CHAPTER

1 Mathematical Preliminaries and Error Analysis

1.1 Introduction

This book examines problems that can be solved by methods of approximation, techniques called *numerical methods*. We begin by considering some of the mathematical and computational topics that arise when approximating a solution to a problem. Nearly all the problems whose solutions can be approximated involve continuous functions, so calculus is the principal tool to use for deriving numerical methods and verifying that they solve the problems. The calculus definitions and results included in the next section provide a handy reference when these concepts are needed later in the book.

There are two things to consider when applying a numerical technique. The first and most obvious is to obtain the approximation. The equally important second objective is to determine a safety factor for the approximation: some assurance, or at least a sense, of the accuracy of the approximation. Sections 1.3 and 1.4 deal with a standard difficulty that occurs when applying techniques to approximate the solution to a problem:

- Where and why is computational error produced and how can it be controlled?

The final section in this chapter describes various types and sources of mathematical software for implementing numerical methods.

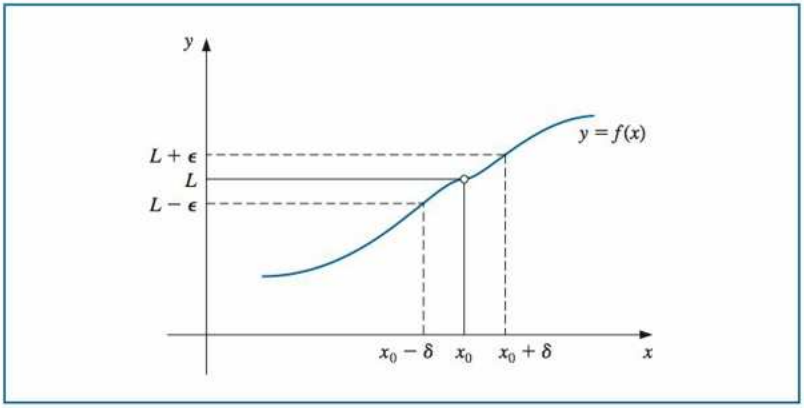
1.2 Review of Calculus

Limits and Continuity

The limit of a function at a specific number tells, in essence, what the function values approach as the numbers in the domain approach the specific number. The limit concept is basic to calculus, and the major developments of calculus were discovered in the latter part of the seventeenth century, primarily by Isaac Newton and Gottfried Leibnitz. However, it was not until 200 years later that Augustus Cauchy, based on work of Karl Weierstrass, first expressed the limit concept in the form we now use.

We say that a function f defined on a set X of real numbers has the **limit** L at x_0, written $\lim_{x \to x_0} f(x) = L$, if, given any real number $\varepsilon > 0$, there exists a real number $\delta > 0$ such that $|f(x) - L| < \varepsilon$ whenever $0 < |x - x_0| < \delta$. This definition ensures that values of the function will be close to L whenever x is sufficiently close to x_0. (See Figure 1.1.)

Figure 1.1



A function is said to be continuous at a number in its domain when the limit at the number agrees with the value of the function at the number. So a function f is **continuous** at x_0 if $\lim_{x \to x_0} f(x) = f(x_0)$.

A function f is **continuous on the set** X if it is continuous at each number in X. We use $C(X)$ to denote the set of all functions that are continuous on X. When X is an interval of the real line, the parentheses in this notation are omitted. For example, the set of all functions that are continuous on the closed interval $[a, b]$ is denoted $C[a, b]$.

The limit of a sequence of real or complex numbers is defined in a similar manner. An infinite sequence $\{x_n\}_{n=1}^{\infty}$ **converges** to a number x if, given any $\varepsilon > 0$, there exists a positive integer $N(\varepsilon)$ such that $|x_n - x| < \varepsilon$ whenever $n > N(\varepsilon)$. The notation $\lim_{n \to \infty} x_n = x$, or $x_n \to x$ as $n \to \infty$, means that the sequence $\{x_n\}_{n=1}^{\infty}$ converges to x.

Continuity and Sequence Convergence

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

a. f is continuous at x_0.

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

$$\lim_{n \to \infty} f(x_n) = f(x_0).$$

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

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

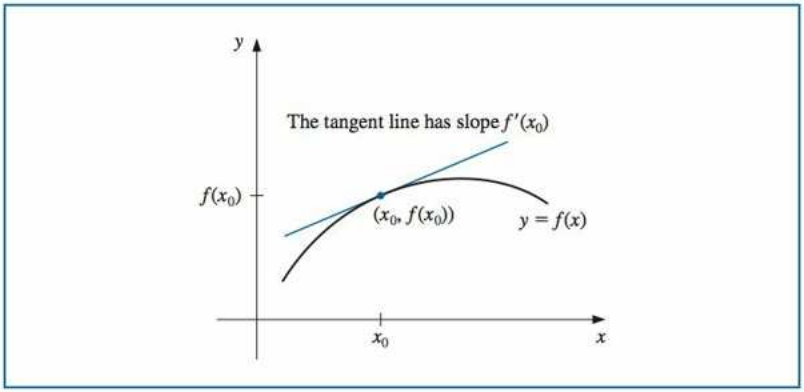
Differentiability

If f is a function defined in an open interval containing x_0, then f is **differentiable** at x_0 when

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

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

Figure 1.2



A function that has a derivative at each number in a set X is **differentiable** on X. Differentiability is a stronger condition on a function than continuity in the following sense.

Differentiability Implies Continuity

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

The set of all functions that have n continuous derivatives on X is denoted $C^n(X)$, and the set of functions that have derivatives of all orders on X is denoted $C^\infty(X)$. Polynomial, rational, trigonometric, exponential, and logarithmic functions are in $C^\infty(X)$, where X consists of all numbers at which the function is defined.

