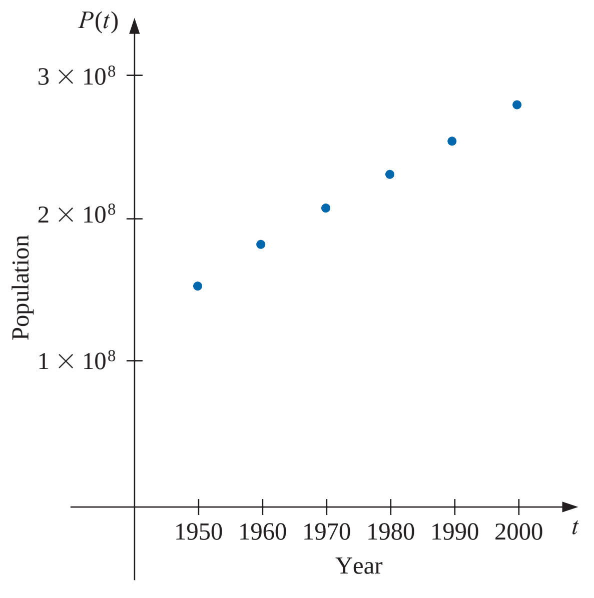
CHAPTER

3 Interpolation and Polynomial Approximation

Introduction

A census of the population of the United States is taken every 10 years. The following table lists the population, in thousands of people, from 1950 to 2000, and the data are also represented in the figure.

Year	1950	1960	1970	1980	1990	2000
Population (in thousands)	151,326	179,323	203,302	226,542	249,633	281,422



In reviewing these data, we might ask whether they could be used to provide a reasonable estimate of the population, say, in 1975 or even in the year 2020. Predictions of this type can be obtained by using a function that fits the given data. This process is called *interpolation* and is the subject of this chapter. This population problem is considered throughout the chapter and in Exercises 18 of Section 3.1, 18 of Section 3.3, and 28 of Section 3.5.

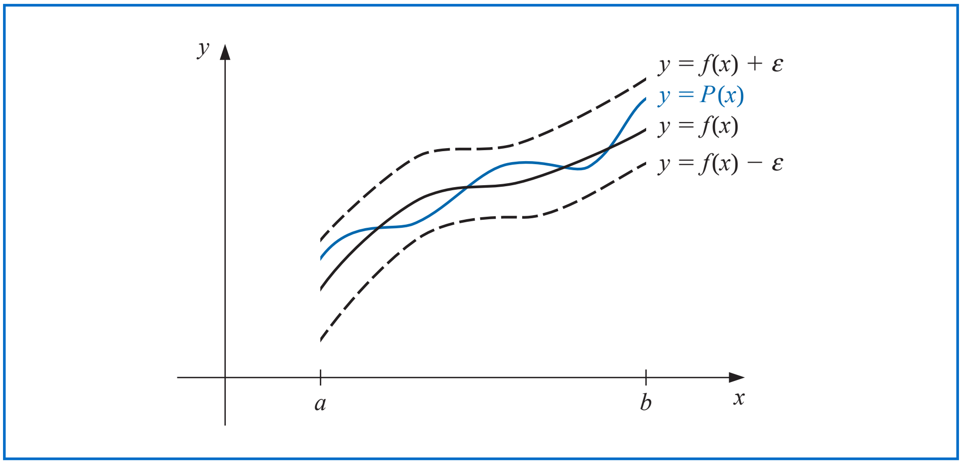
3.1 Interpolation and the Lagrange Polynomial

One of the most useful and well-known classes of functions mapping the set of real numbers into itself is the *algebraic polynomials*, the set of functions of the form

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

where n is a nonnegative integer and $a_0, \ldots, a_n$ are real constants. One reason for their importance is that they uniformly approximate continuous functions. By this we mean that given any function, defined and continuous on a closed and bounded interval, there exists a polynomial that is as "close" to the given function as desired. This result is expressed precisely in the Weierstrass Approximation Theorem. (See Figure 3.1.)

Figure 3.1



Theorem 3.1 **(Weierstrass Approximation Theorem)**

Suppose that f is defined and continuous on $[a, b]$. For each $\epsilon > 0$, there exists a polynomial $P(x)$, with the property that

$$|f(x) - P(x)| < \epsilon, \quad \text{for all } x \text{ in } [a, b]. \qquad ■$$

The proof of this theorem can be found in most elementary texts on real analysis (see, for example, [Bart], pp. 165–172).

Karl Weierstrass (1815–1897) is often referred to as the father of modern analysis because of his insistence on rigor in the demonstration of mathematical results. He was instrumental in developing tests for convergence of series, and determining ways to rigorously define irrational numbers. He was the first to demonstrate that a function could be everywhere continuous but nowhere differentiable, a result that shocked some of his contemporaries.

Another important reason for considering the class of polynomials in the approximation of functions is that the derivative and indefinite integral of a polynomial are easy to determine and are also polynomials. For these reasons, polynomials are often used for approximating continuous functions.

The Taylor polynomials were introduced in Section 1.1, where they were described as one of the fundamental building blocks of numerical analysis. Given this prominence, you might expect that polynomial interpolation would make heavy use of these functions. However this is not the case. The Taylor polynomials agree as closely as possible with a given function at a specific point, but they concentrate their accuracy near that point. A good interpolation polynomial needs to provide a relatively accurate approximation over an entire interval, and Taylor polynomials do not generally do this. For example, suppose we calculate the first six Taylor polynomials about $x_0 = 0$ for $f(x) = e^x$. Since the derivatives of $f(x)$ are all e^x, which evaluated at $x_0 = 0$ gives 1, the Taylor polynomials are

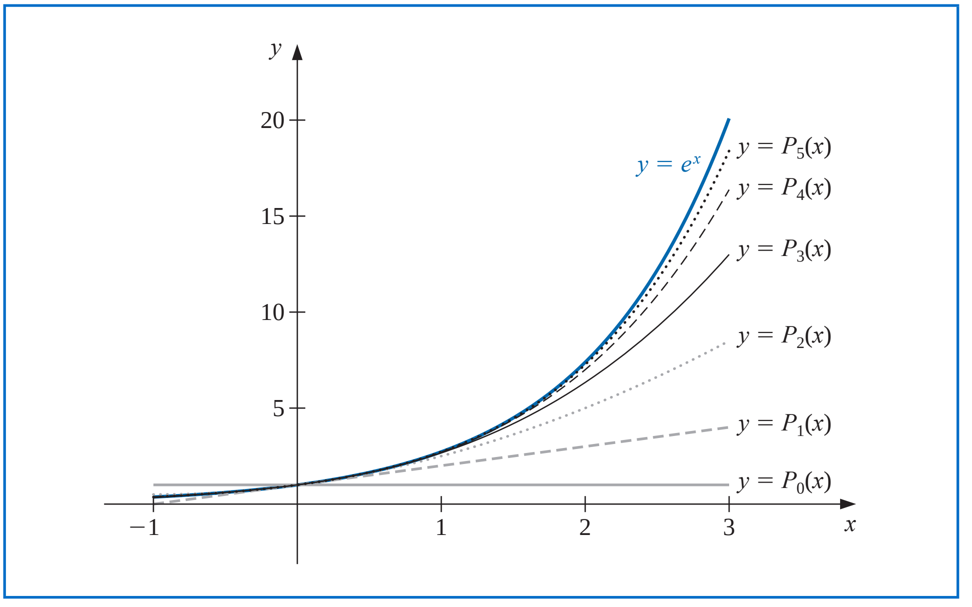
Very little of Weierstrass's work was published during his lifetime, but his lectures, particularly on the theory of functions, had significant influence on an entire generation of students.

$$P_0(x) = 1, \quad P_1(x) = 1 + x, \quad P_2(x) = 1 + x + \frac{x^2}{2}, \quad P_3(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6},$$

$$P_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}, \quad \text{and} \quad P_5(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120}.$$

The graphs of the polynomials are shown in Figure 3.2. (Notice that even for the higher-degree polynomials, the error becomes progressively worse as we move away from zero.)

Figure 3.2



Although better approximations are obtained for $f(x) = e^x$ if higher-degree Taylor polynomials are used, this is not true for all functions. Consider, as an extreme example, using Taylor polynomials of various degrees for $f(x) = 1/x$ expanded about $x_0 = 1$ to approximate $f(3) = 1/3$. Since

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

and, in general,

$$f^{(k)}(x) = (-1)^k k! x^{-k-1},$$

the Taylor polynomials are

$$P_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(1)}{k!}(x-1)^k = \sum_{k=0}^{n} (-1)^k (x-1)^k.$$

To approximate $f(3) = 1/3$ by $P_n(3)$ for increasing values of n, we obtain the values in Table 3.1—rather a dramatic failure! When we approximate $f(3) = 1/3$ by $P_n(3)$ for larger values of n, the approximations become increasingly inaccurate.

Table 3.1

n	0	1	2	3	4	5	6	7
$P_n(3)$	1	−1	3	−5	11	−21	43	−85

For the Taylor polynomials all the information used in the approximation is concentrated at the single number x_0, so these polynomials will generally give inaccurate approximations as we move away from x_0. This limits Taylor polynomial approximation to the situation in which approximations are needed only at numbers close to x_0. For ordinary computational purposes it is more efficient to use methods that include information at various points. We consider this in the remainder of the chapter. The primary use of Taylor polynomials in numerical analysis is not for approximation purposes, but for the derivation of numerical techniques and error estimation.

Lagrange Interpolating Polynomials

The problem of determining a polynomial of degree one that passes through the distinct points (x_0, y_0) and (x_1, y_1) is the same as approximating a function f for which $f(x_0) = y_0$ and $f(x_1) = y_1$ by means of a first-degree polynomial **interpolating**, or agreeing with, the values of f at the given points. Using this polynomial for approximation within the interval given by the endpoints is called polynomial **interpolation**.

Define the functions

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \quad \text{and} \quad L_1(x) = \frac{x - x_0}{x_1 - x_0}.$$

The linear **Lagrange interpolating polynomial** through (x_0, y_0) and (x_1, y_1) is

$$P(x) = L_0(x)f(x_0) + L_1(x)f(x_1) = \frac{x - x_1}{x_0 - x_1}f(x_0) + \frac{x - x_0}{x_1 - x_0}f(x_1).$$

Note that

$$L_0(x_0) = 1, \quad L_0(x_1) = 0, \quad L_1(x_0) = 0, \quad \text{and} \quad L_1(x_1) = 1,$$

which implies that

$$P(x_0) = 1 \cdot f(x_0) + 0 \cdot f(x_1) = f(x_0) = y_0$$

and

$$P(x_1) = 0 \cdot f(x_0) + 1 \cdot f(x_1) = f(x_1) = y_1.$$

So P is the unique polynomial of degree at most one that passes through (x_0, y_0) and (x_1, y_1).

Example 1 Determine the linear Lagrange interpolating polynomial that passes through the points $(2, 4)$ and $(5, 1)$.

