t} + t^2 - 2t + 2$

2. $y' = 3y + 3t$ with $y(0) = 1$, $y(t) = \frac{4}{3}e^{3t} - t - \frac{1}{3}$

3. $y' = -ty$ with $y(0) = 1$, $y(t) = e^{-t^2/2}$

4. $y' = e^{-2t} - 2y$ with $y(0) = \frac{1}{10}$, $y(t) = \frac{1}{10}e^{-2t} + te^{-2t}$

Numerical Solutions

A numerical solution to (1) over the interval $a \leq t \leq b$ is found by considering the differentials

$$dx = f(t, x, y)\, dt \qquad \text{and} \qquad dy = g(t, x, y)\, dt. \tag{5}$$

Euler's method for solving the system is easy to formulate. The differentials $dt = t_{k+1} - t_k$, $dx = x_{k+1} - x_k$, and $dy = y_{k+1} - y_k$ are substituted into (5) to get

$$\begin{aligned} x_{k+1} - x_k &\approx f(t_k, x_k, y_k)(t_{k+1} - t_k), \\ y_{k+1} - y_k &\approx g(t_k, x_k, y_k)(t_{k+1} - t_k). \end{aligned} \tag{6}$$

The interval is divided into M subintervals of width $h = (b - a)/M$, and the mesh points are $t_{k+1} = t_k + h$. This is used in (6) to get the recursive formulas for Euler's method:

$$\begin{aligned} t_{k+1} &= t_k + h, \\ x_{k+1} &= x_k + hf(t_k, x_k, y_k), \\ y_{k+1} &= y_k + hg(t_k, x_k, y_k) \qquad \text{for } k = 0,\ 1,\ \ldots,\ M - 1. \end{aligned} \tag{7}$$

A higher-order method should be used to achieve a reasonable amount of accuracy. For example, the Runge-Kutta formulas of order 4 are

$$\begin{aligned} x_{k+1} &= x_k + \frac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4), \\ y_{k+1} &= y_k + \frac{h}{6}(g_1 + 2g_2 + 2g_3 + g_4), \end{aligned} \tag{8}$$

where

$$\begin{aligned} f_1 &= f(t_k, x_k, y_k), & g_1 &= g(t_k, x_k, y_k), \\ f_2 &= f\left(t_k + \frac{h}{2}, x_k + \frac{h}{2}f_1, y_k + \frac{h}{2}g_1\right), & g_2 &= g\left(t_k + \frac{h}{2}, x_k + \frac{h}{2}f_1, y_k + \frac{h}{2}g_1\right), \\ f_3 &= f\left(t_k + \frac{h}{2}, x_k + \frac{h}{2}f_2, y_k + \frac{h}{2}g_2\right), & g_3 &= g\left(t_k + \frac{h}{2}, x_k + \frac{h}{2}f_2, y_k + \frac{h}{2}g_2\right), \\ f_4 &= f\left(t_k + h, x_k + hf_3, y_k + hg_3\right), & g_4 &= g\left(t_k + h, x_k + hf_3, y_k + hg_3\right). \end{aligned}$$

Example 9.15. Use the Runge-Kutta method given in (8) and compute the numerical solution to (3) over the interval [0.0, 0.2] using 10 subintervals and the step size $h = 0.02$.

For the first point we have $t_1 = 0.02$, and the intermediate calculations required to

Exercises for Systems of Differential Equations

In Exercises 1 through 4, use $h = 0.05$ and

(a) Euler's method (7) by hand to find (x_1, y_1) and (x_2, y_2).

(b) the Runge-Kutta method (8) by hand to find (x_1, y_1).

1. Solve the system $x' = 2x+3y$, $y' = 2x+y$ with the initial condition $x(0) = -2.7$ and $y(0) = 2.8$ over the interval $0 \le t \le 1.0$ using the step size $h = 0.05$. The polygonal path formed by the solution set is given in Figure 9.14 and can be compared with the analytic solution:

$$x(t) = -\frac{69}{25}e^{-t} + \frac{3}{50}e^{4t} \quad \text{and} \quad y(t) = \frac{69}{25}e^{-t} + \frac{1}{25}e^{4t}.$$

2. Solve the system $x' = 3x - y$, $y' = 4x - y$ with the initial condition $x(0) = 0.2$ and $y(0) = 0.5$ over the interval $0 \le t \le 2$ using the step size $h = 0.05$. The polygonal path formed by the solution set is given in Figure 9.15 and can be compared with the analytic solution:

$$x(t) = \frac{1}{5}e^{t} - \frac{1}{10}te^{t} \quad \text{and} \quad y(t) = \frac{1}{2}e^{t} - \frac{1}{5}te^{t}.$$

3. Solve the system $x' = x - 4y$, $y' = x + y$ with the initial condition $x(0) = 2$ and $y(0) = 3$ over the interval $0 \le t \le 2$ using the step size $h = 0.05$. The polygonal path formed by the solution set is given in Figure 9.16 and can be compared with the analytic solution:

$$x(t) = -2e^{t} + 4e^{t}\cos^2(t) - 12e^{t}\cos(t)\sin(t)$$

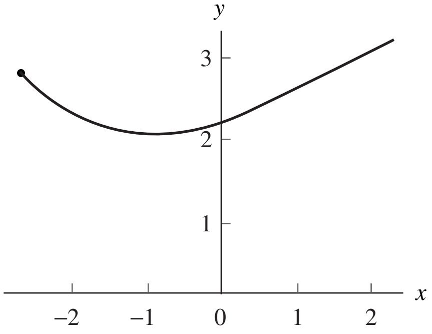


Figure 9.14 The solution to the system $x' = 2x+3y$ and $y' = 2x+y$ over [0.0, 1.0].

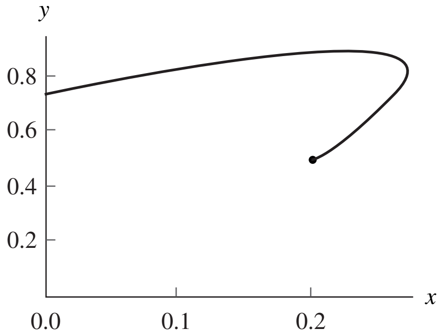


Figure 9.15 The solution to the system $x' = 3x - y$ and $y' = 4x - y$ over [0.0, 2.0].