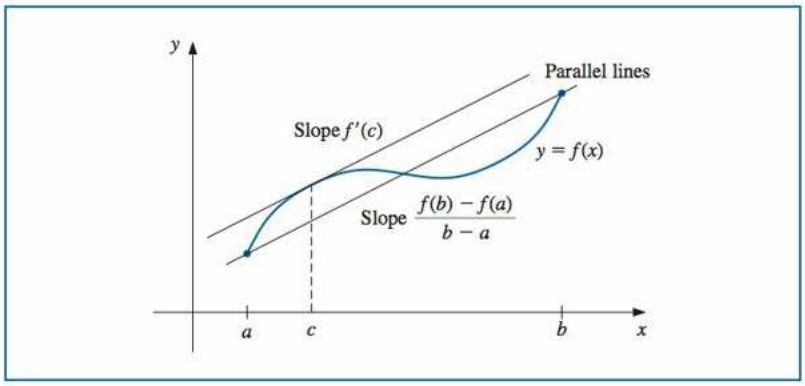
The next results are of fundamental importance in deriving methods for error estimation. The proofs of most of these can be found in any standard calculus text.

Mean Value Theorem

If $f \in C[a, b]$ and f is differentiable on (a, b), then a number c in (a, b) exists such that (see Figure 1.3)

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

Figure 1.3



The following result is frequently used to determine bounds for error formulas.

Extreme Value Theorem

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

The values where a continuous function has its derivative 0 or where the derivative does not exist are called *critical points* of the function. So the Extreme Value Theorem states that a maximum or minimum value of a continuously differentiable function on a closed interval can occur only at the critical points or the endpoints.

Our first example gives some illustrations of applications of the Extreme Value Theorem and MATLAB.

Example 1 Use MATLAB to find the absolute minimum and absolute maximum values of

$$f(x) = 5 \cos 2x - 2x \sin 2x$$

on the intervals **(a)** $[1, 2]$, and **(b)** $[0.5, 1]$.

Solution The solution to this problem is one that is commonly needed in calculus. It provides a good example for illustrating some commonly used commands in MATLAB and the response to the commands that MATLAB gives. In our presentations of MATLAB material, input statements appear left-justified using a `typewriter-like` font. To add emphasis to the responses from MATLAB, these appear centered and in cyan type.

For better readability, we will delete the ≫ symbols needed for input statements as well as the blank lines from MATLAB responses. Other than these changes, the statements will agree with that of MATLAB.

The following command defines $f(x) = 5 \cos 2x - 2x \sin 2x$ as a function of x.

```
f = inline('5*cos(2*x)-2*x*sin(2*x)','x')
```

and MATLAB responds with (actually, the response is on two separate lines, but we will compress the MATLAB responses, here and throughout)

$$\text{Inline function: } f(x) = 5 * cos(2 * x) - 2 * x * sin(2 * x)$$

We have now defined our base function $f(x)$. The x in the command indicates that x is the argument of the function f.

To find the absolute minimum and maximum values of $f(x)$ on the given intervals, we also need its derivative $f'(x)$, which is

$$f'(x) = -12 \sin 2x - 4x \cos 2x.$$

Then we define the function $fp(x) \equiv f'(x)$ in MATLAB to represent the derivative with the inline command

```
fp = inline('-12*sin(2*x)-4*x*cos(2*x)','x')
```

By default, MATLAB displays only a five-digit result, as illustrated by the following command which computes $f(0.5)$:

```
f(0.5)
```

The result from MATLAB is

$$ans = 1.8600$$

We can increase the number of digits of display with the command

```
format long
```

Then the command

```
f(0.5)
```

produces

$$ans = 1.860040544532602$$

We will use this extended precision version of MATLAB output in the remainder of the text.

(a) The absolute minimum and maximum of the continuously differentiable function f occur only at the endpoints of the interval $[1, 2]$ or at a critical point within this interval. We obtain the values at the endpoints with

```
f(1),f(2)
```

and MATLAB responds with

$$ans = -3.899329036387075, \quad ans = -0.241008123086347$$

To determine critical points of the function f, we need to find zeros of $f'(x)$. For this we use the `fzero` command in MATLAB:

```
p =fzero(fp,[1,2])
```

and MATLAB responds with

$$p = 1.358229873843064$$

Evaluating f at this single critical point with

```
f(p)
```

gives

$$ans = -5.675301337592883$$

In summary, the absolute minimum and absolute maximum values of $f(x)$ on the interval $[1, 2]$ are approximately

$$f(1.358229873843064) = -5.675301337592883 \quad \text{and} \quad f(2) = -0.241008123086347.$$

(b) When the interval is $[0.5, 1]$ we have the values at the endpoints given by

$$f(0.5) = 5\cos 1 - 1\sin 1 = 1.860040544532602 \quad \text{and}$$
$$f(1) = 5\cos 2 - 2\sin 2 = -3.899329036387075.$$

However, when we attempt to determine critical points in the interval $[0.5, 1]$ with the command

```
p1 = fzero(fp,[0.5 1])
```

MATLAB returns the response

??? Error using ==> fzero at 293

This indicates that MATLAB could not find a solution to this equation, which is the correct response because f is strictly decreasing on $[0.5, 1]$ and no solution exists. Hence the approximate absolute minimum and absolute maximum values on the interval $[0.5, 1]$ are