Solution In this case we have

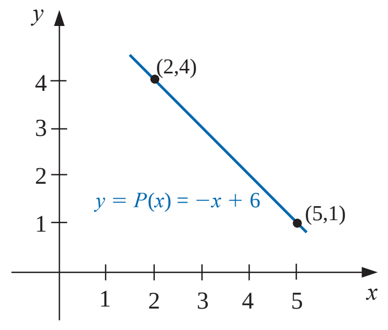
$$L_0(x) = \frac{x - 5}{2 - 5} = -\frac{1}{3}(x - 5) \quad \text{and} \quad L_1(x) = \frac{x - 2}{5 - 2} = \frac{1}{3}(x - 2),$$

so

$$P(x) = -\frac{1}{3}(x - 5) \cdot 4 + \frac{1}{3}(x - 2) \cdot 1 = -\frac{4}{3}x + \frac{20}{3} + \frac{1}{3}x - \frac{2}{3} = -x + 6.$$

The graph of $y = P(x)$ is shown in Figure 3.3. ■

Figure 3.3

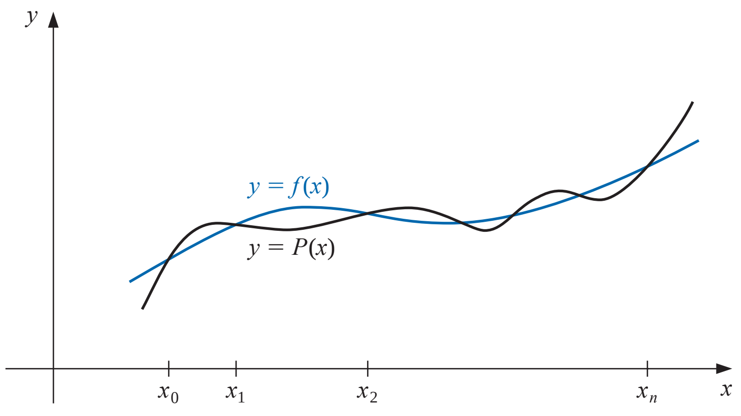


To generalize the concept of linear interpolation, consider the construction of a polynomial of degree at most n that passes through the $n+1$ points

$$(x_0, f(x_0)),\ (x_1, f(x_1)), \ldots, (x_n, f(x_n)).$$

(See Figure 3.4.)

Figure 3.4



In this case we first construct, for each $k = 0, 1, \ldots, n$, a function $L_{n,k}(x)$ with the property that $L_{n,k}(x_i) = 0$ when $i \neq k$ and $L_{n,k}(x_k) = 1$. To satisfy $L_{n,k}(x_i) = 0$ for each $i \neq k$ requires that the numerator of $L_{n,k}(x)$ contain the term

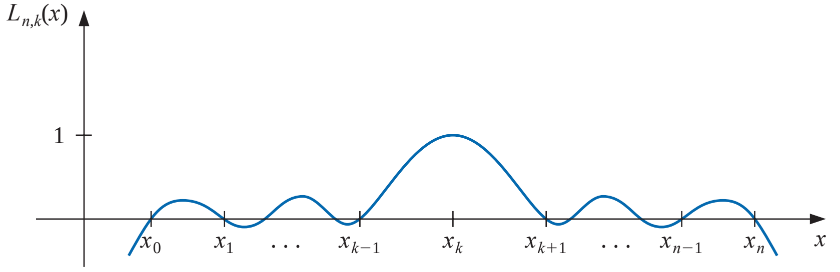
$$(x - x_0)(x - x_1)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n).$$

To satisfy $L_{n,k}(x_k) = 1$, the denominator of $L_{n,k}(x)$ must be this same term but evaluated at $x = x_k$. Thus

$$L_{n,k}(x) = \frac{(x - x_0)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)}{(x_k - x_0)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)}.$$

A sketch of the graph of a typical $L_{n,k}$ (when n is even) is shown in Figure 3.5.

Figure 3.5



The interpolating polynomial is easily described once the form of $L_{n,k}$ is known. This polynomial, called the ***n*th Lagrange interpolating polynomial**, is defined in the following theorem.

Theorem 3.2 If $x_0, x_1, \ldots, x_n$ are $n+1$ distinct numbers and f is a function whose values are given at these numbers, then a unique polynomial $P(x)$ of degree at most n exists with

$$f(x_k) = P(x_k), \quad \text{for each } k = 0, 1, \ldots, n.$$

This polynomial is given by

$$P(x) = f(x_0)L_{n,0}(x) + \cdots + f(x_n)L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x), \tag{3.1}$$

where, for each $k = 0, 1, \ldots, n$,

$$L_{n,k}(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)} \tag{3.2}$$

$$= \prod_{\substack{i=0 \\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k-x_i)}.$$ ■

The interpolation formula named for Joseph Louis Lagrange (1736–1813) was likely known by Isaac Newton around 1675, but it appears to first have been published in 1779 by Edward Waring (1736–1798). Lagrange wrote extensively on the subject of interpolation and his work had significant influence on later mathematicians. He published this result in 1795.

The symbol $\prod$ is used to write products compactly and parallels the symbol $\sum$, which is used for writing sums.

We will write $L_{n,k}(x)$ simply as $L_k(x)$ when there is no confusion as to its degree.

Example 2

(a) Use the numbers (called *nodes*) $x_0 = 2$, $x_1 = 2.75$, and $x_2 = 4$ to find the second Lagrange interpolating polynomial for $f(x) = 1/x$.

(b) Use this polynomial to approximate $f(3) = 1/3$.

Solution **(a)** We first determine the coefficient polynomials $L_0(x)$, $L_1(x)$, and $L_2(x)$. In nested form they are

$$L_0(x) = \frac{(x-2.75)(x-4)}{(2-2.5)(2-4)} = \frac{2}{3}(x-2.75)(x-4),$$

$$L_1(x) = \frac{(x-2)(x-4)}{(2.75-2)(2.75-4)} = -\frac{16}{15}(x-2)(x-4),$$

and

$$L_2(x) = \frac{(x-2)(x-2.75)}{(4-2)(4-2.5)} = \frac{2}{5}(x-2)(x-2.75).$$

Also, $f(x_0) = f(2) = 1/2$, $f(x_1) = f(2.75) = 4/11$, and $f(x_2) = f(4) = 1/4$, so

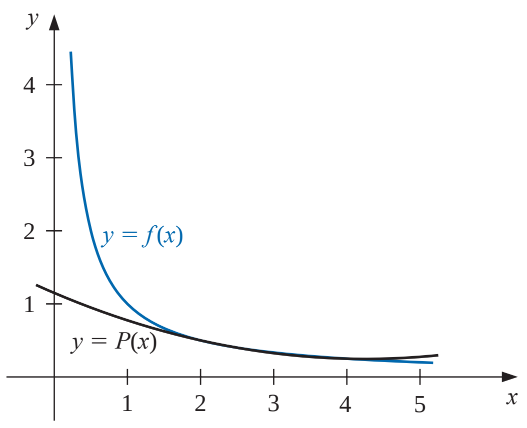
$$\begin{aligned} P(x) &= \sum_{k=0}^{2} f(x_k)L_k(x) \\ &= \frac{1}{3}(x-2.75)(x-4) - \frac{64}{165}(x-2)(x-4) + \frac{1}{10}(x-2)(x-2.75) \\ &= \frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44}. \end{aligned}$$

(b) An approximation to $f(3) = 1/3$ (see Figure 3.6) is

$$f(3) \approx P(3) = \frac{9}{22} - \frac{105}{88} + \frac{49}{44} = \frac{29}{88} \approx 0.32955.$$

Recall that in the opening section of this chapter (see Table 3.1) we found that no Taylor polynomial expanded about $x_0 = 1$ could be used to reasonably approximate $f(x) = 1/x$ at $x = 3$. ■

Figure 3.6



The interpolating polynomial P of degree less than or equal to 3 is defined in Maple with

$P := x \rightarrow interp([2, 11/4, 4], [1/2, 4/11, 1/4], x)$

$$x \rightarrow interp\left(\left[2, \frac{11}{4}, 4\right], \left[\frac{1}{2}, \frac{4}{11}, \frac{1}{4}\right], x\right)$$

To see the polynomial, enter

$P(x)$

$$\frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44}$$

Evaluating $P(3)$ as an approximation to $f(3) = 1/3$, is found with

$evalf(P(3))$

$$0.3295454545$$

The interpolating polynomial can also be defined in Maple using the *CurveFitting* package and the call *PolynomialInterpolation*.

The next step is to calculate a remainder term or bound for the error involved in approximating a function by an interpolating polynomial.

Theorem 3.3 Suppose $x_0, x_1, \ldots, x_n$ are distinct numbers in the interval $[a, b]$ and $f \in C^{n+1}[a, b]$. Then, for each x in $[a, b]$, a number $\xi(x)$ (generally unknown) between $x_0, x_1, \ldots, x_n$, and hence in (a, b), exists with

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_n), \tag{3.3}$$

where $P(x)$ is the interpolating polynomial given in Eq. (3.1). ■

There are other ways that the error term for the Lagrange polynomial can be expressed, but this is the most useful form and the one that most closely agrees with the standard Taylor polynomial error form.

Proof Note first that if $x = x_k$, for any $k = 0, 1, \ldots, n$, then $f(x_k) = P(x_k)$, and choosing $\xi(x_k)$ arbitrarily in (a, b) yields Eq. (3.3).

If $x \neq x_k$, for all $k = 0, 1, \ldots, n$, define the function g for t in $[a, b]$ by

$$\begin{aligned} g(t) &= f(t) - P(t) - [f(x) - P(x)]\frac{(t - x_0)(t - x_1)\cdots(t - x_n)}{(x - x_0)(x - x_1)\cdots(x - x_n)} \\ &= f(t) - P(t) - [f(x) - P(x)]\prod_{i=0}^{n}\frac{(t - x_i)}{(x - x_i)}. \end{aligned}$$

Since $f \in C^{n+1}[a, b]$, and $P \in C^{\infty}[a, b]$, it follows that $g \in C^{n+1}[a, b]$. For $t = x_k$, we have

$$g(x_k) = f(x_k) - P(x_k) - [f(x) - P(x)]\prod_{i=0}^{n}\frac{(x_k - x_i)}{(x - x_i)} = 0 - [f(x) - P(x)] \cdot 0 = 0.$$

Moreover,

$$g(x) = f(x) - P(x) - [f(x) - P(x)]\prod_{i=0}^{n}\frac{(x - x_i)}{(x - x_i)} = f(x) - P(x) - [f(x) - P(x)] = 0.$$

Thus $g \in C^{n+1}[a, b]$, and g is zero at the $n + 2$ distinct numbers $x, x_0, x_1, \ldots, x_n$. By Generalized Rolle's Theorem 1.10, there exists a number ξ in (a, b) for which $g^{(n+1)}(\xi) = 0$. So

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - P^{(n+1)}(\xi) - [f(x) - P(x)]\frac{d^{n+1}}{dt^{n+1}}\left[\prod_{i=0}^{n}\frac{(t - x_i)}{(x - x_i)}\right]_{t=\xi}. \tag{3.4}$$

However $P(x)$ is a polynomial of degree at most n, so the $(n+1)$st derivative, $P^{(n+1)}(x)$, is identically zero. Also, $\prod_{i=0}^{n}[(t - x_i)/(x - x_i)]$ is a polynomial of degree $(n + 1)$, so

$$\prod_{i=0}^{n}\frac{(t - x_i)}{(x - x_i)} = \left[\frac{1}{\prod_{i=0}^{n}(x - x_i)}\right]t^{n+1} + (\text{lower-degree terms in } t),$$

and

$$\frac{d^{n+1}}{dt^{n+1}}\prod_{i=0}^{n}\frac{(t - x_i)}{(x - x_i)} = \frac{(n+1)!}{\prod_{i=0}^{n}(x - x_i)}.$$

Equation (3.4) now becomes

$$0 = f^{(n+1)}(\xi) - 0 - [f(x) - P(x)]\frac{(n+1)!}{\prod_{i=0}^{n}(x - x_i)},$$

and, upon solving for $f(x)$, we have

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x - x_i). \quad \blacksquare\ \blacksquare\ \blacksquare$$

The error formula in Theorem 3.3 is an important theoretical result because Lagrange polynomials are used extensively for deriving numerical differentiation and integration methods. Error bounds for these techniques are obtained from the Lagrange error formula.