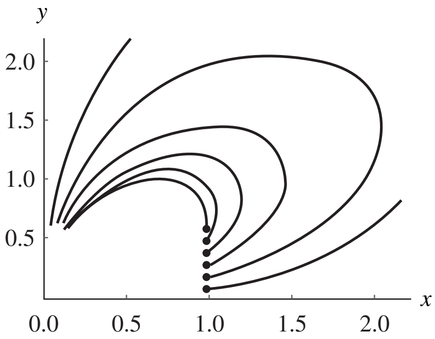


Figure 9.22 Solutions to the system $x' = x^3 - 2xy^2$ and $y' = 2x^2y - y^3$.

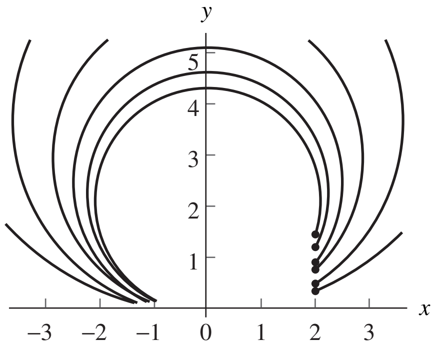


Figure 9.23 Solutions to the system $x' = x^2 - y^2$ and $y' = 2xy$.

17. Solve $x' = 1 - y$, $y' = x^2 - y^2$ with $x(0) = -1.2$ and $y(0) = 0.0$ over $[0, 5]$ using $h = 0.1$. The point $(1, 1)$ is a spiral point that is asymptotically stable, and the point $(-1, 1)$ is an unstable saddle point. The polygonal path formed by the solution set is one of the curves shown in Figure 9.21.

18. Solve $x' = x^3 - 2xy^2$, $y' = 2x^2y - y^3$ with $x(0) = 1.0$ and $y(0) = 0.2$ over $[0, 2]$ using $h = 0.025$. This system has an unstable critical point at the origin. The polygonal path formed by the solution set is one of the curves shown in Figure 9.22.

19. Solve $x' = x^2 - y^2$, $y' = 2xy$ with $x(0) = 2.0$ and $y(0) = 0.6$ over $[0.0, 1.6]$ using $h = 0.02$. The origin is an unstable critical point. The polygonal path formed by the solution set is one of the curves shown in Figure 9.23.

9.8 Boundary Value Problems

Another type of differential equation has the form

$$x'' = f(t, x, x') \quad \text{for } a \le t \le b, \tag{1}$$

with the boundary conditions

$$x(a) = \alpha \quad \text{and} \quad x(b) = \beta. \tag{2}$$

This is called a ***boundary value problem***.

The conditions that guarantee that a solution to (1) exists should be checked before any numerical scheme is applied; otherwise, a list of meaningless output may be generated. The general conditions are stated in the following theorem.

Theorem 9.8 (Boundary Value Problem). Assume that $f(t, x, y)$ is continuous on the region $R = \{(t, x, y) : a \le t \le b, -\infty < x < \infty, -\infty < y < \infty\}$ and that $\partial f/\partial x = f_x(t, x, y)$ and $\partial f/\partial y = f_y(t, x, y)$ are continuous on R. If there exists a constant $M > 0$ for which f_x and f_y satisfy

$$f_x(t, x, y) > 0 \qquad \text{for all } (t, x, y) \in R \text{ and} \tag{3}$$

$$|f_y(t, x, y)| \le M \qquad \text{for all } (t, x, y) \in R, \tag{4}$$

then the boundary value problem

$$x'' = f(t, x, x') \qquad \text{with } x(a) = \alpha \text{ and } x(b) = \beta \tag{5}$$

has a unique solution $x = x(t)$ for $a \le t \le b$.

The notation $y = x'(t)$ has been used to distinguish the third variable of the function $f(t, x, x')$. Finally, the special case of linear differential equations is worthy of mention.

Corollary 9.1 (Linear Boundary Value Problem). Assume that f in Theorem 9.8 has the form $f(t, x, y) = p(t)y + q(t)x + r(t)$ and that f and its partial derivatives $\partial f/\partial x = q(t)$ and $\partial f/\partial y = p(t)$ are continuous on R. If there exists a constant $M > 0$ for which $p(t)$ and $q(t)$ satisfy

$$q(t) > 0 \qquad \text{for all } t \in [a, b] \tag{6}$$

and

$$|p(t)| \le M = \max_{a \le t \le b} \{|p(t)|\}, \tag{7}$$

then the ***linear boundary value problem***

$$x'' = p(t)x'(t) + q\,(t)\,x(t) + r(t) \qquad \text{with } x(a) = \alpha \text{ and } x(b) = \beta \tag{8}$$

has a unique solution $x = x(t)$ over $a \le t \le b$.

Reduction to Two I.V.P.s: Linear Shooting Method

Finding the solution of a linear boundary problem is assisted by the linear structure of the equation and the use of two special initial value problems. Suppose that $u(t)$ is the unique solution to the I.V.P.

$$u'' = p(t)u'(t) + q(t)u(t) + r(t) \qquad \text{with } u(a) = \alpha \text{ and } u'(a) = 0. \tag{9}$$

Furthermore, suppose that $v(t)$ is the unique solution to the I.V.P.

$$(10) \qquad v'' = p(t)v'(t) + q(t)v(t) \quad \text{with } v(a) = 0 \text{ and } v'(a) = 1.$$

Then the linear combination

$$(11) \qquad x(t) = u(t) + Cv(t)$$

is a solution to $x'' = p(t)x'(t) + q(t)x(t) + r(t)$ as seen by the computation

$$\begin{aligned} x'' = u'' + Cv'' &= p(t)u'(t) + q(t)u(t) + r(t) + p(t)Cv'(t) + q(t)Cv(t) \\ &= p(t)(u'(t) + Cv'(t)) + q(t)(u(t) + Cv(t)) + r(t) \\ &= p(t)x'(t) + q(t)x(t) + r(t). \end{aligned}$$

The solution $x(t)$ in equation (11) takes on the boundary values

$$(12) \qquad \begin{aligned} x(a) &= u(a) + Cv(a) = \alpha + 0 = \alpha, \\ x(b) &= u(b) + Cv(b). \end{aligned}$$

Imposing the boundary condition $x(b) = \beta$ in (12) produces $C = (\beta - u(b))/v(b)$. Therefore, if $v(b) \neq 0$, the unique solution to (8) is

$$(13) \qquad x(t) = u(t) + \frac{\beta - u(b)}{v(b)}v(t).$$

Remark. If q fulfills the hypotheses of Corollary 9.1, this rules out the troublesome solution $v(t) \equiv 0$, so that (13) is the form of the required solution. The details are left for the reader to investigate in the exercises.