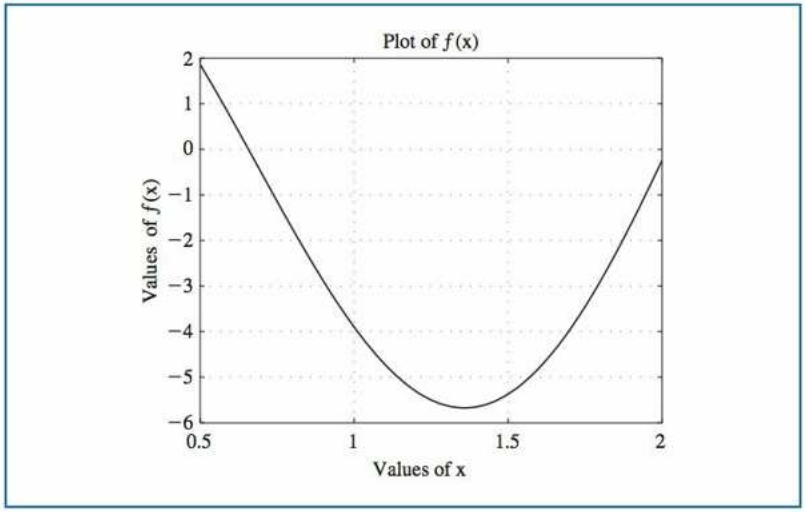
$$f(1) = -3.899329036387075 \quad \text{and} \quad f(0.5) = 1.860040544532602. \qquad \blacksquare$$

The following five commands plot the function on the interval $[0.5, 2]$ with titles for the graph and axes on a grid.

```
fplot(f,[0.5 2])
title('Plot of f(x)')
xlabel('Values of x')
ylabel('Values of f(x)')
grid
```

Figure 1.4 shows the screen that results from these commands. They confirm the results we obtained in Example 1. The graph is displayed in a window that can be saved in a variety of forms for use in technical presentations.

Figure 1.4

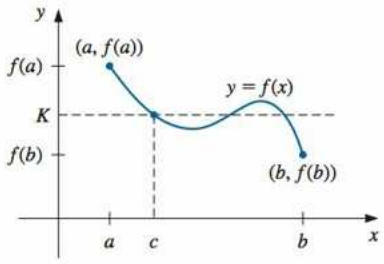


The next result is the Intermediate Value Theorem. Although its statement is not difficult, the proof is beyond the scope of the usual calculus course.

Intermediate Value Theorem

If $f \in C[a, b]$ and K is any number between $f(a)$ and $f(b)$, then there exists a number c in (a, b) for which $f(c) = K$. (Figure 1.5 shows one of the three possibilities for this function and interval.)

Figure 1.5



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

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

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

The Intermediate Value Theorem implies that a number x exists in $(0, 1)$ with $x^5 - 2x^3 + 3x^2 - 1 = 0$. ■

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

Integration

The integral is the other basic concept of calculus. The **Riemann integral** of the function f on the interval $[a, b]$ is the following limit, provided it exists:

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

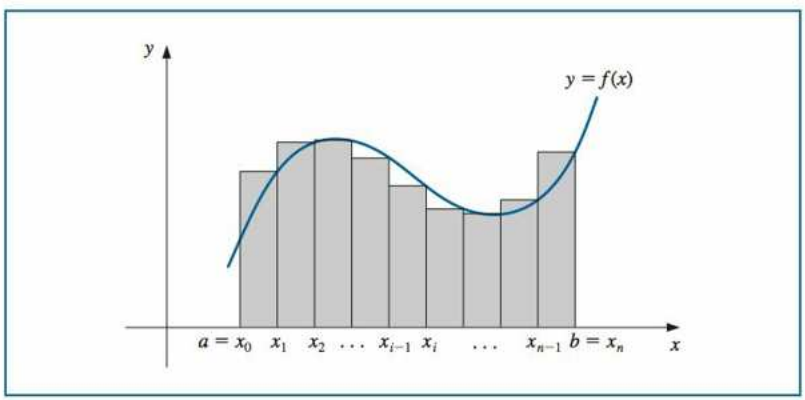
where the numbers $x_0, x_1, \ldots, x_n$ satisfy $a = x_0 < x_1 < \cdots < x_n = b$ and where $\Delta x_i = x_i - x_{i-1}$, for each $i = 1, 2, \ldots, n$, and z_i is arbitrarily chosen in the interval $[x_{i-1}, x_i]$.

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

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

where the numbers shown in Figure 1.6 as x_i are $x_i = a + (i(b-a)/n)$.

Figure 1.6



Two more basic results are needed in our study of numerical methods. The first is a generalization of the usual Mean Value Theorem for Integrals.

Mean Value Theorem for Integrals

If $f \in C[a, b]$, g is integrable on $[a, b]$, and $g(x)$ does not change sign on $[a, b]$, then there exists a number c in (a, b) with

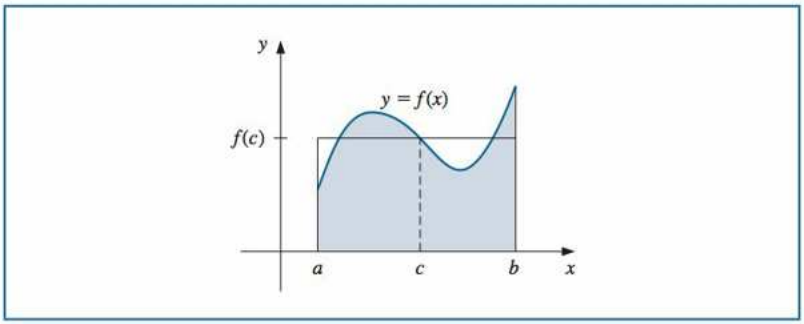
$$\int_a^b f(x)g(x)\,dx = f(c)\int_a^b g(x)\,dx.$$

When $g(x) \equiv 1$, this result reduces to the usual Mean Value Theorem for Integrals. It gives the **average value** of the function f over the interval $[a, b]$ as

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

(See Figure 1.7.)

Figure 1.7



Taylor Polynomials and Series

The final result in this review from calculus describes the development of the Taylor polynomials. The importance of the Taylor polynomials to the study of numerical analysis cannot be overemphasized, and the following result is used repeatedly.