Note that the error form for the Lagrange polynomial is quite similar to that for the Taylor polynomial. The nth Taylor polynomial about x_0 concentrates all the known information at x_0 and has an error term of the form

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

The Lagrange polynomial of degree n uses information at the distinct numbers $x_0, x_1, \ldots, x_n$ and, in place of $(x - x_0)^n$, its error formula uses a product of the $n + 1$ terms $(x - x_0)$, $(x - x_1), \ldots, (x - x_n)$:

$$\frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1)\cdots(x - x_n).$$

Example 3 In Example 2 we found the second Lagrange polynomial for $f(x) = 1/x$ on $[2, 4]$ using the nodes $x_0 = 2$, $x_1 = 2.75$, and $x_2 = 4$. Determine the error form for this polynomial, and the maximum error when the polynomial is used to approximate $f(x)$ for $x \in [2, 4]$.

Solution Because $f(x) = x^{-1}$, we have

$$f'(x) = -x^{-2}, \quad f''(x) = 2x^{-3}, \quad \text{and} \quad f'''(x) = -6x^{-4}.$$

As a consequence, the second Lagrange polynomial has the error form

$$\frac{f'''(\xi(x))}{3!}(x-x_0)(x-x_1)(x-x_2) = -(\xi(x))^{-4}(x-2)(x-2.75)(x-4), \quad \text{for } \xi(x) \text{ in } (2, 4).$$

The maximum value of $(\xi(x))^{-4}$ on the interval is $2^{-4} = 1/16$. We now need to determine the maximum value on this interval of the absolute value of the polynomial

$$g(x) = (x - 2)(x - 2.75)(x - 4) = x^3 - \frac{35}{4}x^2 + \frac{49}{2}x - 22.$$

Because

$$D_x\left(x^3 - \frac{35}{4}x^2 + \frac{49}{2}x - 22\right) = 3x^2 - \frac{35}{2}x + \frac{49}{2} = \frac{1}{2}(3x - 7)(2x - 7),$$

the critical points occur at

$$x = \frac{7}{3}, \text{ with } g\left(\frac{7}{3}\right) = \frac{25}{108}, \quad \text{and} \quad x = \frac{7}{2}, \text{ with } g\left(\frac{7}{2}\right) = -\frac{9}{16}.$$

Hence, the maximum error is

$$\frac{f'''(\xi(x))}{3!}|(x - x_0)(x - x_1)(x - x_2)| \le \frac{1}{16 \cdot 6}\left|-\frac{9}{16}\right| = \frac{3}{512} \approx 0.00586. \quad \blacksquare$$

The next example illustrates how the error formula can be used to prepare a table of data that will ensure a specified interpolation error within a specified bound.

Example 4 Suppose a table is to be prepared for the function $f(x) = e^x$, for x in $[0, 1]$. Assume the number of decimal places to be given per entry is $d \geq 8$ and that the difference between adjacent x-values, the step size, is h. What step size h will ensure that linear interpolation gives an absolute error of at most 10^{-6} for all x in $[0, 1]$?

Solution Let $x_0, x_1, \ldots$ be the numbers at which f is evaluated, x be in [0,1], and suppose j satisfies $x_j \leq x \leq x_{j+1}$. Eq. (3.3) implies that the error in linear interpolation is

$$|f(x) - P(x)| = \left|\frac{f^{(2)}(\xi)}{2!}(x - x_j)(x - x_{j+1})\right| = \frac{|f^{(2)}(\xi)|}{2}|(x - x_j)||(x - x_{j+1})|.$$

The step size is h, so $x_j = jh$, $x_{j+1} = (j+1)h$, and

$$|f(x) - P(x)| \leq \frac{|f^{(2)}(\xi)|}{2!}|(x - jh)(x - (j+1)h)|.$$

Hence

$$|f(x) - P(x)| \leq \frac{\max_{\xi \in [0,1]} e^\xi}{2} \max_{x_j \leq x \leq x_{j+1}} |(x - jh)(x - (j+1)h)|$$
$$\leq \frac{e}{2} \max_{x_j \leq x \leq x_{j+1}} |(x - jh)(x - (j+1)h)|.$$

Consider the function $g(x) = (x - jh)(x - (j+1)h)$, for $jh \leq x \leq (j+1)h$. Because

$$g'(x) = (x - (j+1)h) + (x - jh) = 2\left(x - jh - \frac{h}{2}\right),$$

the only critical point for g is at $x = jh + h/2$, with $g(jh + h/2) = (h/2)^2 = h^2/4$.

Since $g(jh) = 0$ and $g((j+1)h) = 0$, the maximum value of $|g'(x)|$ in $[jh, (j+1)h]$ must occur at the critical point which implies that

$$|f(x) - P(x)| \leq \frac{e}{2} \max_{x_j \leq x \leq x_{j+1}} |g(x)| \leq \frac{e}{2} \cdot \frac{h^2}{4} = \frac{eh^2}{8}.$$

Consequently, to ensure that the the error in linear interpolation is bounded by 10^{-6}, it is sufficient for h to be chosen so that

$$\frac{eh^2}{8} \leq 10^{-6}. \quad \text{This implies that} \quad h < 1.72 \times 10^{-3}.$$

Because $n = (1 - 0)/h$ must be an integer, a reasonable choice for the step size is $h = 0.001$. ■

EXERCISE SET 3.1

1. For the given functions $f(x)$, let $x_0 = 0$, $x_1 = 0.6$, and $x_2 = 0.9$. Construct interpolation polynomials of degree at most one and at most two to approximate $f(0.45)$, and find the absolute error.

   a. $f(x) = \cos x$

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

   c. $f(x) = \ln(x + 1)$

   d. $f(x) = \tan x$

2. For the given functions $f(x)$, let $x_0 = 1, x_1 = 1.25$, and $x_2 = 1.6$. Construct interpolation polynomials of degree at most one and at most two to approximate $f(1.4)$, and find the absolute error.

a. $f(x) = \sin \pi x$
b. $f(x) = \sqrt[3]{x-1}$
c. $f(x) = \log_{10}(3x - 1)$
d. $f(x) = e^{2x} - x$

3. Use Theorem 3.3 to find an error bound for the approximations in Exercise 1.

4. Use Theorem 3.3 to find an error bound for the approximations in Exercise 2.

5. Use appropriate Lagrange interpolating polynomials of degrees one, two, and three to approximate each of the following:

a. $f(8.4)$ if $f(8.1) = 16.94410$, $f(8.3) = 17.56492$, $f(8.6) = 18.50515$, $f(8.7) = 18.82091$
b. $f\left(-\frac{1}{3}\right)$ if $f(-0.75) = -0.07181250$, $f(-0.5) = -0.02475000$, $f(-0.25) = 0.33493750$, $f(0) = 1.10100000$
c. $f(0.25)$ if $f(0.1) = 0.62049958$, $f(0.2) = -0.28398668$, $f(0.3) = 0.00660095$, $f(0.4) = 0.24842440$
d. $f(0.9)$ if $f(0.6) = -0.17694460$, $f(0.7) = 0.01375227$, $f(0.8) = 0.22363362$, $f(1.0) = 0.65809197$

6. Use appropriate Lagrange interpolating polynomials of degrees one, two, and three to approximate each of the following:

a. $f(0.43)$ if $f(0) = 1$, $f(0.25) = 1.64872$, $f(0.5) = 2.71828$, $f(0.75) = 4.48169$
b. $f(0)$ if $f(-0.5) = 1.93750$, $f(-0.25) = 1.33203$, $f(0.25) = 0.800781$, $f(0.5) = 0.687500$
c. $f(0.18)$ if $f(0.1) = -0.29004986$, $f(0.2) = -0.56079734$, $f(0.3) = -0.81401972$, $f(0.4) = -1.0526302$
d. $f(0.25)$ if $f(-1) = 0.86199480$, $f(-0.5) = 0.95802009$, $f(0) = 1.0986123$, $f(0.5) = 1.2943767$

7. The data for Exercise 5 were generated using the following functions. Use the error formula to find a bound for the error, and compare the bound to the actual error for the cases $n = 1$ and $n = 2$.

a. $f(x) = x \ln x$
b. $f(x) = x^3 + 4.001x^2 + 4.002x + 1.101$
c. $f(x) = x \cos x - 2x^2 + 3x - 1$
d. $f(x) = \sin(e^x - 2)$

8. The data for Exercise 6 were generated using the following functions. Use the error formula to find a bound for the error, and compare the bound to the actual error for the cases $n = 1$ and $n = 2$.

a. $f(x) = e^{2x}$
b. $f(x) = x^4 - x^3 + x^2 - x + 1$
c. $f(x) = x^2 \cos x - 3x$
d. $f(x) = \ln(e^x + 2)$

9. Let $P_3(x)$ be the interpolating polynomial for the data $(0, 0)$, $(0.5, y)$, $(1, 3)$, and $(2, 2)$. The coefficient of x^3 in $P_3(x)$ is 6. Find y.

10. Let $f(x) = \sqrt{x - x^2}$ and $P_2(x)$ be the interpolation polynomial on $x_0 = 0$, x_1 and $x_2 = 1$. Find the largest value of x_1 in $(0, 1)$ for which $f(0.5) - P_2(0.5) = -0.25$.

11. Use the following values and four-digit rounding arithmetic to construct a third Lagrange polynomial approximation to $f(1.09)$. The function being approximated is $f(x) = \log_{10}(\tan x)$. Use this knowledge to find a bound for the error in the approximation.

$$f(1.00) = 0.1924 \quad f(1.05) = 0.2414 \quad f(1.10) = 0.2933 \quad f(1.15) = 0.3492$$

12. Use the Lagrange interpolating polynomial of degree three or less and four-digit chopping arithmetic to approximate $\cos 0.750$ using the following values. Find an error bound for the approximation.

$$\cos 0.698 = 0.7661 \quad \cos 0.733 = 0.7432 \quad \cos 0.768 = 0.7193 \quad \cos 0.803 = 0.6946$$

The actual value of $\cos 0.750$ is 0.7317 (to four decimal places). Explain the discrepancy between the actual error and the error bound.

13. Construct the Lagrange interpolating polynomials for the following functions, and find a bound for the absolute error on the interval $[x_0, x_n]$.

a. $f(x) = e^{2x} \cos 3x, \quad x_0 = 0, x_1 = 0.3, x_2 = 0.6, n = 2$

b. $f(x) = \sin(\ln x), \quad x_0 = 2.0, x_1 = 2.4, x_2 = 2.6, n = 2$

c. $f(x) = \ln x, \quad x_0 = 1, x_1 = 1.1, x_2 = 1.3, x_3 = 1.4, n = 3$

d. $f(x) = \cos x + \sin x, \quad x_0 = 0, x_1 = 0.25, x_2 = 0.5, x_3 = 1.0, n = 3$

14. Let $f(x) = e^x$, for $0 \le x \le 2$.

a. Approximate $f(0.25)$ using linear interpolation with $x_0 = 0$ and $x_1 = 0.5$.

b. Approximate $f(0.75)$ using linear interpolation with $x_0 = 0.5$ and $x_1 = 1$.

c. Approximate $f(0.25)$ and $f(0.75)$ by using the second interpolating polynomial with $x_0 = 0$, $x_1 = 1$, and $x_2 = 2$.

d. Which approximations are better and why?