Example 9.17. Solve the boundary value problem

$$x''(t) = \frac{2t}{1+t^2}x'(t) - \frac{2}{1+t^2}x(t) + 1$$

with $x(0) = 1.25$ and $x(4) = -0.95$ over the interval [0, 4].

The functions p, q, and r are $p(t) = 2t/(1 + t^2)$, $q(t) = -2/(1 + t^2)$, and $r(t) = 1$, respectively. The Runge-Kutta method of order 4 with step size $h = 0.2$ is used to construct numerical solutions $\{u_j\}$ and $\{v_j\}$ to equations (9) and (10), respectively. The approximations $\{u_j\}$ for $u(t)$ are given in the first column of Table 9.15. Then $u(4) \approx u_{20} = -2.893535$ and $v(4) \approx v_{20} = 4$ are used with (13) to construct

$$w_j = \frac{b - u(4)}{v(4)}v_j = 0.485884v_j.$$

Table 9.15 Approximate Solutions $\{x_j\} = \{u_j + w_j\}$ to the Equation $x''(t) = \dfrac{2t}{1+t^2}x'(t) - \dfrac{2}{1+t^2} + 1$

t_j	u_j	w_j	$x_j = u_j + w_j$
0.0	1.250000	0.000000	1.250000
0.2	1.220131	0.097177	1.317308
0.4	1.132073	0.194353	1.326426
0.6	0.990122	0.291530	1.281652
0.8	0.800569	0.388707	1.189276
1.0	0.570844	0.485884	1.056728
1.2	0.308850	0.583061	0.891911
1.4	0.022522	0.680237	0.702759
1.6	−0.280424	0.777413	0.496989
1.8	−0.592609	0.874591	0.281982
2.0	−0.907039	0.971767	0.064728
2.2	−1.217121	1.068944	−0.148177
2.4	−1.516639	1.166121	−0.350518
2.6	−1.799740	1.263297	−0.536443
2.8	−2.060904	1.360474	−0.700430
3.0	−2.294916	1.457651	−0.837265
3.2	−2.496842	1.554828	−0.942014
3.4	−2.662004	1.652004	−1.010000
3.6	−2.785960	1.749181	−1.036779
3.8	−2.864481	1.846358	−1.018123
4.0	−2.893535	1.943535	−0.950000

Then the required approximate solution is $\{x_j\} = \{u_j + w_j\}$. Sample computations are given in Table 9.15, and Figure 9.24 shows their graphs. The reader can verify that $v(t) = t$ is the analytic solution for boundary value problem (10); that is,

$$v''(t) = \frac{2t}{1+t^2}v'(t) - \frac{2}{1+t^2}v(t)$$

with the initial conditions $v(0) = 0$ and $v'(0) = 1$.

The approximations in Table 9.16 compare numerical solutions obtained with the linear shooting method with the step sizes $h = 0.2$ and $h = 0.1$ and the analytic solution

$$x(t) = 1.25 + 0.4860896526t - 2.25t^2 + 2t\arctan(t) - \frac{1}{2}\ln(1+t^2) + \frac{1}{2}t^2\ln(1+t^2).$$

A graph of the approximate solution when $h = 0.2$ is given in Figure 9.25. Included in the table are columns for the error. Since the Runge-Kutta solutions have error of order $\boldsymbol{O}(h^4)$, the error in the solution with the smaller step size $h = 0.1$ is about $\frac{1}{16}$ the error of the solution with the large step size $h = 0.2$. ■

Program 9.10 will call Program 9.9 to solve the initial value problems (9) and (10). Program 9.9 approximates solutions of systems of differential equations using a modification of the Runge-Kutta method of order $N = 4$. Thus, it is necessary to save

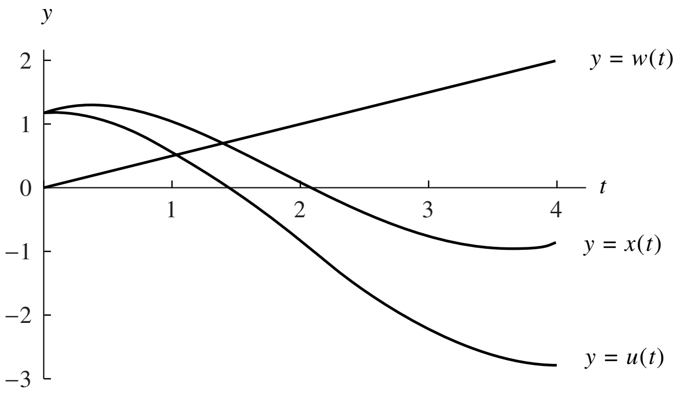


Figure 9.24 The numerical approximations $u(t)$ and $w(t)$ used to form $x(t) = u(t) + w(t)$, which is the solution to

$$x''(t) = \frac{2t}{1+t^2}x'(t) - \frac{2}{1+t^2}x(t) + 1.$$

Table 9.16 Numerical Approximations for $x''(t) = \dfrac{2t}{1+t^2}x'(t) - \dfrac{2}{1+t^2}x(t) + 1$