Taylor's Theorem

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

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

where

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

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

and

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

Here $P_n(x)$ is called the ***n*th Taylor polynomial** for f about x_0, and $R_n(x)$ is called the **truncation error** (or *remainder term*) associated with $P_n(x)$. The number $\xi(x)$ in the truncation error $R_n(x)$ depends on the value of x at which the polynomial $P_n(x)$ is being evaluated, so it is actually a function of the variable x. However, we should not expect to

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

be able to explicitly determine the function $\xi(x)$. Taylor's Theorem simply ensures that such a function exists, and that its value lies between x and x_0. In fact, one of the common problems in numerical methods is to try to determine a realistic bound for the value of $f^{(n+1)}(\xi(x))$ for values of x within some specified interval.

The infinite series obtained by taking the limit of $P_n(x)$ as $n \to \infty$ is called the *Taylor series* for f about x_0. The term *truncation error* in the Taylor polynomial refers to the error involved in using a truncated (that is, finite) summation to approximate the sum of an infinite series.

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

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

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

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

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

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

so

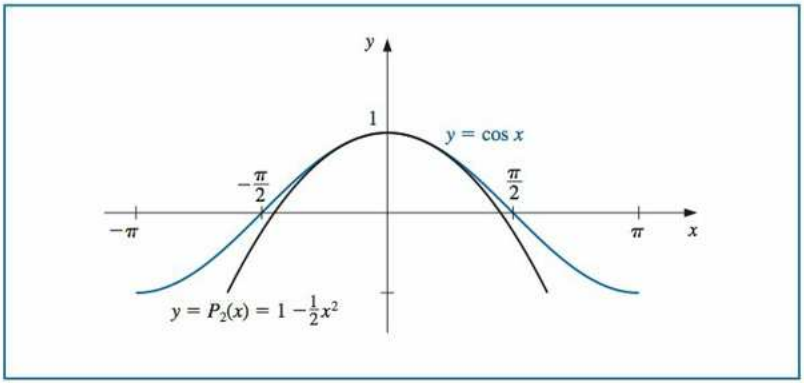
$$f(0) = 1, \quad f'(0) = 0, \quad f''(0) = -1, \quad \text{and} \quad f'''(0) = 0.$$

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

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

where $\xi(x)$ is some (generally unknown) number between 0 and x. (See Figure 1.8.)

Figure 1.8



When $x = 0.01$, this becomes

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

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

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

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

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

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

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

The error bound is much larger than the actual error. This is due in part to the poor bound we used for $|\sin\xi(x)|$. It is shown in Exercise 16 that for all values of x, we have $|\sin x| \leq |x|$. Since $0 \leq \xi < 0.01$, we could have used the fact that $|\sin\xi(x)| \leq 0.01$ in the error formula, producing the bound $0.1\overline{6} \times 10^{-8}$.

(b) Since $f'''(0) = 0$, the third Taylor polynomial with remainder term about $x_0 = 0$ has no x^3 term. It is

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

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

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

So

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

and

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

■

Example 3 illustrates the two basic objectives of numerical methods:

- Find an approximation to the solution of a given problem.
- Determine a bound for the accuracy of the approximation.

The second and third Taylor polynomials gave the same result for the first objective, but the third Taylor polynomial gave a much better result for the second objective.

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

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

Therefore

$$\int_0^{0.1} \cos x \, dx \approx 0.1 - \frac{1}{6}(0.1)^3 = 0.09983\overline{3}.$$

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

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

The true value of this integral is

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

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

MATLAB can be used to obtain these results by first defining $f(x) = \cos x$ and the second Taylor polynomial $T2(x) \equiv T_2(x) = 1 - \frac{1}{2}x^2$ with

```
f = inline('cos(x)','x')
T2 = inline('1-0.5.*x.^ 2','x')
```

The next commands evaluate f at 0.01, $T2$ at 0.01 and compute the error in approximating cos(0.01) with the $T_2(0.01)$.

```
y1 = f(0.01), y2 = T2(0.01), err1 = abs(y1-y2)
```

giving

$$y1 = 0.999950000416665, \quad y2 = 0.999950000000000,$$
$$err1 = 4.166652578518892e - 010$$

To obtain a graph similar to Figure 1.8 requires creating an `M-file` which can load more than one command at a time. We need this in order to define both $f(x)$ and $T2(x)$ if we want to plot them both on the same graph.

An `M-file` is created by selecting `File` on the MATLAB toolbar. Then select `New` and `Script`. The three statements are entered and the result is saved as a file named `ourfunction1`. The `M-file` consists of the following commands:

```
function Y = ourfunction1(x)
Y(:,1)=cos(x(:));
Y(:,2)=1-0.5.*x(:).^2;
```

From the worksheet, we need to enter the following command to create a reference to the function `M-file`:

```
fh = @ourfunction1
```

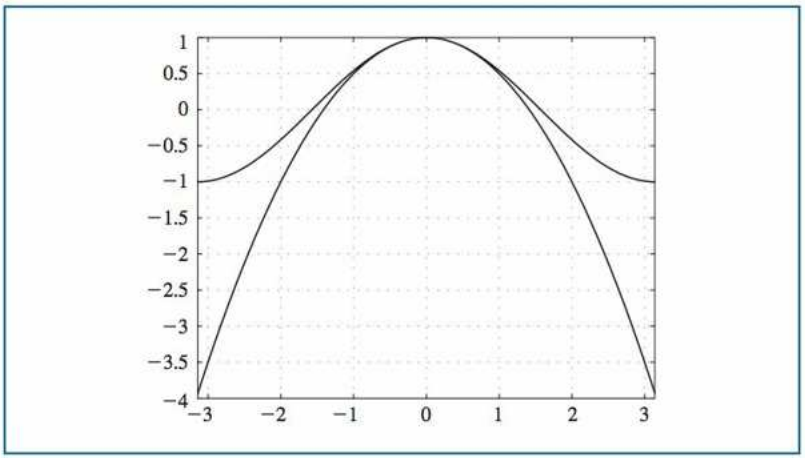
The response from MATLAB is

$$fh = @ourfunction1$$

The graph shown in Figure 1.9 can then be created with the commands

```
fplot(fh,[-pi pi])
grid
```