15. Repeat Exercise 11 using Maple with *Digits* set to 10.

16. Repeat Exercise 12 using Maple with *Digits* set to 10.

17. Suppose you need to construct eight-decimal-place tables for the common, or base-10, logarithm function from $x = 1$ to $x = 10$ in such a way that linear interpolation is accurate to within 10^{-6}. Determine a bound for the step size for this table. What choice of step size would you make to ensure that $x = 10$ is included in the table?

18. **a.** The introduction to this chapter included a table listing the population of the United States from 1950 to 2000. Use Lagrange interpolation to approximate the population in the years 1940, 1975, and 2020.

b. The population in 1940 was approximately 132,165,000. How accurate do you think your 1975 and 2020 figures are?

19. It is suspected that the high amounts of tannin in mature oak leaves inhibit the growth of the winter moth (*Operophtera bromata L., Geometridae*) larvae that extensively damage these trees in certain years. The following table lists the average weight of two samples of larvae at times in the first 28 days after birth. The first sample was reared on young oak leaves, whereas the second sample was reared on mature leaves from the same tree.

a. Use Lagrange interpolation to approximate the average weight curve for each sample.

b. Find an approximate maximum average weight for each sample by determining the maximum of the interpolating polynomial.

Day	0	6	10	13	17	20	28
Sample 1 average weight (mg)	6.67	17.33	42.67	37.33	30.10	29.31	28.74
Sample 2 average weight (mg)	6.67	16.11	18.89	15.00	10.56	9.44	8.89

20. In Exercise 26 of Section 1.1 a Maclaurin series was integrated to approximate erf(1), where erf(x) is the normal distribution error function defined by

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

a. Use the Maclaurin series to construct a table for erf(x) that is accurate to within 10^{-4} for erf(x_i), where $x_i = 0.2i$, for $i = 0, 1, \ldots, 5$.

b. Use both linear interpolation and quadratic interpolation to obtain an approximation to erf($\frac{1}{3}$). Which approach seems most feasible?

21. Prove Taylor's Theorem 1.14 by following the procedure in the proof of Theorem 3.3. [*Hint:* Let

$$g(t) = f(t) - P(t) - [f(x) - P(x)] \cdot \frac{(t - x_0)^{n+1}}{(x - x_0)^{n+1}},$$

where P is the nth Taylor polynomial, and use the Generalized Rolle's Theorem 1.10.]

22. Show that $\max_{x_j \le x \le x_{j+1}} |g(x)| = h^2/4$, where $g(x) = (x - jh)(x - (j+1)h)$.

23. The Bernstein polynomial of degree n for $f \in C[0, 1]$ is given by

$$B_n(x) = \sum_{k=0}^{n} \binom{n}{k} f\left(\frac{k}{n}\right) x^k (1 - x)^{n-k},$$

where $\binom{n}{k}$ denotes $n!/k!(n - k)!$. These polynomials can be used in a constructive proof of the Weierstrass Approximation Theorem 3.1 (see [Bart]) because $\lim_{n\to\infty} B_n(x) = f(x)$, for each $x \in [0, 1]$.

a. Find $B_3(x)$ for the functions

i. $f(x) = x$ **ii.** $f(x) = 1$

b. Show that for each $k \le n$,

$$\binom{n-1}{k-1} = \left(\frac{k}{n}\right)\binom{n}{k}.$$

c. Use part (b) and the fact, from (ii) in part (a), that

$$1 = \sum_{k=0}^{n} \binom{n}{k} x^k (1 - x)^{n-k}, \quad \text{for each } n,$$

to show that, for $f(x) = x^2$,

$$B_n(x) = \left(\frac{n-1}{n}\right) x^2 + \frac{1}{n} x.$$

d. Use part (c) to estimate the value of n necessary for $\left|B_n(x) - x^2\right| \le 10^{-6}$ to hold for all x in $[0, 1]$.

3.2 Data Approximation and Neville's Method

In the previous section we found an explicit representation for Lagrange polynomials and their error when approximating a function on an interval. A frequent use of these polynomials involves the interpolation of tabulated data. In this case an explicit representation of the polynomial might not be needed, only the values of the polynomial at specified points. In this situation the function underlying the data might not be known so the explicit form of the error cannot be used. We will now illustrate a practical application of interpolation in such a situation.

Illustration

Table 3.2 lists values of a function f at various points. The approximations to $f(1.5)$ obtained by various Lagrange polynomials that use this data will be compared to try and determine the accuracy of the approximation.

Table 3.2

x	$f(x)$
1.0	0.7651977
1.3	0.6200860
1.6	0.4554022
1.9	0.2818186
2.2	0.1103623

The most appropriate linear polynomial uses $x_0 = 1.3$ and $x_1 = 1.6$ because 1.5 is between 1.3 and 1.6. The value of the interpolating polynomial at 1.5 is

$$\begin{aligned} P_1(1.5) &= \frac{(1.5 - 1.6)}{(1.3 - 1.6)} f(1.3) + \frac{(1.5 - 1.3)}{(1.6 - 1.3)} f(1.6) \\ &= \frac{(1.5 - 1.6)}{(1.3 - 1.6)}(0.6200860) + \frac{(1.5 - 1.3)}{(1.6 - 1.3)}(0.4554022) = 0.5102968. \end{aligned}$$

Two polynomials of degree 2 can reasonably be used, one with $x_0 = 1.3$, $x_1 = 1.6$, and $x_2 = 1.9$, which gives

$$P_2(1.5) = \frac{(1.5-1.6)(1.5-1.9)}{(1.3-1.6)(1.3-1.9)}(0.6200860) + \frac{(1.5-1.3)(1.5-1.9)}{(1.6-1.3)(1.6-1.9)}(0.4554022)$$
$$+ \frac{(1.5-1.3)(1.5-1.6)}{(1.9-1.3)(1.9-1.6)}(0.2818186) = 0.5112857,$$

and one with $x_0 = 1.0$, $x_1 = 1.3$, and $x_2 = 1.6$, which gives $\hat{P}_2(1.5) = 0.5124715$.

In the third-degree case, there are also two reasonable choices for the polynomial. One with $x_0 = 1.3$, $x_1 = 1.6$, $x_2 = 1.9$, and $x_3 = 2.2$, which gives $P_3(1.5) = 0.5118302$.

The second third-degree approximation is obtained with $x_0 = 1.0$, $x_1 = 1.3$, $x_2 = 1.6$, and $x_3 = 1.9$, which gives $\hat{P}_3(1.5) = 0.5118127$. The fourth-degree Lagrange polynomial uses all the entries in the table. With $x_0 = 1.0$, $x_1 = 1.3$, $x_2 = 1.6$, $x_3 = 1.9$, and $x_4 = 2.2$, the approximation is $P_4(1.5) = 0.5118200$.

Because $P_3(1.5)$, $\hat{P}_3(1.5)$, and $P_4(1.5)$ all agree to within 2×10^{-5} units, we expect this degree of accuracy for these approximations. We also expect $P_4(1.5)$ to be the most accurate approximation, since it uses more of the given data.

The function we are approximating is actually the Bessel function of the first kind of order zero, whose value at 1.5 is known to be 0.5118277. Therefore, the true accuracies of the approximations are as follows:

$$|P_1(1.5) - f(1.5)| \approx 1.53 \times 10^{-3},$$
$$|P_2(1.5) - f(1.5)| \approx 5.42 \times 10^{-4},$$
$$|\hat{P}_2(1.5) - f(1.5)| \approx 6.44 \times 10^{-4},$$
$$|P_3(1.5) - f(1.5)| \approx 2.5 \times 10^{-6},$$
$$|\hat{P}_3(1.5) - f(1.5)| \approx 1.50 \times 10^{-5},$$
$$|P_4(1.5) - f(1.5)| \approx 7.7 \times 10^{-6}.$$

Although $P_3(1.5)$ is the most accurate approximation, if we had no knowledge of the actual value of $f(1.5)$, we would accept $P_4(1.5)$ as the best approximation since it includes the most data about the function. The Lagrange error term derived in Theorem 3.3 cannot be applied here because we have no knowledge of the fourth derivative of f. Unfortunately, this is generally the case. □

Neville's Method

A practical difficulty with Lagrange interpolation is that the error term is difficult to apply, so the degree of the polynomial needed for the desired accuracy is generally not known until computations have been performed. A common practice is to compute the results given from various polynomials until appropriate agreement is obtained, as was done in the previous Illustration. However, the work done in calculating the approximation by the second polynomial does not lessen the work needed to calculate the third approximation; nor is the fourth approximation easier to obtain once the third approximation is known, and so on. We will now derive these approximating polynomials in a manner that uses the previous calculations to greater advantage.

Definition 3.4 Let f be a function defined at $x_0, x_1, x_2, \ldots, x_n$, and suppose that $m_1, m_2, \ldots, m_k$ are k distinct integers, with $0 \le m_i \le n$ for each i. The Lagrange polynomial that agrees with $f(x)$ at the k points $x_{m_1}, x_{m_2}, \ldots, x_{m_k}$ is denoted $P_{m_1,m_2,\ldots,m_k}(x)$. ■

Example 1 Suppose that $x_0 = 1$, $x_1 = 2$, $x_2 = 3$, $x_3 = 4$, $x_4 = 6$, and $f(x) = e^x$. Determine the interpolating polynomial denoted $P_{1,2,4}(x)$, and use this polynomial to approximate $f(5)$.

Solution This is the Lagrange polynomial that agrees with $f(x)$ at $x_1 = 2$, $x_2 = 3$, and $x_4 = 6$. Hence

$$P_{1,2,4}(x) = \frac{(x-3)(x-6)}{(2-3)(2-6)}e^2 + \frac{(x-2)(x-6)}{(3-2)(3-6)}e^3 + \frac{(x-2)(x-3)}{(6-2)(6-3)}e^6.$$

So

$$\begin{aligned} f(5) \approx P(5) &= \frac{(5-3)(5-6)}{(2-3)(2-6)}e^2 + \frac{(5-2)(5-6)}{(3-2)(3-6)}e^3 + \frac{(5-2)(5-3)}{(6-2)(6-3)}e^6 \\ &= -\frac{1}{2}e^2 + e^3 + \frac{1}{2}e^6 \approx 218.105. \end{aligned}$$

■

The next result describes a method for recursively generating Lagrange polynomial approximations.

Theorem 3.5 Let f be defined at $x_0, x_1, \ldots, x_k$, and let x_j and x_i be two distinct numbers in this set. Then

$$P(x) = \frac{(x-x_j)P_{0,1,\ldots,j-1,j+1,\ldots,k}(x) - (x-x_i)P_{0,1,\ldots,i-1,i+1,\ldots,k}(x)}{(x_i - x_j)}$$

is the kth Lagrange polynomial that interpolates f at the $k+1$ points $x_0, x_1, \ldots, x_k$. ■

Proof For ease of notation, let $Q \equiv P_{0,1,\ldots,i-1,i+1,\ldots,k}$ and $\hat{Q} \equiv P_{0,1,\ldots,j-1,j+1,\ldots,k}$. Since $Q(x)$ and $\hat{Q}(x)$ are polynomials of degree $k-1$ or less, $P(x)$ is of degree at most k.

First note that $\hat{Q}(x_i) = f(x_i)$, implies that

$$P(x_i) = \frac{(x_i - x_j)\hat{Q}(x_i) - (x_i - x_i)Q(x_i)}{x_i - x_j} = \frac{(x_i - x_j)}{(x_i - x_j)} f(x_i) = f(x_i).$$

Similarly, since $Q(x_j) = f(x_j)$, we have $P(x_j) = f(x_j)$.