t_j	x_j $h = 0.2$	$x(t_j)$ exact	$x(t_j)-x_j$ error	t_j	x_j $h = 0.1$	$x(t_j)$ exact	$x(t_j)-x_j$ error
0.0	1.250000	1.250000	0.000000	0.0	1.250000	1.250000	0.000000
				0.1	1.291116	1.291117	0.000001
0.2	1.317308	1.317350	0.000042	0.2	1.317348	1.317350	0.000002
				0.3	1.328986	1.328990	0.000004
0.4	1.326426	1.326505	0.000079	0.4	1.326500	1.326505	0.000005
				0.5	1.310508	1.310514	0.000006
0.6	1.281652	1.281762	0.000110	0.6	1.281756	1.281762	0.000006
0.8	1.189276	1.189412	0.000136	0.8	1.189404	1.189412	0.000008
1.0	1.056728	1.056886	0.000158	1.0	1.056876	1.056886	0.000010
1.2	0.891911	0.892086	0.000175	1.2	0.892076	0.892086	0.000010
1.6	0.496989	0.497187	0.000198	1.6	0.497175	0.497187	0.000012
2.0	0.064728	0.064931	0.000203	2.0	0.064919	0.064931	0.000012
2.4	−0.350518	−0.350325	0.000193	2.4	−0.350337	−0.350325	0.000012
2.8	−0.700430	−0.700262	0.000168	2.8	−0.700273	−0.700262	0.000011
3.2	−0.942014	−0.941888	0.000126	3.2	−0.941895	−0.941888	0.000007
3.6	−1.036779	−1.036708	0.000071	3.6	−1.036713	−1.036708	0.000005
4.0	−0.950000	−0.950000	0.000000	4.0	−0.950000	−0.950000	0.000000

the equations (9) and (10) in the form of the system of equations (11) of Section 9.7. As an illustration, consider the boundary value problem in Example 9.17. The follow-

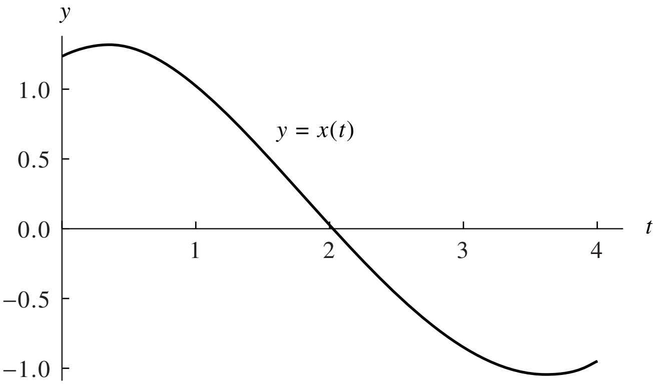


Figure 9.25 The graph of the numerical approximation for

$$x''(t) = \frac{2t}{1+t^2}x'(t) - \frac{2}{1+t^2}x(t) + 1$$

(using $h = 0.2$).

ing M-file, named F1, will save the I.V.P. (9) in the form of a system of differential equations.

```
function Z=F1(t,Z)
x=Z(1);y=Z(2);
Z=[y,2*t*y/(1+t^2)-2*x/(1+t^2)+1];
```

A similar M-file, named F2, will save the I.V.P. (10) (just let $r(t) = 0$ in F1) in the appropriate form.

A plot of the approximation obtained from Program 9.10 can be constructed by using the command `plot(L(:,1), L(:,2))`.

Program 9.9 (Runge-Kutta Method of Order $N = 4$ for Systems). To approximate the solution of the system of differential equations

$$\begin{aligned} x_1'(t) &= f_1(t, x_1(t), \ldots, x_n(t)) \\ \vdots \quad & \quad \vdots \\ x_n'(t) &= f_n(t, x_1(t), \ldots, x_n(t)) \end{aligned}$$

with $x_1(a) = \alpha_1, \ldots, x_n(a) = \alpha_n$ over the interval $[a, b]$.

```
function [T,Z]=rks4(F,a,b,Za,M)
%Input   - F is the system input as a string 'F'
%        - a and b are the endpoints of the interval
%        - Za=[x(a) y(a)] are the initial conditions
%        - M is the number of steps
```

Exercises for Boundary Value Problems

1. Verify that the function $x(t)$ is the solution to the boundary value problem.

(a) $x'' = (-2/t)x' + (2/t^2)x + (10\cos(\ln(t)))/t^2$ over $[1, 3]$ with $x(1) = 1$ and $x(3) = -1$.

$$x(t) = \frac{4.335950689 - 0.3359506908t^3 - 3t^2\cos(\ln(t)) + t^2\sin(\ln(t))}{t^2}$$

(b) $x'' = -2x' - 2x + e^{-t} + \sin(2t)$ over $[0, 4]$ with $x(0) = 0.6$ and $x(4) = -0.1$.

$$x(t) = \frac{1}{5} + e^{-t} - \frac{1}{5}e^{-t}\cos(t) - \frac{2}{5}\cos^2(t) + 3.670227413e^{-t}\sin(t) - \frac{1}{5}\cos(t)\sin(t)$$

(c) $x'' = -4x' - 4x + 5\cos(4t) + \sin(2t)$ over $[0, 2]$ with $x(0) = 0.75$ and $x(2) = 0.25$.

$$x(t) = -\frac{1}{40} + 1.025e^{-2t} - 1.915729975te^{-2t} + \frac{19}{20}\cos^2(t) - \frac{6}{5}\cos^4(t) - \frac{4}{5}\cos(t)\sin(t) + \frac{8}{5}\cos^3(t)\sin(t)$$

(d) $x'' + (1/t)x' + (1 - 1/(4t^2))x = 0$ over $[1, 6]$ with $x(1) = 1$ and $x(6) = 0$.

$$x(t) = \frac{0.2913843206\cos(t) + 1.001299385\sin(t)}{\sqrt{t}}$$

(e) $x'' - (1/t)x' + (1/t^2)x = 1$ over $[0.5, 4.5]$ with $x(0.5) = 1$ and $x(4.5) = 2$.

$$x(t) = t^2 - 0.2525826491t - 2.528442297t\ln(t)$$

2. Does the boundary value problem in Exercise 1(e) satisfy the hypotheses of Corollary 9.1? Explain.

3. If q fulfills the hypothesis of Corollary 9.1, show that $v(t) \equiv 0$ is the unique solution to the boundary value problem

$$v'' = p(t)v'(t) + q(t)v(t) \quad \text{with} \quad v(a) = 0 \ \text{and} \ v(b) = 0.$$

Algorithms and Programs

1. (a) Use Programs 9.9 and 9.10 to solve each of the boundary value problems in Exercise 1, using the step size $h = 0.05$.