Figure 1.9



We can also compute the integrals of $f(x)$ and the Taylor polynomial $T_2(x)$ on the interval [0, 0.1] using MATLAB. We use the commands

```
q1 = quad('cos(x)',0,0.1)
q2 = quad('1-0.5*x.^2',0,0.1)
```

and MATLAB produces

$$q1 = 0.099833416646828, \quad q2 = 0.099833333333333$$

EXERCISE SET 1.2

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

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

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

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

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

2. Find intervals containing solutions to the following equations.

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

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

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

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

3. Show that the first derivatives of the following functions are zero at least once in the given intervals.

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

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

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

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

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

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

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

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

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

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

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

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

c. Repeat (a) with $x_0 = 1$.

d. Repeat (b) for the polynomial found in (c).

6. Let $f(x) = \sqrt{x+1}$.

a. Find the third Taylor polynomial $P_3(x)$ about $x_0 = 0$.

b. Use $P_3(x)$ to approximate $\sqrt{0.5}$, $\sqrt{0.75}$, $\sqrt{1.25}$, and $\sqrt{1.5}$.

c. Determine the actual error of the approximations in (b).

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

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

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

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

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

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

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

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

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

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

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

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

11. Let $f(x) = e^{x/2}\sin(x/3)$. Use MATLAB to determine the following.

a. The third Maclaurin polynomial $P_3(x)$.

b. A bound for the error $|f(x) - P_3(x)|$ on $[0, 1]$.

12. Let $f(x) = \ln(x^2 + 2)$. Use MATLAB to determine the following.

a. The Taylor polynomial $P_3(x)$ for f expanded about $x_0 = 1$.

b. The maximum error $|f(x) - P_3(x)|$ for $0 \le x \le 1$.

c. The Maclaurin polynomial $\tilde{P}_3(x)$ for f.

d. The maximum error $|f(x) - \tilde{P}_3(x)|$ for $0 \le x \le 1$.

e. Does $P_3(0)$ approximate $f(0)$ better than $\tilde{P}_3(1)$ approximates $f(1)$?

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

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

a. Explain why this description is accurate.

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

15. The *error function* defined by

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

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

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

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

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

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

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

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

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

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

16. In Example 3 it is stated that x we have $|\sin x| \le |x|$. Use the following to verify this statement.

a. Show that for all $x \ge 0$ we have $f(x) = x - \sin x$ is non-decreasing, which implies that $\sin x \le x$ with equality only when $x = 0$.

b. Reach the conclusion by using the fact that for all values of x, $\sin(-x) = -\sin x$.

1.3 Round-Off Error and Computer Arithmetic

The arithmetic performed by a calculator or computer is different from the arithmetic that we use in our algebra and calculus courses. From your past experience, you might expect that we always have as true statements such things as $2+2 = 4$, $4 \cdot 8 = 32$, and $(\sqrt{3})^2 = 3$.

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

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

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

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

Binary Machine Numbers

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

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

The normalized form for the nonzero double precision numbers has $0 < c < 2^{11} - 1 = 2047$. Since c is positive, a bias of 1023 is subtracted from c to give an actual exponent in the interval $(-1023, 1024)$. This permits adequate representation of numbers with both large and small magnitude. The first bit of the fractional part of a number is assumed to be 1 and is not stored in order to give one additional bit of precision to the representation, Since 53 binary digits correspond to between 15 and 16 decimal digits, we can assume that a number represented using this system has at least 15 decimal digits of precision. Thus, numbers represented in normalized double precision have the form

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

Illustration Consider the machine number

0 10000000011 1011100100010000000000000000000000000000000000000000000000.

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

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

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

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

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

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

However, the next smallest machine number is

0 10000000011 1011100100001111111111111111111111111111111111111111,

and the next largest machine number is

0 10000000011 1011100100010000000000000000000000000000000000000001.

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

$$\begin{aligned}&[27.566406249999998223643160599749535322189331054687 5,\\ &27.566406250000001776356839400250464677810668945312 5).\end{aligned}$$ □

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

$$2^{-1022} \cdot (1 + 0) \approx 0.225 \times 10^{-307}.$$

The largest normalized positive number that can be represented has $s = 0$, $c = 2046$, and $f = 1 - 2^{-52}$, and is equivalent to the decimal number

$$2^{1023} \cdot (1 + (1 - 2^{-52})) \approx 0.17977 \times 10^{309}.$$

Numbers occurring in calculations that have too small a magnitude to be represented result in **underflow**, and are generally set to 0 with computations continuing. However, numbers occurring in calculations that have too large a magnitude to be represented result in **overflow** and typically cause the computations to stop. Note that there are two representations for the number zero; a positive 0 when $s = 0$, $c = 0$, and $f = 0$ and a negative 0 when $s = 1$, $c = 0$, and $f = 0$.

Decimal Machine Numbers

The use of binary digits tends to complicate the computational problems that occur when a finite collection of machine numbers is used to represent all the real numbers. To examine

these problems, we now assume, for simplicity, that machine numbers are represented in the normalized *decimal* form

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

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

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

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

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

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

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

The other method of terminating the mantissa of y at k decimal points is called **rounding**. If the $k+1$st digit is smaller than 5, then the result is the same as chopping. If the $k+1$st digit is 5 or greater, then 1 is added to the kth digit and the resulting number is chopped. As a consequence, rounding can be accomplished by simply adding $5 \times 10^{n-(k+1)}$ to y and then chopping the result to obtain $fl(y)$. Note that when rounding up, the exponent n could increase by 1. In summary, when rounding we add one to d_k to obtain $fl(y)$ whenever $d_{k+1} \ge 5$, that is, we round up; when $d_{k+1} < 5$, we chop off all but the first k digits, so we round down.