In addition, if $0 \le r \le k$ and r is neither i nor j, then $Q(x_r) = \hat{Q}(x_r) = f(x_r)$. So

$$P(x_r) = \frac{(x_r - x_j)\hat{Q}(x_r) - (x_r - x_i)Q(x_r)}{x_i - x_j} = \frac{(x_i - x_j)}{(x_i - x_j)} f(x_r) = f(x_r).$$

But, by definition, $P_{0,1,\ldots,k}(x)$ is the unique polynomial of degree at most k that agrees with f at $x_0, x_1, \ldots, x_k$. Thus, $P \equiv P_{0,1,\ldots,k}$. ■ ■ ■

Theorem 3.5 implies that the interpolating polynomials can be generated recursively. For example, we have

$$P_{0,1} = \frac{1}{x_1 - x_0}[(x-x_0)P_1 - (x-x_1)P_0], \qquad P_{1,2} = \frac{1}{x_2 - x_1}[(x-x_1)P_2 - (x-x_2)P_1],$$

$$P_{0,1,2} = \frac{1}{x_2 - x_0}[(x-x_0)P_{1,2} - (x-x_2)P_{0,1}],$$

and so on. They are generated in the manner shown in Table 3.3, where each row is completed before the succeeding rows are begun.

Table 3.3

x_0	P_0				
x_1	P_1	$P_{0,1}$			
x_2	P_2	$P_{1,2}$	$P_{0,1,2}$		
x_3	P_3	$P_{2,3}$	$P_{1,2,3}$	$P_{0,1,2,3}$	
x_4	P_4	$P_{3,4}$	$P_{2,3,4}$	$P_{1,2,3,4}$	$P_{0,1,2,3,4}$

The procedure that uses the result of Theorem 3.5 to recursively generate interpolating polynomial approximations is called **Neville's method**. The P notation used in Table 3.3 is cumbersome because of the number of subscripts used to represent the entries. Note, however, that as an array is being constructed, only two subscripts are needed. Proceeding down the table corresponds to using consecutive points x_i with larger i, and proceeding to the right corresponds to increasing the degree of the interpolating polynomial. Since the points appear consecutively in each entry, we need to describe only a starting point and the number of additional points used in constructing the approximation.

Eric Harold Neville (1889–1961) gave this modification of the Lagrange formula in a paper published in 1932.[N]

To avoid the multiple subscripts, we let $Q_{i,j}(x)$, for $0 \le j \le i$, denote the interpolating polynomial of degree j on the $(j+1)$ numbers $x_{i-j}, x_{i-j+1}, \ldots, x_{i-1}, x_i$; that is,

$$Q_{i,j} = P_{i-j,i-j+1,\ldots,i-1,i}.$$

Using this notation provides the Q notation array in Table 3.4.

Table 3.4

x_0	$P_0 = Q_{0,0}$				
x_1	$P_1 = Q_{1,0}$	$P_{0,1} = Q_{1,1}$			
x_2	$P_2 = Q_{2,0}$	$P_{1,2} = Q_{2,1}$	$P_{0,1,2} = Q_{2,2}$		
x_3	$P_3 = Q_{3,0}$	$P_{2,3} = Q_{3,1}$	$P_{1,2,3} = Q_{3,2}$	$P_{0,1,2,3} = Q_{3,3}$	
x_4	$P_4 = Q_{4,0}$	$P_{3,4} = Q_{4,1}$	$P_{2,3,4} = Q_{4,2}$	$P_{1,2,3,4} = Q_{4,3}$	$P_{0,1,2,3,4} = Q_{4,4}$

Example 2 Values of various interpolating polynomials at $x = 1.5$ were obtained in the Illustration at the beginning of the Section using the data shown in Table 3.5. Apply Neville's method to the data by constructing a recursive table of the form shown in Table 3.4.

Table 3.5

x	$f(x)$
1.0	0.7651977
1.3	0.6200860
1.6	0.4554022
1.9	0.2818186
2.2	0.1103623

Solution Let $x_0 = 1.0$, $x_1 = 1.3$, $x_2 = 1.6$, $x_3 = 1.9$, and $x_4 = 2.2$, then $Q_{0,0} = f(1.0)$, $Q_{1,0} = f(1.3)$, $Q_{2,0} = f(1.6)$, $Q_{3,0} = f(1.9)$, and $Q_{4,0} = f(2.2)$. These are the five polynomials of degree zero (constants) that approximate $f(1.5)$, and are the same as data given in Table 3.5.

Calculating the first-degree approximation $Q_{1,1}(1.5)$ gives

$$\begin{aligned} Q_{1,1}(1.5) &= \frac{(x - x_0)Q_{1,0} - (x - x_1)Q_{0,0}}{x_1 - x_0} \\ &= \frac{(1.5 - 1.0)Q_{1,0} - (1.5 - 1.3)Q_{0,0}}{1.3 - 1.0} \\ &= \frac{0.5(0.6200860) - 0.2(0.7651977)}{0.3} = 0.5233449. \end{aligned}$$

Similarly,

$$Q_{2,1}(1.5) = \frac{(1.5 - 1.3)(0.4554022) - (1.5 - 1.6)(0.6200860)}{1.6 - 1.3} = 0.5102968,$$

$$Q_{3,1}(1.5) = 0.5132634, \quad \text{and} \quad Q_{4,1}(1.5) = 0.5104270.$$

The best linear approximation is expected to be $Q_{2,1}$ because 1.5 is between $x_1 = 1.3$ and $x_2 = 1.6$.

In a similar manner, approximations using higher-degree polynomials are given by

$$Q_{2,2}(1.5) = \frac{(1.5 - 1.0)(0.5102968) - (1.5 - 1.6)(0.5233449)}{1.6 - 1.0} = 0.5124715,$$

$$Q_{3,2}(1.5) = 0.5112857, \quad \text{and} \quad Q_{4,2}(1.5) = 0.5137361.$$

The higher-degree approximations are generated in a similar manner and are shown in Table 3.6. ■

Table 3.6

1.0	0.7651977				
1.3	0.6200860	0.5233449			
1.6	0.4554022	0.5102968	0.5124715		
1.9	0.2818186	0.5132634	0.5112857	0.5118127	
2.2	0.1103623	0.5104270	0.5137361	0.5118302	0.5118200

If the latest approximation, $Q_{4,4}$, was not sufficiently accurate, another node, x_5, could be selected, and another row added to the table:

$$x_5 \quad Q_{5,0} \quad Q_{5,1} \quad Q_{5,2} \quad Q_{5,3} \quad Q_{5,4} \quad Q_{5,5}.$$

Then $Q_{4,4}$, $Q_{5,4}$, and $Q_{5,5}$ could be compared to determine further accuracy.

The function in Example 2 is the Bessel function of the first kind of order zero, whose value at 2.5 is -0.0483838, and the next row of approximations to $f(1.5)$ is

$$2.5 \quad -0.0483838 \quad 0.4807699 \quad 0.5301984 \quad 0.5119070 \quad 0.5118430 \quad 0.5118277.$$

The final new entry, 0.5118277, is correct to all seven decimal places.

The *NumericalAnalysis* package in Maple can be used to apply Neville's method for the values of x and $f(x) = y$ in Table 3.6. After loading the package we define the data with

$xy := [[1.0, 0.7651977], [1.3, 0.6200860], [1.6, 0.4554022], [1.9, 0.2818186]]$

Neville's method using this data gives the approximation at $x = 1.5$ with the command

$p3 := PolynomialInterpolation(xy, method = neville, extrapolate = [1.5])$

The output from Maple for this command is

$$POLYINTERP([[1.0, 0.7651977], [1.3, 0.6200860], [1.6, 0.4554022], [1.9, 0.2818186]],$$
$$method = neville, extrapolate = [1.5], INFO)$$

which isn't very informative. To display the information, we enter the command

$NevilleTable(p3, 1.5)$

and Maple returns an array with four rows and four columns. The nonzero entries corresponding to the top four rows of Table 3.6 (with the first column deleted), the zero entries are simply used to fill up the array.

To add the additional row to the table using the additional data (2.2, 0.1103623) we use the command *p3a* := *AddPoint*(*p3*, [2.2, 0.1103623])

and a new array with all the approximation entries in Table 3.6 is obtained with

NevilleTable(*p3a*, 1.5)

Example 3 Table 3.7 lists the values of $f(x) = \ln x$ accurate to the places given. Use Neville's method and four-digit rounding arithmetic to approximate $f(2.1) = \ln 2.1$ by completing the Neville table.

Table 3.7

i	x_i	$\ln x_i$
0	2.0	0.6931
1	2.2	0.7885
2	2.3	0.8329

Solution Because $x - x_0 = 0.1$, $x - x_1 = -0.1$, $x - x_2 = -0.2$, and we are given $Q_{0,0} = 0.6931$, $Q_{1,0} = 0.7885$, and $Q_{2,0} = 0.8329$, we have

$$Q_{1,1} = \frac{1}{0.2}\left[(0.1)0.7885 - (-0.1)0.6931\right] = \frac{0.1482}{0.2} = 0.7410$$

and

$$Q_{2,1} = \frac{1}{0.1}\left[(-0.1)0.8329 - (-0.2)0.7885\right] = \frac{0.07441}{0.1} = 0.7441.$$

The final approximation we can obtain from this data is

$$Q_{2,1} = \frac{1}{0.3}\left[(0.1)0.7441 - (-0.2)0.7410\right] = \frac{0.2276}{0.3} = 0.7420.$$

These values are shown in Table 3.8. ■

Table 3.8

i	x_i	$x - x_i$	Q_{i0}	Q_{i1}	Q_{i2}
0	2.0	0.1	0.6931		
1	2.2	−0.1	0.7885	0.7410	
2	2.3	−0.2	0.8329	0.7441	0.7420

In the preceding example we have $f(2.1) = \ln 2.1 = 0.7419$ to four decimal places, so the absolute error is

$$|f(2.1) - P_2(2.1)| = |0.7419 - 0.7420| = 10^{-4}.$$

However, $f'(x) = 1/x$, $f''(x) = -1/x^2$, and $f'''(x) = 2/x^3$, so the Lagrange error formula (3.3) in Theorem 3.3 gives the error bound

$$\begin{aligned}|f(2.1) - P_2(2.1)| &= \left|\frac{f'''(\xi(2.1))}{3!}(x - x_0)(x - x_1)(x - x_2)\right| \\ &= \left|\frac{1}{3\left(\xi(2.1)\right)^3}(0.1)(-0.1)(-0.2)\right| \le \frac{0.002}{3(2)^3} = 8.\overline{3} \times 10^{-5}.\end{aligned}$$

Notice that the actual error, 10^{-4}, exceeds the error bound, $8.\overline{3} \times 10^{-5}$. This apparent contradiction is a consequence of finite-digit computations. We used four-digit rounding arithmetic, and the Lagrange error formula (3.3) assumes infinite-digit arithmetic. This caused our actual errors to exceed the theoretical error estimate.

- Remember: You cannot expect more accuracy than the arithmetic provides.

Algorithm 3.1 constructs the entries in Neville's method by rows.

ALGORITHM 3.1

Neville's Iterated Interpolation

To evaluate the interpolating polynomial P on the $n+1$ distinct numbers $x_0, \ldots, x_n$ at the number x for the function f:

INPUT numbers $x, x_0, x_1, \ldots, x_n$; values $f(x_0), f(x_1), \ldots, f(x_n)$ as the first column $Q_{0,0}, Q_{1,0}, \ldots, Q_{n,0}$ of Q.

OUTPUT the table Q with $P(x) = Q_{n,n}$.