(b) Graph your solution and the actual solution on the same coordinate system.

2. Construct programs analogous to Program 9.9 based on

(a) Heun's method,

(b) the Adams-Bashforth-Moulton method, and

(c) Hamming's method.

3. (a) Modify Program 9.10 to call each of your programs from Problem 2.

(b) Use your programs to solve each of the five boundary value problems in Exercise 1 using the step size $h = 0.05$.

(c) Graph your solutions and the actual solution on the same coordinate system.

9.9 Finite-Difference Method

Methods involving difference quotient approximations for derivatives can be used for solving certain second-order boundary value problems. Consider the linear equation

$$x'' = p(t)x'(t) + q(t)x(t) + r(t) \tag{1}$$

over $[a, b]$ with $x(a) = \alpha$ and $x(b) = \beta$. Form a partition of $[a, b]$ using the points $a = t_0 < t_1 < \cdots < t_N = b$, where $h = (b - a)/N$ and $t_j = a + jh$ for $j = 0, 1, \ldots, N$. The central-difference formulas discussed in Chapter 6 are used to approximate the derivatives

$$x'(t_j) = \frac{x(t_{j+1}) - x(t_{j-1})}{2h} + \boldsymbol{O}(h^2) \tag{2}$$

and

$$x''(t_j) = \frac{x(t_{j+1}) - 2x(t_j) + x(t_{j-1})}{h^2} + \boldsymbol{O}(h^2). \tag{3}$$

To start the derivation, we replace each term $x(t_j)$ on the right side of (2) and (3) with x_j, and the resulting equations are substituted into (1) to obtain the relation

$$\begin{aligned}\frac{x_{j+1} - 2x_j + x_{j-1}}{h^2} + \boldsymbol{O}(h^2) = p(t_j)&\left(\frac{x_{j+1} - x_{j-1}}{2h} + \boldsymbol{O}(h^2)\right)\\ &+ q(t_j)x_j + r(t_j).\end{aligned} \tag{4}$$

Next, we drop the two terms $\boldsymbol{O}(h^2)$ in (4) and introduce the notation $p_j = p(t_j)$, $q_j = q(t_j)$, and $r_j = r(t_j)$; this produces the difference equation

$$\frac{x_{j+1} - 2x_j + x_{j-1}}{h^2} = p_j \frac{x_{j+1} - x_{j-1}}{2h} + q_j x_j + r_j, \tag{5}$$

which is used to compute numerical approximations to the differential equation (1). This is carried out by multiplying each side of (5) by h^2 and then collecting terms involving x_{j-1}, x_j, and x_{j+1} and arranging them in a system of linear equations:

$$\left(\frac{-h}{2}p_j - 1\right)x_{j-1} + (2 + h^2 q_j)x_j + \left(\frac{h}{2}p_j - 1\right)x_{j+1} = -h^2 r_j, \tag{6}$$

for $j = 1, 2, \ldots, N-1$, where $x_0 = \alpha$ and $x_N = \beta$. The system in (6) has the familiar tridiagonal form, which is more visible when displayed with matrix notation:

$$\begin{bmatrix} 2+h^2q_1 & \frac{h}{2}p_1-1 & & & & \\ \frac{-h}{2}p_2-1 & 2+h^2q_2 & \frac{h}{2}p_2-1 & & & \boldsymbol{O} \\ & \frac{-h}{2}p_j-1 & 2+h^2q_j & \frac{h}{2}p_j-1 & & \\ \boldsymbol{O} & & \frac{-h}{2}p_{N-2}-1 & 2+h^2q_{N-2} & \frac{h}{2}p_{N-2}-1 & \\ & & & \frac{-h}{2}p_{N-1}-1 & 2+h^2q_{N-1} & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_j \\ x_{N-2} \\ x_{N-1} \end{bmatrix}$$

$$= \begin{bmatrix} -h^2r_1+e_0 \\ -h^2r_2 \\ -h^2r_j \\ -h^2r_{N-2} \\ -h^2r_{N-1}+e_N \end{bmatrix},$$

where

$$e_0 = \left(\frac{h}{2}p_1 + 1\right)\alpha \quad \text{and} \quad e_N = \left(\frac{-h}{2}p_{N-1} + 1\right)\beta.$$

When computations with step size h are used, the numerical approximation to the solution is a set of discrete points $\{(t_j, x_j)\}$; if the analytic solution $x(t_j)$ is known, we can compare x_j and $x(t_j)$.

Example 9.18. Solve the boundary value problem

$$x''(t) = \frac{2t}{1+t^2}x'(t) - \frac{2}{1+t^2}x(t) + 1$$

with $x(0) = 1.25$ and $x(4) = -0.95$ over the interval $[0, 4]$.

The functions p, q, and r are $p(t) = 2t/(1+t^2)$, $q(t) = -2/(1+t^2)$, and $r(t) = 1$, respectively. The finite-difference method is used to construct numerical solutions $\{x_j\}$ using the system of equations (6). Sample values of the approximations $\{x_{j,1}\}$, $\{x_{j,2}\}$, $\{x_{j,3}\}$, and $\{x_{j,4}\}$ corresponding to the step sizes $h_1 = 0.2$, $h_2 = 0.1$, $h_3 = 0.05$, and $h_4 = 0.025$ are given in Table 9.17. Figure 9.26 shows the graph of the polygonal path formed from $\{(t_j, x_{j,1})\}$ for the case $h_1 = 0.2$. There are 41 terms in the sequence generated with $h_2 = 0.1$, and the sequence $\{x_{j,2}\}$ only includes every other term from these computations; they correspond to the 21 values of $\{t_j\}$ given in Table 9.17. Similarly, the sequences $\{x_{j,3}\}$ and $\{x_{j,4}\}$ are a portion of the values generated with step sizes $h_3 = 0.05$ and $h_4 = 0.025$, respectively, and they correspond to the 21 values of $\{t_j\}$ in Table 9.17.