The next examples illustrate floating-point arithmetic when the number of digits being retained is quite small. Although the floating-point arithmetic that is performed on a calculator or computer will retain many more digits, the problems this arithmetic can cause are essentially the same regardless of the number of digits. Retaining more digits simply postpones the awareness of the situation.

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

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

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

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

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

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

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

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

There are two common methods for measuring approximation errors.

The approximation p^* to p has **absolute error** $|p - p^*|$ and **relative error** $|p - p^*|/|p|$, provided that $p \ne 0$.

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

(a) $p = 3.000$ and $p^* = 3.100$;

(b) $p = 0.003000$ and $p^* = 0.003100$;

(c) $p = 3000$ and $p^* = 3100$.

Solution First we need to write these numbers in standard floating-point form:

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

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

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

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

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

Finite-Digit Arithmetic

The arithmetic operations of addition, subtraction, multiplication, and division performed by a computer on floating-point numbers also introduce error. These arithmetic operations involve manipulating binary digits by various shifting and logical operations, but the actual mechanics of the arithmetic are not pertinent to our discussion. To illustrate the problems that can occur, we simulate this *finite-digit arithmetic* by first performing, at each stage in a calculation, the appropriate operation using exact arithmetic on the floating-point representations of the numbers. We then convert the result to decimal machine-number representation. The most common round-off error producing arithmetic operation involves the subtraction of nearly equal numbers.

Example 3 Use four-digit decimal chopping arithmetic to simulate the problem of performing the computer operation $\pi - \frac{22}{7}$.

Solution The floating-point representations of these numbers are

$$fl(\pi) = 0.3141 \times 10^1 \qquad \text{and} \qquad fl\left(\frac{22}{7}\right) = 0.3142 \times 10^1.$$

Performing the exact arithmetic on the floating-point numbers gives

$$fl(\pi) - fl\left(\frac{22}{7}\right) = -0.0001 \times 10^1,$$

which converts to the floating-point approximation of this calculation:

$$p^* = fl\left(fl(\pi) - fl\left(\frac{22}{7}\right)\right) = -0.1000 \times 10^{-2}.$$

Although the relative errors using the floating-point representations for π and $\frac{22}{7}$ are small,

$$\left|\frac{\pi - fl(\pi)}{\pi}\right| \le 0.0002 \quad \text{and} \quad \left|\frac{\frac{22}{7} - fl\left(\frac{22}{7}\right)}{\frac{22}{7}}\right| \le 0.0003,$$

the relative error produced by subtracting the nearly equal numbers π and $\frac{22}{7}$ is about 700 times as large:

$$\left|\frac{\left(\pi - \frac{22}{7}\right) - p^*}{\left(\pi - \frac{22}{7}\right)}\right| \approx 0.2092.$$

■

Rounding arithmetic in MATLAB involves the use of the function `round`. This function rounds to the nearest whole number and the function `fix` chops off the fractional part. Suppose we define the numbers $x1$, $x2$, $x3$, and $x4$ with the MATLAB command

```
x1=-123.4,x2=-123.5,x3=123.4,x4=123.5
```

MATLAB responds with,

$$x1 = -1.2340000000000000e + 002$$
$$x2 = -1.2350000000000000e + 002$$
$$x3 = 1.2340000000000000e + 002$$
$$x4 = 1.2350000000000000e + 002$$

Invoking the commands `round` and `fix` gives, for `round`,

$$ans = -123, \quad ans = -124, \quad ans = 123, \quad ans = 124$$

and, for `fix`,

$$ans = -123, \quad ans = -123, \quad ans = 123, \quad ans = 123$$

Exercise 12 illustrates how these functions can be used to perform rounding and chopping arithmetic to a specific number of digits.

EXERCISE SET 1.3

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

a. $p = \pi$, $p^* = \frac{22}{7}$

b. $p = \pi$, $p^* = 3.1416$

c. $p = e$, $p^* = 2.718$

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

e. $p = e^{10}$, $p^* = 22000$

f. $p = 10^{\pi}$, $p^* = 1400$

g. $p = 8!$, $p^* = 39900$

h. $p = 9!$, $p^* = \sqrt{18\pi}\,(9/e)^9$

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

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

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

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

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

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

a. $133 + 0.921$

b. $133 - 0.499$

c. $(121 - 0.327) - 119$

d. $(121 - 119) - 0.327$

e. $\dfrac{\frac{13}{14} - \frac{6}{7}}{2e - 5.4}$

f. $-10\pi + 6e - \frac{3}{62}$

g. $\left(\frac{2}{9}\right) \cdot \left(\frac{9}{7}\right)$

h. $\dfrac{\pi - \frac{22}{7}}{\frac{1}{17}}$

4. Repeat Exercise 3 using three-digit chopping arithmetic.

5. Repeat Exercise 3 using four-digit rounding arithmetic.

6. Repeat Exercise 3 using four-digit chopping arithmetic.

7. The first three nonzero terms of the Maclaurin series for the arctan x are $x - \frac{1}{3}x^3 + \frac{1}{5}x^5$. Compute the absolute error and relative error in the following approximations of π using the polynomial in place of the arctan x:

   a. $4\left[\arctan\left(\frac{1}{2}\right) + \arctan\left(\frac{1}{3}\right)\right]$ **b.** $16\arctan\left(\frac{1}{5}\right) - 4\arctan\left(\frac{1}{239}\right)$

8. The two-by-two linear system

$$ax + by = e,$$
$$cx + dy = f,$$

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

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

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

   a. $1.130x - 6.990y = 14.20$
   $8.110x + 12.20y = -0.1370$

   b. $1.013x - 6.099y = 14.22$
   $-18.11x + 112.2y = -0.1376$

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

$$x = \frac{x_0y_1 - x_1y_0}{y_1 - y_0} \quad \text{and} \quad x = x_0 - \frac{(x_1 - x_0)y_0}{y_1 - y_0}.$$

   a. Show that both formulas are algebraically correct.