Step 1 For $i = 1, 2, \ldots, n$
for $j = 1, 2, \ldots, i$

$$\text{set } Q_{i,j} = \frac{(x - x_{i-j})Q_{i,j-1} - (x - x_i)Q_{i-1,j-1}}{x_i - x_{i-j}}.$$

Step 2 OUTPUT (Q);
STOP.

The algorithm can be modified to allow for the addition of new interpolating nodes. For example, the inequality

$$|Q_{i,i} - Q_{i-1,i-1}| < \varepsilon$$

can be used as a stopping criterion, where ε is a prescribed error tolerance. If the inequality is true, $Q_{i,i}$ is a reasonable approximation to $f(x)$. If the inequality is false, a new interpolation point, x_{i+1}, is added.

EXERCISE SET 3.2

1. Use Neville's method to obtain the approximations for Lagrange interpolating polynomials of degrees one, two, and three to approximate each of the following:

a. $f(8.4)$ if $f(8.1) = 16.94410$, $f(8.3) = 17.56492$, $f(8.6) = 18.50515$, $f(8.7) = 18.82091$

b. $f\left(-\frac{1}{3}\right)$ if $f(-0.75) = -0.07181250$, $f(-0.5) = -0.02475000$, $f(-0.25) = 0.33493750$, $f(0) = 1.10100000$

c. $f(0.25)$ if $f(0.1) = 0.62049958$, $f(0.2) = -0.28398668$, $f(0.3) = 0.00660095$, $f(0.4) = 0.24842440$

d. $f(0.9)$ if $f(0.6) = -0.17694460$, $f(0.7) = 0.01375227$, $f(0.8) = 0.22363362$, $f(1.0) = 0.65809197$

2. Use Neville's method to obtain the approximations for Lagrange interpolating polynomials of degrees one, two, and three to approximate each of the following:

a. $f(0.43)$ if $f(0) = 1$, $f(0.25) = 1.64872$, $f(0.5) = 2.71828$, $f(0.75) = 4.48169$

b. $f(0)$ if $f(-0.5) = 1.93750$, $f(-0.25) = 1.33203$, $f(0.25) = 0.800781$, $f(0.5) = 0.687500$

c. $f(0.18)$ if $f(0.1) = -0.29004986$, $f(0.2) = -0.56079734$, $f(0.3) = -0.81401972$, $f(0.4) = -1.0526302$

d. $f(0.25)$ if $f(-1) = 0.86199480$, $f(-0.5) = 0.95802009$, $f(0) = 1.0986123$, $f(0.5) = 1.2943767$

3. Use Neville's method to approximate $\sqrt{3}$ with the following functions and values.

a. $f(x) = 3^x$ and the values $x_0 = -2$, $x_1 = -1$, $x_2 = 0$, $x_3 = 1$, and $x_4 = 2$.

b. $f(x) = \sqrt{x}$ and the values $x_0 = 0$, $x_1 = 1$, $x_2 = 2$, $x_3 = 4$, and $x_4 = 5$.

c. Compare the accuracy of the approximation in parts (a) and (b).

4. Let $P_3(x)$ be the interpolating polynomial for the data $(0, 0)$, $(0.5, y)$, $(1, 3)$, and $(2, 2)$. Use Neville's method to find y if $P_3(1.5) = 0$.

5. Neville's method is used to approximate $f(0.4)$, giving the following table.

$x_0 = 0$	$P_0 = 1$			
$x_1 = 0.25$	$P_1 = 2$	$P_{01} = 2.6$		
$x_2 = 0.5$	P_2	$P_{1,2}$	$P_{0,1,2}$	
$x_3 = 0.75$	$P_3 = 8$	$P_{2,3} = 2.4$	$P_{1,2,3} = 2.96$	$P_{0,1,2,3} = 3.016$

Determine $P_2 = f(0.5)$.

6. Neville's method is used to approximate $f(0.5)$, giving the following table.

$x_0 = 0$	$P_0 = 0$		
$x_1 = 0.4$	$P_1 = 2.8$	$P_{0,1} = 3.5$	
$x_2 = 0.7$	P_2	$P_{1,2}$	$P_{0,1,2} = \frac{27}{7}$

Determine $P_2 = f(0.7)$.

7. Suppose $x_j = j$, for $j = 0, 1, 2, 3$ and it is known that

$$P_{0,1}(x) = 2x + 1, \quad P_{0,2}(x) = x + 1, \quad \text{and} \quad P_{1,2,3}(2.5) = 3.$$

Find $P_{0,1,2,3}(2.5)$.

8. Suppose $x_j = j$, for $j = 0, 1, 2, 3$ and it is known that

$$P_{0,1}(x) = x + 1, \quad P_{1,2}(x) = 3x - 1, \quad \text{and} \quad P_{1,2,3}(1.5) = 4.$$

Find $P_{0,1,2,3}(1.5)$.

9. Neville's Algorithm is used to approximate $f(0)$ using $f(-2)$, $f(-1)$, $f(1)$, and $f(2)$. Suppose $f(-1)$ was understated by 2 and $f(1)$ was overstated by 3. Determine the error in the original calculation of the value of the interpolating polynomial to approximate $f(0)$.

10. Neville's Algorithm is used to approximate $f(0)$ using $f(-2)$, $f(-1)$, $f(1)$, and $f(2)$. Suppose $f(-1)$ was overstated by 2 and $f(1)$ was understated by 3. Determine the error in the original calculation of the value of the interpolating polynomial to approximate $f(0)$.

11. Construct a sequence of interpolating values y_n to $f(1+\sqrt{10})$, where $f(x) = (1+x^2)^{-1}$ for $-5 \le x \le 5$, as follows: For each $n = 1, 2, \ldots, 10$, let $h = 10/n$ and $y_n = P_n(1+\sqrt{10})$, where $P_n(x)$ is the interpolating polynomial for $f(x)$ at the nodes $x_0^{(n)}, x_1^{(n)}, \ldots, x_n^{(n)}$ and $x_j^{(n)} = -5 + jh$, for each $j = 0, 1, 2, \ldots, n$. Does the sequence $\{y_n\}$ appear to converge to $f(1+\sqrt{10})$?

Inverse Interpolation Suppose $f \in C^1[a, b]$, $f'(x) \neq 0$ on $[a, b]$ and f has one zero p in $[a, b]$. Let $x_0, \ldots, x_n$, be $n + 1$ distinct numbers in $[a, b]$ with $f(x_k) = y_k$, for each $k = 0, 1, \ldots, n$. To approximate p construct the interpolating polynomial of degree n on the nodes $y_0, \ldots, y_n$ for f^{-1}. Since $y_k = f(x_k)$ and $0 = f(p)$, it follows that $f^{-1}(y_k) = x_k$ and $p = f^{-1}(0)$. Using iterated interpolation to approximate $f^{-1}(0)$ is called *iterated inverse interpolation.*

12. Use iterated inverse interpolation to find an approximation to the solution of $x - e^{-x} = 0$, using the data

x	0.3	0.4	0.5	0.6
e^{-x}	0.740818	0.670320	0.606531	0.548812

13. Construct an algorithm that can be used for inverse interpolation.

3.3 Divided Differences

Iterated interpolation was used in the previous section to generate successively higher-degree polynomial approximations at a specific point. Divided-difference methods introduced in this section are used to successively generate the polynomials themselves.

Suppose that $P_n(x)$ is the nth Lagrange polynomial that agrees with the function f at the distinct numbers $x_0, x_1, \ldots, x_n$. Although this polynomial is unique, there are alternate algebraic representations that are useful in certain situations. The divided differences of f with respect to $x_0, x_1, \ldots, x_n$ are used to express $P_n(x)$ in the form

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)\cdots(x - x_{n-1}), \quad (3.5)$$

for appropriate constants $a_0, a_1, \ldots, a_n$. To determine the first of these constants, a_0, note that if $P_n(x)$ is written in the form of Eq. (3.5), then evaluating $P_n(x)$ at x_0 leaves only the constant term a_0; that is,

$$a_0 = P_n(x_0) = f(x_0).$$

As in so many areas, Isaac Newton is prominent in the study of difference equations. He developed interpolation formulas as early as 1675, using his Δ notation in tables of differences. He took a very general approach to the difference formulas, so explicit examples that he produced, including Lagrange's formulas, are often known by other names.

Similarly, when $P(x)$ is evaluated at x_1, the only nonzero terms in the evaluation of $P_n(x_1)$ are the constant and linear terms,

$$f(x_0) + a_1(x_1 - x_0) = P_n(x_1) = f(x_1);$$

so

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \quad (3.6)$$

We now introduce the divided-difference notation, which is related to Aitken's Δ^2 notation used in Section 2.5. The *zeroth divided difference* of the function f with respect to x_i, denoted $f[x_i]$, is simply the value of f at x_i:

$$f[x_i] = f(x_i). \quad (3.7)$$

The remaining divided differences are defined recursively; the *first divided difference* of f with respect to x_i and x_{i+1} is denoted $f[x_i, x_{i+1}]$ and defined as

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}. \quad (3.8)$$

The *second divided difference*, $f[x_i, x_{i+1}, x_{i+2}]$, is defined as

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}.$$

Similarly, after the $(k-1)$st divided differences,

$$f[x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+k-1}] \quad \text{and} \quad f[x_{i+1}, x_{i+2}, \ldots, x_{i+k-1}, x_{i+k}],$$

have been determined, the **kth divided difference** relative to $x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+k}$ is

$$f[x_i, x_{i+1}, \ldots, x_{i+k-1}, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}. \quad (3.9)$$

The process ends with the single *nth divided difference*,

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, x_2, \ldots, x_n] - f[x_0, x_1, \ldots, x_{n-1}]}{x_n - x_0}.$$

Because of Eq. (3.6) we can write $a_1 = f[x_0, x_1]$, just as a_0 can be expressed as $a_0 = f(x_0) = f[x_0]$. Hence the interpolating polynomial in Eq. (3.5) is

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}).$$

As might be expected from the evaluation of a_0 and a_1, the required constants are

$$a_k = f[x_0, x_1, x_2, \ldots, x_k],$$

for each $k = 0, 1, \ldots, n$. So $P_n(x)$ can be rewritten in a form called Newton's Divided-Difference:

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0) \cdots (x - x_{k-1}). \qquad (3.10)$$

The value of $f[x_0, x_1, \ldots, x_k]$ is independent of the order of the numbers $x_0, x_1, \ldots, x_k$, as shown in Exercise 21.

The generation of the divided differences is outlined in Table 3.9. Two fourth and one fifth difference can also be determined from these data.

Table 3.9

x	$f(x)$	First divided differences	Second divided differences	Third divided differences
x_0	$f[x_0]$			
		$f[x_0, x_1] = \dfrac{f[x_1] - f[x_0]}{x_1 - x_0}$		
x_1	$f[x_1]$		$f[x_0, x_1, x_2] = \dfrac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$	
		$f[x_1, x_2] = \dfrac{f[x_2] - f[x_1]}{x_2 - x_1}$		$f[x_0, x_1, x_2, x_3] = \dfrac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}$
x_2	$f[x_2]$		$f[x_1, x_2, x_3] = \dfrac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}$	
		$f[x_2, x_3] = \dfrac{f[x_3] - f[x_2]}{x_3 - x_2}$		$f[x_1, x_2, x_3, x_4] = \dfrac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1}$
x_3	$f[x_3]$		$f[x_2, x_3, x_4] = \dfrac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2}$	
		$f[x_3, x_4] = \dfrac{f[x_4] - f[x_3]}{x_4 - x_3}$		$f[x_2, x_3, x_4, x_5] = \dfrac{f[x_3, x_4, x_5] - f[x_2, x_3, x_4]}{x_5 - x_2}$
x_4	$f[x_4]$		$f[x_3, x_4, x_5] = \dfrac{f[x_4, x_5] - f[x_3, x_4]}{x_5 - x_3}$	
		$f[x_4, x_5] = \dfrac{f[x_5] - f[x_4]}{x_5 - x_4}$		
x_5	$f[x_5]$			

ALGORITHM 3.2

Newton's Divided-Difference Formula

To obtain the divided-difference coefficients of the interpolatory polynomial P on the $(n+1)$ distinct numbers $x_0, x_1, \ldots, x_n$ for the function f:

INPUT numbers $x_0, x_1, \ldots, x_n$; values $f(x_0), f(x_1), \ldots, f(x_n)$ as $F_{0,0}, F_{1,0}, \ldots, F_{n,0}$.