Next we compare numerical solutions in Table 9.17 with the analytic solution: $x(t) = 1.25 + 0.486089652t - 2.25t^2 + 2t\arctan(t) - \frac{1}{2}\ln(1+t^2) + \frac{1}{2}t^2\ln(1+t^2)$. The numerical

Table 9.17 Numerical Approximations for $x''(t) = \frac{2t}{1+t^2}x'(t) - \frac{2}{1+t^2}x(t) + 1$

t_j	$x_{j,1}$ $h = 0.2$	$x_{j,2}$ $h = 0.1$	$x_{j,3}$ $h = 0.05$	$x_{j,4}$ $h = 0.025$	$x(t_j)$ exact
0.0	1.250000	1.250000	1.250000	1.250000	1.250000
0.2	1.314503	1.316646	1.317174	1.317306	1.317350
0.4	1.320607	1.325045	1.326141	1.326414	1.326505
0.6	1.272755	1.279533	1.281206	1.281623	1.281762
0.8	1.177399	1.186438	1.188670	1.189227	1.189412
1.0	1.042106	1.053226	1.055973	1.056658	1.056886
1.2	0.874878	0.887823	0.891023	0.891821	0.892086
1.4	0.683712	0.698181	0.701758	0.702650	0.702947
1.6	0.476372	0.492027	0.495900	0.496865	0.497187
1.8	0.260264	0.276749	0.280828	0.281846	0.282184
2.0	0.042399	0.059343	0.063537	0.064583	0.064931
2.2	−0.170616	−0.153592	−0.149378	−0.148327	−0.147977
2.4	−0.372557	−0.355841	−0.351702	−0.350669	−0.350325
2.6	−0.557565	−0.541546	−0.537580	−0.536590	−0.536261
2.8	−0.720114	−0.705188	−0.701492	−0.700570	−0.700262
3.0	−0.854988	−0.841551	−0.838223	−0.837393	−0.837116
3.2	−0.957250	−0.945700	−0.942839	−0.942125	−0.941888
3.4	−1.022221	−1.012958	−1.010662	−1.010090	−1.009899
3.6	−1.045457	−1.038880	−1.037250	−1.036844	−1.036709
3.8	−1.022727	−1.019238	−1.018373	−1.018158	−1.018086
4.0	−0.950000	−0.950000	−0.950000	−0.950000	−0.950000

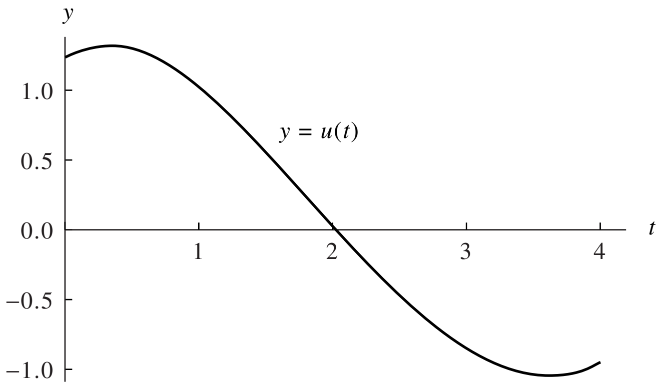


Figure 9.26 The graph of the numerical approximation for $x(t) = u(t) + w(t)$, which is the solution to

$$x''(t) = \frac{2t}{1+t^2}x'(t) - \frac{2}{1+t^2}x(t) + 1$$

(using $h = 0.2$).

Table 9.18 Errors in Numerical Approximations Using the Finite-Difference Method

t_j	$x(t_j) - x_{j,1}$ $= e_{j,1}$	$x(t_j) - x_{j,2}$ $= e_{j,2}$	$x(t_j) - x_{j,3}$ $= e_{j,3}$	$x(t_j) - x_{j,4}$ $= e_{j,4}$
	$h_1 = 0.2$	$h_2 = 0.1$	$h_3 = 0.05$	$h_4 = 0.025$
0.0	0.000000	0.000000	0.000000	0.000000
0.2	0.002847	0.000704	0.000176	0.000044
0.4	0.005898	0.001460	0.000364	0.000091
0.6	0.009007	0.002229	0.000556	0.000139
0.8	0.012013	0.002974	0.000742	0.000185
1.0	0.014780	0.003660	0.000913	0.000228
1.2	0.017208	0.004263	0.001063	0.000265
1.4	0.019235	0.004766	0.001189	0.000297
1.6	0.020815	0.005160	0.001287	0.000322
1.8	0.021920	0.005435	0.001356	0.000338
2.0	0.022533	0.005588	0.001394	0.000348
2.2	0.022639	0.005615	0.001401	0.000350
2.4	0.022232	0.005516	0.001377	0.000344
2.6	0.021304	0.005285	0.001319	0.000329
2.8	0.019852	0.004926	0.001230	0.000308
3.0	0.017872	0.004435	0.001107	0.000277
3.2	0.015362	0.003812	0.000951	0.000237
3.4	0.012322	0.003059	0.000763	0.000191
3.6	0.008749	0.002171	0.000541	0.000135
3.8	0.004641	0.001152	0.000287	0.000072
4.0	0.000000	0.000000	0.000000	0.000000

solutions can be shown to have error of order $\boldsymbol{O}(h^2)$. Hence reducing the step size by a factor of $\frac{1}{2}$ results in the error being reduced by about $\frac{1}{4}$. A careful scrutiny of Table 9.18 will reveal that this is happening. For instance, at $t_j = 1.0$ the errors incurred with step sizes h_1, h_2, h_3, and h_4 are $e_{j,1} = 0.014780$, $e_{j,2} = 0.003660$, $e_{j,3} = 0.000913$, and $e_{j,4} = 0.000228$, respectively. Their successive ratios $e_{j,2}/e_{j,1} = 0.003660/0.014780 = 0.2476$, $e_{j,3}/e_{j,2} = 0.000913/0.003660 = 0.2495$, and $e_{j,4}/e_{j,3} = 0.000228/0.000913 = 0.2497$ are approaching $\frac{1}{4}$.