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

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

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

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

11. A rectangular parallelepiped has sides 3 cm, 4 cm, and 5 cm, measured to the nearest centimeter.

    a. What are the best upper and lower bounds for the volume of this parallelepiped?

    b. What are the best upper and lower bounds for the surface area?

12. The following MATLAB M-file rounds or chops a number x to t digits where rnd = 1 for rounding and rnd = 0 for chopping.

```
function [res] = CHIP(rnd,t,x)
% This program is used to round or chop a number x to a specific number t
% of digits.
if x == 0
  w = 0;
else
  ee = fix(log10(abs(x)));
  if abs(x) > 1
     ee = ee + 1;
  end;
  if rnd == 1
     w = round(x*10^(t-ee))*10^(ee-t);
  else
     w = fix(x*10^(t-ee))*10^(ee-t);
  end;
end;
res = w ;
```

Verify that the procedure works for the following values.

a. $x = 124.031, t = 5$ **b.** $x = 124.036, t = 5$

c. $x = -0.00653, t = 2$ **d.** $x = -0.00656, t = 2$

13. The binomial coefficient

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

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

a. Suppose decimal machine numbers are of the form

$$\pm 0.d_1d_2d_3d_4 \times 10^n, \quad \text{with } 1 \le d_1 \le 9,\ 0 \le d_i \le 9, \quad \text{if } i = 2, 3, 4 \quad \text{and} \quad |n| \le 15.$$

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

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

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

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

d. Use the equation in (b) and four-digit chopping arithmetic to compute the number of possible 5-card hands in a 52-card deck. Compute the actual and relative errors.

1.4 Errors in Scientific Computation

In the previous section we saw how computational devices represent and manipulate numbers using finite-digit arithmetic. We now examine how the problems with this arithmetic can compound and look at ways to arrange arithmetic calculations to reduce this inaccuracy.

The loss of accuracy due to round-off error can often be avoided by a careful sequencing of operations or a reformulation of the problem. This is most easily described by considering a common computational problem.

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

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

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

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

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

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

we have

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

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

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

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

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

has the small relative error

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

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

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

which simplifies to an alternate quadratic formula

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

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

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

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

Using Eq. (1.2) gives

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

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

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

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

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

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

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

Nested Arithmetic

Example 1 Evaluate $f(x) = x^3 - 6.1x^2 + 3.2x + 1.5$ at $x = 4.71$ using three-digit arithmetic.

Solution Table 1.1 gives the intermediate results in the calculations.

Table 1.1

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

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

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

First we find

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

Then we use this value of x^2 to find

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

Also,

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

and

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

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

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

and for rounding arithmetic we have

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

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

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

so the relative errors for the three-digit methods are

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

$$\text{Rounding:} \left|\frac{-14.263899 + 13.4}{-14.263899}\right| \approx 0.06. \quad \blacksquare$$

Illustration As an alternative approach, the polynomial $f(x)$ in Example 1 can be written in a **nested** manner as

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

Using three-digit chopping arithmetic now produces

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

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

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

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

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

Characterizing Algorithms

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

One criterion we will impose, whenever possible, is that of **stability**. A method is called **stable** if small changes in the initial data produce correspondingly small changes in the final results. When it is possible to have small changes in the initial data producing large changes in the final results, the method is **unstable**. Some methods are stable only for certain choices of initial data. These methods are called **conditionally stable**. We attempt to characterize stability properties whenever possible.

One of the most important topics affecting the stability of a method is the way in which the round-off error grows as the method is successively applied. Suppose an error with

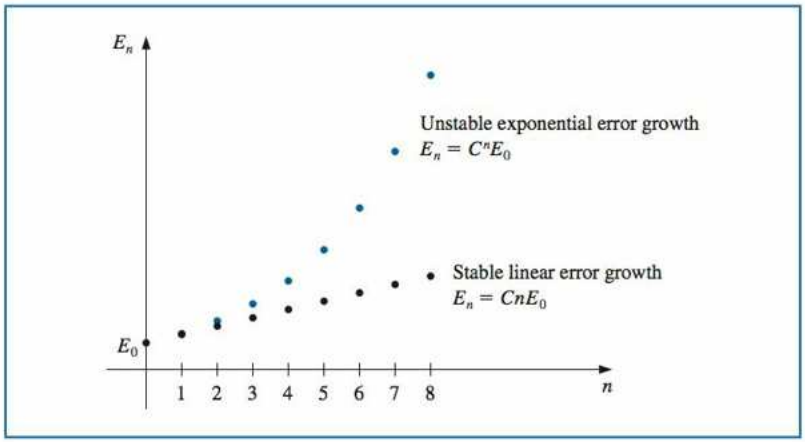
magnitude $E_0 > 0$ is introduced at some stage in the calculations and that the magnitude of the error after n subsequent operations is E_n. There are two distinct cases that often arise in practice.

- If a constant C exists independent of n, with $E_n \approx CnE_0$, the growth of error is **linear**.
- If a constant $C > 1$ exists independent of n, with $E_n \approx C^n E_0$, the growth of error is **exponential**.

It would be unlikely to have $E_n \approx C^n E_0$, with $C < 1$, because this implies that the error tends to zero.