OUTPUT the numbers $F_{0,0}, F_{1,1}, \ldots, F_{n,n}$ where

$$P_n(x) = F_{0,0} + \sum_{i=1}^{n} F_{i,i} \prod_{j=0}^{i-1}(x - x_j). \qquad (F_{i,i} \text{ is } f[x_0, x_1, \ldots, x_i].)$$

Step 1 For $i = 1, 2, \ldots, n$
  For $j = 1, 2, \ldots, i$
    set $F_{i,j} = \dfrac{F_{i,j-1} - F_{i-1,j-1}}{x_i - x_{i-j}}$. $\quad (F_{i,j} = f[x_{i-j}, \ldots, x_i].)$

Step 2 OUTPUT $(F_{0,0}, F_{1,1}, \ldots, F_{n,n})$;
  STOP. ■

The form of the output in Algorithm 3.2 can be modified to produce all the divided differences, as shown in Example 1.

Example 1 Complete the divided difference table for the data used in Example 1 of Section 3.2, and reproduced in Table 3.10, and construct the interpolating polynomial that uses all this data.

Table 3.10

x	$f(x)$
1.0	0.7651977
1.3	0.6200860
1.6	0.4554022
1.9	0.2818186
2.2	0.1103623

Solution The first divided difference involving x_0 and x_1 is

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{0.6200860 - 0.7651977}{1.3 - 1.0} = -0.4837057.$$

The remaining first divided differences are found in a similar manner and are shown in the fourth column in Table 3.11.

Table 3.11

i	x_i	$f[x_i]$	$f[x_{i-1}, x_i]$	$f[x_{i-2}, x_{i-1}, x_i]$	$f[x_{i-3}, \ldots, x_i]$	$f[x_{i-4}, \ldots, x_i]$
0	1.0	0.7651977				
			−0.4837057			
1	1.3	0.6200860		−0.1087339		
			−0.5489460		0.0658784	
2	1.6	0.4554022		−0.0494433		0.0018251
			−0.5786120		0.0680685	
3	1.9	0.2818186		0.0118183		
			−0.5715210			
4	2.2	0.1103623				

The second divided difference involving x_0, x_1, and x_2 is

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{-0.5489460 - (-0.4837057)}{1.6 - 1.0} = -0.1087339.$$

The remaining second divided differences are shown in the 5th column of Table 3.11. The third divided difference involving x_0, x_1, x_2, and x_3 and the fourth divided difference involving all the data points are, respectively,

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} = \frac{-0.0494433 - (-0.1087339)}{1.9 - 1.0}$$
$$= 0.0658784,$$

and

$$f[x_0, x_1, x_2, x_3, x_4] = \frac{f[x_1, x_2, x_3, x_4] - f[x_0, x_1, x_2, x_3]}{x_4 - x_0} = \frac{0.0680685 - 0.0658784}{2.2 - 1.0}$$
$$= 0.0018251.$$

All the entries are given in Table 3.11.

The coefficients of the Newton forward divided-difference form of the interpolating polynomial are along the diagonal in the table. This polynomial is

$$P_4(x) = 0.7651977 - 0.4837057(x - 1.0) - 0.1087339(x - 1.0)(x - 1.3)$$
$$+ 0.0658784(x - 1.0)(x - 1.3)(x - 1.6)$$
$$+ 0.0018251(x - 1.0)(x - 1.3)(x - 1.6)(x - 1.9).$$

Notice that the value $P_4(1.5) = 0.5118200$ agrees with the result in Table 3.6 for Example 2 of Section 3.2, as it must because the polynomials are the same. ■

We can use Maple with the *NumericalAnalysis* package to create the Newton Divided-Difference table. First load the package and define the x and $f(x) = y$ values that will be used to generate the first four rows of Table 3.11.

$xy := [[1.0, 0.7651977], [1.3, 0.6200860], [1.6, 0.4554022], [1.9, 0.2818186]]$

The command to create the divided-difference table is

p3 := PolynomialInterpolation(xy, independentvar = 'x', method = newton)

A matrix containing the divided-difference table as its nonzero entries is created with the

DividedDifferenceTable(p3)

We can add another row to the table with the command

p4 := AddPoint(p3, [2.2, 0.1103623])

which produces the divided-difference table with entries corresponding to those in Table 3.11.

The Newton form of the interpolation polynomial is created with

Interpolant(p4)

which produces the polynomial in the form of $P_4(x)$ in Example 1, except that in place of the first two terms of $P_4(x)$:

$$0.7651977 - 0.4837057(x - 1.0)$$

Maple gives this as $1.248903367 - 0.4837056667x$.

The Mean Value Theorem 1.8 applied to Eq. (3.8) when $i = 0$,

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0},$$

implies that when f' exists, $f[x_0, x_1] = f'(\xi)$ for some number ξ between x_0 and x_1. The following theorem generalizes this result.

Theorem 3.6 Suppose that $f \in C^n[a, b]$ and $x_0, x_1, \ldots, x_n$ are distinct numbers in $[a, b]$. Then a number ξ exists in (a, b) with

$$f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$ ■

Proof Let

$$g(x) = f(x) - P_n(x).$$

Since $f(x_i) = P_n(x_i)$ for each $i = 0, 1, \ldots, n$, the function g has $n+1$ distinct zeros in $[a, b]$. Generalized Rolle's Theorem 1.10 implies that a number ξ in (a, b) exists with $g^{(n)}(\xi) = 0$, so

$$0 = f^{(n)}(\xi) - P_n^{(n)}(\xi).$$

Since $P_n(x)$ is a polynomial of degree n whose leading coefficient is $f[x_0, x_1, \ldots, x_n]$,

$$P_n^{(n)}(x) = n!\, f[x_0, x_1, \ldots, x_n],$$

for all values of x. As a consequence,

$$f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!}.$$ ■ ■ ■

Newton's divided-difference formula can be expressed in a simplified form when the nodes are arranged consecutively with equal spacing. In this case, we introduce the notation $h = x_{i+1} - x_i$, for each $i = 0, 1, \ldots, n-1$ and let $x = x_0 + sh$. Then the difference $x - x_i$ is $x - x_i = (s - i)h$. So Eq. (3.10) becomes

$$\begin{aligned} P_n(x) = P_n(x_0 + sh) &= f[x_0] + shf[x_0, x_1] + s(s-1)h^2 f[x_0, x_1, x_2] \\ &\quad + \cdots + s(s-1)\cdots(s-n+1)h^n f[x_0, x_1, \ldots, x_n] \\ &= f[x_0] + \sum_{k=1}^{n} s(s-1)\cdots(s-k+1)h^k f[x_0, x_1, \ldots, x_k]. \end{aligned}$$

Using binomial-coefficient notation,

$$\binom{s}{k} = \frac{s(s-1)\cdots(s-k+1)}{k!},$$

we can express $P_n(x)$ compactly as

$$P_n(x) = P_n(x_0 + sh) = f[x_0] + \sum_{k=1}^{n} \binom{s}{k} k! h^k f[x_0, x_i, \ldots, x_k]. \tag{3.11}$$

Forward Differences

The **Newton forward-difference formula**, is constructed by making use of the forward difference notation Δ introduced in Aitken's Δ^2 method. With this notation,

$$\begin{aligned} f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1}{h}(f(x_1) - f(x_0)) = \frac{1}{h}\Delta f(x_0) \\ f[x_0, x_1, x_2] &= \frac{1}{2h}\left[\frac{\Delta f(x_1) - \Delta f(x_0)}{h}\right] = \frac{1}{2h^2}\Delta^2 f(x_0), \end{aligned}$$

and, in general,

$$f[x_0, x_1, \ldots, x_k] = \frac{1}{k!h^k}\Delta^k f(x_0).$$

Since $f[x_0] = f(x_0)$, Eq. (3.11) has the following form.

Newton Forward-Difference Formula

$$P_n(x) = f(x_0) + \sum_{k=1}^{n} \binom{s}{k} \Delta^k f(x_0) \tag{3.12}$$

Backward Differences

If the interpolating nodes are reordered from last to first as $x_n, x_{n-1}, \ldots, x_0$, we can write the interpolatory formula as

$$\begin{aligned} P_n(x) &= f[x_n] + f[x_n, x_{n-1}](x - x_n) + f[x_n, x_{n-1}, x_{n-2}](x - x_n)(x - x_{n-1}) \\ &\quad + \cdots + f[x_n, \ldots, x_0](x - x_n)(x - x_{n-1})\cdots(x - x_1). \end{aligned}$$

If, in addition, the nodes are equally spaced with $x = x_n + sh$ and $x = x_i + (s+n-i)h$, then

$$\begin{aligned} P_n(x) &= P_n(x_n + sh) \\ &= f[x_n] + shf[x_n, x_{n-1}] + s(s+1)h^2 f[x_n, x_{n-1}, x_{n-2}] + \cdots \\ &\quad + s(s+1)\cdots(s+n-1)h^n f[x_n, \ldots, x_0]. \end{aligned}$$

This is used to derive a commonly applied formula known as the **Newton backward-difference formula**. To discuss this formula, we need the following definition.

Definition 3.7 Given the sequence $\{p_n\}_{n=0}^{\infty}$, define the backward difference ∇p_n (read *nabla* p_n) by

$$\nabla p_n = p_n - p_{n-1}, \quad \text{for } n \geq 1.$$

Higher powers are defined recursively by

$$\nabla^k p_n = \nabla(\nabla^{k-1} p_n), \quad \text{for } k \geq 2. \qquad \blacksquare$$

Definition 3.7 implies that

$$f[x_n, x_{n-1}] = \frac{1}{h}\nabla f(x_n), \quad f[x_n, x_{n-1}, x_{n-2}] = \frac{1}{2h^2}\nabla^2 f(x_n),$$

and, in general,

$$f[x_n, x_{n-1}, \ldots, x_{n-k}] = \frac{1}{k!h^k}\nabla^k f(x_n).$$

Consequently,

$$P_n(x) = f[x_n] + s\nabla f(x_n) + \frac{s(s+1)}{2}\nabla^2 f(x_n) + \cdots + \frac{s(s+1)\cdots(s+n-1)}{n!}\nabla^n f(x_n).$$

If we extend the binomial coefficient notation to include all real values of s by letting

$$\binom{-s}{k} = \frac{-s(-s-1)\cdots(-s-k+1)}{k!} = (-1)^k \frac{s(s+1)\cdots(s+k-1)}{k!},$$

then

$$P_n(x) = f[x_n] + (-1)^1\binom{-s}{1}\nabla f(x_n) + (-1)^2\binom{-s}{2}\nabla^2 f(x_n) + \cdots + (-1)^n\binom{-s}{n}\nabla^n f(x_n).$$

This gives the following result.