Finally, we show how Richardson's improvement scheme can be used to extrapolate the seemingly inaccurate sequences $\{x_{j,1}\}$, $\{x_{j,2}\}$, $\{x_{j,3}\}$, and $\{x_{j,4}\}$ and obtain six digits of precision. Eliminate the error terms $\boldsymbol{O}(h^2)$ and $\boldsymbol{O}((h/2)^2)$ in the approximations $\{x_{j,1}\}$ and $\{x_{j,2}\}$ by generating the extrapolated sequence $\{z_{j,1}\} = \{(4x_{j,2} - x_{j,1})/3\}$. Similarly, the error terms $\boldsymbol{O}((h/2)^2)$ and $\boldsymbol{O}((h/4)^2)$ for $\{x_{j,2}\}$ and $\{x_{j,3}\}$ are eliminated by generating $\{z_{j,2}\} = \{(4x_{j,3} - x_{j,2})/3\}$. It has been shown that the second level of Richardson's improvement scheme applies to the sequences $\{z_{j,1}\}$ and $\{z_{j,2}\}$, so the third improvement is $\{(16z_{j,2} - z_{j,1})/15\}$. Let us illustrate the situation by finding the extrapolated values that

Table 9.19 Extrapolation of the Numerical Approximations $\{x_{j,1}\}$, $\{x_{j,2}\}$, $\{x_{j,3}\}$ Obtained with the Finite-Difference Method

t_j	$\frac{4x_{j,2}-x_{j,1}}{3}$ $= z_{j,1}$	$\frac{4x_{j,3}-x_{j,2}}{3}$ $= z_{j,2}$	$\frac{16z_{j,2}-z_{j,1}}{3}$	$x(t_j)$ Exact solution
0.0	1.250000	1.250000	1.250000	1.250000
0.2	1.317360	1.317351	1.317350	1.317350
0.4	1.326524	1.326506	1.326504	1.326505
0.6	1.281792	1.281764	1.281762	1.281762
0.8	1.189451	1.189414	1.189412	1.189412
1.0	1.056932	1.056889	1.056886	1.056886
1.2	0.892138	0.892090	0.892086	0.892086
1.4	0.703003	0.702951	0.702947	0.702948
1.6	0.497246	0.497191	0.497187	0.497187
1.8	0.282244	0.282188	0.282184	0.282184
2.0	0.064991	0.064935	0.064931	0.064931
2.2	−0.147918	−0.147973	−0.147977	−0.147977
2.4	−0.350268	−0.350322	−0.350325	−0.350325
2.6	−0.536207	−0.536258	−0.536261	−0.536261
2.8	−0.700213	−0.700259	−0.700263	−0.700262
3.0	−0.837072	−0.837113	−0.837116	−0.837116
3.2	−0.941850	−0.941885	−0.941888	−0.941888
3.4	−1.009870	−1.009898	−1.009899	−1.009899
3.6	−1.036688	−1.036707	−1.036708	−1.036708
3.8	−1.018075	−1.018085	−1.018086	−1.018086
4.0	−0.950000	−0.950000	−0.950000	−0.950000

correspond to $t_j = 1.0$. The first extrapolated value is

$$\frac{4x_{j,2} - x_{j,1}}{3} = \frac{4(1.053226) - 1.042106}{3} = 1.056932 = z_{j,1}.$$

The second extrapolated value is

$$\frac{4x_{j,3} - x_{j,2}}{3} = \frac{4(1.055973) - 1.053226}{3} = 1.056889 = z_{j,2}.$$

Finally, the third extrapolation involves the terms $z_{j,1}$ and $z_{j,2}$:

$$\frac{16z_{j,2} - z_{j,1}}{15} = \frac{16(1.056889) - 1.056932}{15} = 1.056886.$$

This last computation contains six decimal places of accuracy. The values at the other points are given in Table 9.19. ■

Program 9.12 will call Program 9.11 to solve the tridiagonal system (6). Program 9.12 requires that the coefficient functions $p(t)$, $q(t)$, and $r(t)$ (boundary value problem (1)) be saved in M-files `p.m`, `q.m`, and `r.m`, respectively.

Program 9.11 (Tridiagonal Systems). To solve the tridiagonal system $\boldsymbol{CX} = \boldsymbol{B}$, where $\boldsymbol{C}$ is a tridiagonal matrix.

```
function X=trisys(A,D,C,B)

%Input  - A is the subdiagonal of the coefficient matrix
%       - D is the main diagonal of the coefficient matrix
%       - C is the superdiagonal of the coefficient matrix
%       - B is the constant vector of the linear system
%Output - X is the solution vector

N=length(B);
for k=2:N
   mult=A(k-1)/D(k-1);
   D(k)=D(k)-mult*C(k-1);
   B(k)=B(k)-mult*B(k-1);
end
X(N)=B(N)/D(N);
for k= N-1:-1:1
   X(k)=(B(k)-C(k)*X(k+1))/D(k);
end
```

Program 9.12 (Finite-Difference Method). To approximate the solution of the boundary value problem $x'' = p(t)x'(t) + q(t)x(t) + r(t)$ with $x(a) = \alpha$ and $x(b) = \beta$ over the interval $[a, b]$ by using the finite-difference method of order $\boldsymbol{O}(h^2)$.
Remark. The mesh is $a = t_1 < \cdots < t_{N+1} = b$ and the solution points are $\{(t_j, x_j)\}_{j=1}^{N+1}$.

```
function F=findiff(p,q,r,a,b,alpha,beta,N)

%Input  - p,q,and r are the coefficient functions of (1)
%         input as strings; 'p','q','r'
%       - a and b are the left and right endpoints
%       - alpha=x(a) and beta=x(b)
%       - N is the number of steps
%Output - F=[T' X']:where T' is the 1xN vector of abscissas
%         and X' is the 1xN vector of ordinates

%Initialize vectors and h
T=zeros(1,N+1);
X=zeros(1,N-1);
Va=zeros(1,N-2);
Vb=zeros(1,N-1);
Vc=zeros(1,N-2);
Vd=zeros(1,N-1);
h=(b-a)/N;
```

```
%Calculate the constant vector B in AX=B
Vt=a+h:h:a+h*(N-1);
Vb=-h^2*feval(r,Vt);
Vb(1)=Vb(1)+(1+h/2*feval(p,Vt(1)))*alpha;
Vb(N-1)=Vb(N-1)+(1-h/2*feval(p,Vt(N-1)))*beta;

%Calculate the main diagonal of A in AX=B
Vd=2+h^2*feval(q,Vt);

%Calculate the superdiagonal of A in AX=B
Vta=Vt(1,2:N-1);
Va=-1-h/2*feval(p,Vta);

%Calculate the subdiagonal of A in AX=B
Vtc=Vt(1,1:N-2);
Vc=-1+h/2*feval(p,Vtc);

%Solve AX=B using trisys
X=trisys(Va,Vd,Vc,Vb);
T=[a,Vt,b];
X=[alpha,X,beta];
F=[T' X'];
```