Linear growth of error is usually unavoidable and, when C and E_0 are small, the results are generally acceptable. Methods having exponential growth of error should be avoided because the term C^n becomes large for even relatively small values of n and E_0. Consequently, a method that exhibits linear error growth is stable, while one exhibiting exponential error growth is unstable. (See Figure 1.10.)

Figure 1.10



Rates of Convergence

Iterative techniques often involve sequences, and the section concludes with a brief discussion of some terminology used to describe the rate at which sequences converge when employing a numerical technique. In general, we would like to choose techniques that converge as rapidly as possible. The following definition is used to compare the convergence rates of various methods.

Suppose that $\{\alpha_n\}_{n=1}^{\infty}$ is a sequence that converges to a number α as n becomes large. If positive constants p and K exist with

$$|\alpha - \alpha_n| \leq \frac{K}{n^p}, \quad \text{for all large values of } n,$$

then we say that $\{\alpha_n\}$ **converges to α with rate, or order, of convergence $O(1/n^p)$** (read "big oh of $1/n^p$"). This is indicated by writing $\alpha_n = \alpha + O(1/n^p)$ and stated as "$\alpha_n \to \alpha$

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

with rate of convergence $1/n^p$." We are generally interested in the *largest* value of p for which $\alpha_n = \alpha + O(1/n^p)$.

We also use the "big oh" notation to describe how some divergent sequences grow as n becomes large. If positive constants p and K exist with

$$|\alpha_n| \le Kn^p, \quad \text{for all large values of } n,$$

then we say that $\{\alpha_n\}$ **goes to ∞ with rate, or order, $O(n^p)$**. In the case of a sequence that becomes infinite, we are interested in the *smallest* value of p for which α_n is $O(n^p)$.

The "big oh" definition for sequences can be extended to incorporate more general sequences, but the definition as presented here is sufficient for our purposes.

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

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

Both of the sequences converge to 0, but the sequence $\{\hat{\alpha}_n\}$ converges much faster than the sequence $\{\alpha_n\}$. Using five-digit rounding arithmetic, we have the values shown in Table 1.2. Determine rates of convergence for these two sequences.

Table 1.2

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

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

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

and

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

Hence the rate of convergence of $\{\alpha_n\}$ to 0 is similar to the convergence of $\{1/n\}$ to 0, whereas $\{\hat{\alpha}_n\}$ converges to 0 at a rate similar to the more rapidly convergent sequence $\{1/n^2\}$. We express this by writing

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

The "big oh" notation is also used to describe the rate of convergence of functions, particularly when the independent variable approaches 0.

Suppose that F is a function that converges to a number L as h goes to 0. If positive constants p and K exist with

$$|F(h) - L| \le Kh^p, \qquad \text{as } h \to 0,$$

then **$F(h)$ converges to L with rate, or order, of convergence $O(h^p)$**. This is written as $F(h) = L + O(h^p)$ and stated as "$F(h) \to L$ with rate of convergence h^p."

We are generally interested in the *largest* value of p for which $F(h) = L + O(h^p)$.

EXERCISE SET 1.4

1. (i) Use four-digit rounding arithmetic and Eqs. (1.2) and (1.3) to find the most accurate approximations to the roots of the following quadratic equations. (ii) Compute the absolute errors and relative errors for these approximations.

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

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

2. Repeat Exercise 1 using four-digit chopping arithmetic.

3. Let $f(x) = 1.013x^5 - 5.262x^3 - 0.01732x^2 + 0.8389x - 1.912$.

a. Evaluate $f(2.279)$ by first calculating $(2.279)^2$, $(2.279)^3$, $(2.279)^4$, and $(2.279)^5$ using four-digit rounding arithmetic.

b. Evaluate $f(2.279)$ using the formula

$$f(x) = (((1.013x^2 - 5.262)x - 0.01732)x + 0.8389)x - 1.912$$

and four-digit rounding arithmetic.

c. Compute the absolute and relative errors in (a) and (b).

4. Repeat Exercise 3 using four-digit chopping arithmetic.

5. The fifth Maclaurin polynomials for e^{2x} and e^{-2x} are

$$P_5(x) = \left(\left(\left(\left(\frac{4}{15}x + \frac{2}{3}\right)x + \frac{4}{3}\right)x + 2\right)x + 2\right)x + 1$$

and

$$\hat{P}_5(x) = \left(\left(\left(\left(-\frac{4}{15}x + \frac{2}{3}\right)x - \frac{4}{3}\right)x + 2\right)x - 2\right)x + 1$$

a. Approximate $e^{-0.98}$ using $\hat{P}_5(0.49)$ and four-digit rounding arithmetic.

b. Compute the absolute and relative error for the approximation in (a).

c. Approximate $e^{-0.98}$ using $1/P_5(0.49)$ and four-digit rounding arithmetic.

d. Compute the absolute and relative errors for the approximation in (c).

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

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

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

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

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

7. Use three-digit chopping arithmetic to compute the sum $\sum_{i=1}^{10} 1/i^2$ first by $\frac{1}{1} + \frac{1}{4} + \cdots + \frac{1}{100}$ and then by $\frac{1}{100} + \frac{1}{81} + \cdots + \frac{1}{1}$. Which method is more accurate, and why?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1.5 Computer Software

Computer software packages for approximating the numerical solutions to problems are available in many forms. On our website for the book

`http://www.math.ysu.edu/~faires/Numerical-Methods/`

we have provided programs written in C, FORTRAN, Maple, Mathematica, MATLAB, and Pascal, as well as JAVA applets. These can be used to solve the problems given in the examples and exercises, and will give satisfactory results for most problems that you may need to solve. However, they are what we call *special-purpose* programs. We use this term to distinguish these programs from those available in the standard mathematical subroutine libraries. The programs in these packages will be called *general purpose*.

General Purpose Algorithms

The programs in general-purpose software packages differ in their intent from the algorithms and programs provided with this book. General-purpose software packages consider