Exercises for Finite-Difference Method

In Exercises 1 through 3, use the finite-difference method to approximate $x(a + 0.5)$.

(a) Let $h_1 = 0.5$ and do one step by hand calculation. Then let $h_2 = 0.25$ and do two steps by hand calculation.

(b) Use extrapolation of the values in part (a) to obtain a better approximation (i.e., $z_{j,1} = (4x_{j,2} - x_{j,1})/3$).

(c) Compare your results from parts (a) and (b) with the exact value $x(a + 0.5)$.

1. $x'' = 2x' - x + t^2 - 1$ over $[0, 1]$ with $x(0) = 5$ and $x(1) = 10$
$x(t) = t^2 + 4t + 5$

2. $x'' + (1/t)x' + (1 - 1/(4t^2))x = 0$ over $[1, 6]$ with $x(1) = 1$ and $x(6) = 0$
$$x(t) = \frac{0.2913843206\cos(t) + 1.001299385\sin(t)}{\sqrt{t}}$$

3. $x'' - (1/t)x' + (1/t^2)x = 1$ over $[0.5, 4.5]$ with $x(0.5) = 1$ and $x(4.5) = 2$
$x(t) = t^2 - 0.2525826491t - 2.528442297t\ln(t)$

4. Assume that p, q, and r are continuous over the interval $[a, b]$ and that $q(t) \geq 0$ for $a \leq t \leq b$. If h satisfies $0 < h < 2/M$, where $M = \max_{a \leq t \leq b}\{|p(t)|\}$, prove that the coefficient matrix of (6) is strictly diagonally dominant and that there is a unique solution.

5. Assume that $p(t) \equiv C_1 > 0$ and $q(t) \equiv C_2 > 0$. **(a)** Write out the tridiagonal linear system for this situation. **(b)** Prove that the tridiagonal system is strictly diagonally dominant and hence has a unique solution, provided that $C_1/C_2 \le h$.

Algorithms and Programs

1. Use Programs 9.11 and 9.12 to solve the given boundary problem using step sizes $h = 0.1$ and $h = 0.01$. Plot your two approximate solutions and the actual solution on the same coordinate system.
   (a) $x'' = 2x' - x + t^2 - 1$ over $[0, 1]$ with $x(0) = 5$ and $x(1) = 10$
   $x(t) = t^2 + 4t + 5$
   (b) $x'' + (1/t)x' + (1 - 1/(4t^2))x = 0$ over $[1, 6]$ with $x(1) = 1$ and $x(6) = 0$
   $$x(t) = \frac{0.2913843206\cos(t) + 1.001299385\sin(t)}{\sqrt{t}}$$
   (c) $x'' - (1/t)x' + (1/t^2)x = 1$ over $[0.5, 4.5]$ with $x(0.5) = 1$ and $x(4.5) = 2$
   $x(t) = t^2 - 0.2525826491t - 2.528442297t\ln(t)$

In Problems 2 through 7, use Programs 9.11 and 9.12 to solve the given boundary problem using step sizes $h = 0.2$, $h = 0.1$, and $h = 0.05$. For each problem, graph the three solutions on the same coordinate system.

2. $x'' = (-2/t)x' + (2/t^2)x + (10\cos(\ln(t)))/t^2$ over $[1, 3]$ with $x(1) = 1$ and $x(3) = -1$
3. $x'' = -5x' - 6x + te^{-2t} + 3.9\cos(3t)$ over $[0, 3]$ with $x(0) = 0.95$ and $x(3) = 0.15$
4. $x'' = -4x' - 4x + 5\cos(4t) + \sin(2t)$ over $[0, 2]$ with $x(0) = 0.75$ and $x(2) = 0.25$
5. $x'' = -2x' - 2x + e^{-t} + \sin(2t)$ over $[0, 4]$ with $x(0) = 0.6$ and $x(4) = -0.1$
6. $x'' + (2/t)x' - (2/t^2)x = \sin(t)/t^2$ over $[1, 6]$ with $x(1) = -0.02$ and $x(6) = 0.02$
7. $x'' + (1/t)x' + (1 - 1/(4t^2))x = \sqrt{t}\cos(t)$ over $[1, 6]$ with $x(1) = 1.0$ and $x(6) = -0.5$
8. Construct a program that will call Programs 9.11 and 9.12 and carry out the extrapolation process illustrated in Example 9.18 and Table 9.19.
9. For each of the given boundary value problems, use your program from Problem 8 and the step sizes $h = 0.1$, $h = 0.05$, and $h = 0.025$ to construct a table analogous to Table 9.19. Plot your extrapolated solution and the actual solution on the same coordinate system.
   (a) $x'' = 2x' - x + t^2 - 1$ over $[0, 1]$ with $x(0) = 5$ and $x(1) = 10$
   $x(t) = t^2 + 4t + 5$
   (b) $x'' + (1/t)x' + (1 - 1/(4t^2))x = 0$ over $[1, 6]$ with $x(1) = 1$ and $x(6) = 0$
   $$x(t) = \frac{0.2913843206\cos(t) + 1.001299385\sin(t)}{\sqrt{t}}$$
   (c) $x'' - (1/t)x' + (1/t^2)x = 1$ over $[0.5, 4.5]$ with $x(0.5) = 1$ and $x(4.5) = 2$
   $x(t) = t^2 - 0.2525826491t - 2.528442297t\ln(t)$