Newton Backward–Difference Formula

$$P_n(x) = f[x_n] + \sum_{k=1}^{n} (-1)^k \binom{-s}{k} \nabla^k f(x_n) \qquad (3.13)$$

Illustration The divided-difference Table 3.12 corresponds to the data in Example 1.

Table 3.12

		First divided differences	Second divided differences	Third divided differences	Fourth divided differences
1.0	0.7651977				
		−0.4837057			
1.3	0.6200860		−0.1087339		
		−0.5489460		0.0658784	
1.6	0.4554022		−0.0494433		0.0018251
		−0.5786120		0.0680685	
1.9	0.2818186		0.0118183		
		−0.5715210			
2.2	0.1103623				

Only one interpolating polynomial of degree at most 4 uses these five data points, but we will organize the data points to obtain the best interpolation approximations of degrees 1, 2, and 3. This will give us a sense of accuracy of the fourth-degree approximation for the given value of x.

If an approximation to $f(1.1)$ is required, the reasonable choice for the nodes would be $x_0 = 1.0$, $x_1 = 1.3$, $x_2 = 1.6$, $x_3 = 1.9$, and $x_4 = 2.2$ since this choice makes the earliest possible use of the data points closest to $x = 1.1$, and also makes use of the fourth divided difference. This implies that $h = 0.3$ and $s = \frac{1}{3}$, so the Newton forward divided-difference formula is used with the divided differences that have a *solid* underline (___) in Table 3.12:

$$\begin{aligned}
P_4(1.1) = P_4(1.0 + \tfrac{1}{3}(0.3))
&= 0.7651977 + \frac{1}{3}(0.3)(-0.4837057) + \frac{1}{3}\left(-\frac{2}{3}\right)(0.3)^2(-0.1087339)\\
&\quad + \frac{1}{3}\left(-\frac{2}{3}\right)\left(-\frac{5}{3}\right)(0.3)^3(0.0658784)\\
&\quad + \frac{1}{3}\left(-\frac{2}{3}\right)\left(-\frac{5}{3}\right)\left(-\frac{8}{3}\right)(0.3)^4(0.0018251)\\
&= 0.7196460.
\end{aligned}$$

To approximate a value when x is close to the end of the tabulated values, say, $x = 2.0$, we would again like to make the earliest use of the data points closest to x. This requires using the Newton backward divided-difference formula with $s = -\frac{2}{3}$ and the divided differences in Table 3.12 that have a *wavy* underline ($\sim\sim\sim$). Notice that the fourth divided difference is used in both formulas.

$$\begin{aligned}
P_4(2.0) = P_4\left(2.2 - \frac{2}{3}(0.3)\right)
&= 0.1103623 - \frac{2}{3}(0.3)(-0.5715210) - \frac{2}{3}\left(\frac{1}{3}\right)(0.3)^2(0.0118183)\\
&\quad - \frac{2}{3}\left(\frac{1}{3}\right)\left(\frac{4}{3}\right)(0.3)^3(0.0680685) - \frac{2}{3}\left(\frac{1}{3}\right)\left(\frac{4}{3}\right)\left(\frac{7}{3}\right)(0.3)^4(0.0018251)\\
&= 0.2238754.
\end{aligned}$$

□

Centered Differences

The Newton forward- and backward-difference formulas are not appropriate for approximating $f(x)$ when x lies near the center of the table because neither will permit the highest-order difference to have x_0 close to x. A number of divided-difference formulas are available for this case, each of which has situations when it can be used to maximum advantage. These methods are known as **centered-difference formulas**. We will consider only one centered-difference formula, Stirling's method.

For the centered-difference formulas, we choose x_0 near the point being approximated and label the nodes directly below x_0 as $x_1, x_2, \ldots$ and those directly above as $x_{-1}, x_{-2}, \ldots$. With this convention, **Stirling's formula** is given by

$$\begin{aligned} P_n(x) = P_{2m+1}(x) &= f[x_0] + \frac{sh}{2}(f[x_{-1}, x_0] + f[x_0, x_1]) + s^2h^2 f[x_{-1}, x_0, x_1] \\ &+ \frac{s(s^2-1)h^3}{2} f[x_{-2}, x_{-1}, x_0, x_1] + f[x_{-1}, x_0, x_1, x_2]) \\ &+ \cdots + s^2(s^2-1)(s^2-4)\cdots(s^2-(m-1)^2)h^{2m} f[x_{-m}, \ldots, x_m] \\ &+ \frac{s(s^2-1)\cdots(s^2-m^2)h^{2m+1}}{2}(f[x_{-m-1}, \ldots, x_m] + f[x_{-m}, \ldots, x_{m+1}]), \end{aligned} \tag{3.14}$$

James Stirling (1692–1770) published this and numerous other formulas in *Methodus Differentialis* in 1720. Techniques for accelerating the convergence of various series are included in this work.

if $n = 2m + 1$ is odd. If $n = 2m$ is even, we use the same formula but delete the last line. The entries used for this formula are underlined in Table 3.13.

Table 3.13

x	$f(x)$	First divided differences	Second divided differences	Third divided differences	Fourth divided differences
x_{-2}	$f[x_{-2}]$				
		$f[x_{-2}, x_{-1}]$			
x_{-1}	$f[x_{-1}]$		$f[x_{-2}, x_{-1}, x_0]$		
		$\underline{f[x_{-1}, x_0]}$		$\underline{f[x_{-2}, x_{-1}, x_0, x_1]}$	
x_0	$\underline{f[x_0]}$		$\underline{f[x_{-1}, x_0, x_1]}$		$\underline{f[x_{-2}, x_{-1}, x_0, x_1, x_2]}$
		$\underline{f[x_0, x_1]}$		$\underline{f[x_{-1}, x_0, x_1, x_2]}$	
x_1	$f[x_1]$		$f[x_0, x_1, x_2]$		
		$f[x_1, x_2]$			
x_2	$f[x_2]$				

Example 2 Consider the table of data given in the previous examples. Use Stirling's formula to approximate $f(1.5)$ with $x_0 = 1.6$.

Solution To apply Stirling's formula we use the *underlined* entries in the difference Table 3.14.

Table 3.14

x	$f(x)$	First divided differences	Second divided differences	Third divided differences	Fourth divided differences
1.0	0.7651977				
		−0.4837057			
1.3	0.6200860		−0.1087339		
		<u>−0.5489460</u>		<u>0.0658784</u>	
1.6	<u>0.4554022</u>		<u>−0.0494433</u>		<u>0.0018251</u>
		<u>−0.5786120</u>		<u>0.0680685</u>	
1.9	0.2818186		0.0118183		
		−0.5715210			
2.2	0.1103623				

The formula, with $h = 0.3$, $x_0 = 1.6$, and $s = -\frac{1}{3}$, becomes

$$
\begin{aligned}
f(1.5) &\approx P_4\left(1.6 + \left(-\frac{1}{3}\right)(0.3)\right) \\
&= 0.4554022 + \left(-\frac{1}{3}\right)\left(\frac{0.3}{2}\right)((-0.5489460) + (-0.5786120)) \\
&\quad + \left(-\frac{1}{3}\right)^2 (0.3)^2(-0.0494433) \\
&\quad + \frac{1}{2}\left(-\frac{1}{3}\right)\left(\left(-\frac{1}{3}\right)^2 - 1\right)(0.3)^3(0.0658784 + 0.0680685) \\
&\quad + \left(-\frac{1}{3}\right)^2\left(\left(-\frac{1}{3}\right)^2 - 1\right)(0.3)^4(0.0018251) = 0.5118200.
\end{aligned}
$$

■

Most texts on numerical analysis written before the wide-spread use of computers have extensive treatments of divided-difference methods. If a more comprehensive treatment of this subject is needed, the book by Hildebrand [Hild] is a particularly good reference.

EXERCISE SET 3.3

1. Use Eq. (3.10) or Algorithm 3.2 to construct interpolating polynomials of degree one, two, and three for the following data. Approximate the specified value using each of the polynomials.

a. $f(8.4)$ if $f(8.1) = 16.94410$, $f(8.3) = 17.56492$, $f(8.6) = 18.50515$, $f(8.7) = 18.82091$

b. $f(0.9)$ if $f(0.6) = -0.17694460$, $f(0.7) = 0.01375227$, $f(0.8) = 0.22363362$, $f(1.0) = 0.65809197$

2. Use Eq. (3.10) or Algorithm 3.2 to construct interpolating polynomials of degree one, two, and three for the following data. Approximate the specified value using each of the polynomials.

a. $f(0.43)$ if $f(0) = 1$, $f(0.25) = 1.64872$, $f(0.5) = 2.71828$, $f(0.75) = 4.48169$

b. $f(0)$ if $f(-0.5) = 1.93750$, $f(-0.25) = 1.33203$, $f(0.25) = 0.800781$, $f(0.5) = 0.687500$

3. Use Newton the forward-difference formula to construct interpolating polynomials of degree one, two, and three for the following data. Approximate the specified value using each of the polynomials.

a. $f\left(-\frac{1}{3}\right)$ if $f(-0.75) = -0.07181250$, $f(-0.5) = -0.02475000$, $f(-0.25) = 0.33493750$, $f(0) = 1.10100000$

b. $f(0.25)$ if $f(0.1) = -0.62049958$, $f(0.2) = -0.28398668$, $f(0.3) = 0.00660095$, $f(0.4) = 0.24842440$

4. Use the Newton forward-difference formula to construct interpolating polynomials of degree one, two, and three for the following data. Approximate the specified value using each of the polynomials.

a. $f(0.43)$ if $f(0) = 1$, $f(0.25) = 1.64872$, $f(0.5) = 2.71828$, $f(0.75) = 4.48169$

b. $f(0.18)$ if $f(0.1) = -0.29004986$, $f(0.2) = -0.56079734$, $f(0.3) = -0.81401972$, $f(0.4) = -1.0526302$

5. Use the Newton backward-difference formula to construct interpolating polynomials of degree one, two, and three for the following data. Approximate the specified value using each of the polynomials.

a. $f(-1/3)$ if $f(-0.75) = -0.07181250$, $f(-0.5) = -0.02475000$, $f(-0.25) = 0.33493750$, $f(0) = 1.10100000$

b. $f(0.25)$ if $f(0.1) = -0.62049958$, $f(0.2) = -0.28398668$, $f(0.3) = 0.00660095$, $f(0.4) = 0.24842440$

6. Use the Newton backward-difference formula to construct interpolating polynomials of degree one, two, and three for the following data. Approximate the specified value using each of the polynomials.
   a. $f(0.43)$ if $f(0) = 1$, $f(0.25) = 1.64872$, $f(0.5) = 2.71828$, $f(0.75) = 4.48169$
   b. $f(0.25)$ if $f(-1) = 0.86199480$, $f(-0.5) = 0.95802009$, $f(0) = 1.0986123$, $f(0.5) = 1.2943767$
7. a. Use Algorithm 3.2 to construct the interpolating polynomial of degree three for the unequally spaced points given in the following table:

x	$f(x)$
−0.1	5.30000
0.0	2.00000
0.2	3.19000
0.3	1.00000

   b. Add $f(0.35) = 0.97260$ to the table, and construct the interpolating polynomial of degree four.
8. a. Use Algorithm 3.2 to construct the interpolating polynomial of degree four for the unequally spaced points given in the following table:

x	$f(x)$
0.0	−6.00000
0.1	−5.89483
0.3	−5.65014
0.6	−5.17788
1.0	−4.28172

   b. Add $f(1.1) = -3.99583$ to the table, and construct the interpolating polynomial of degree five.
9. a. Approximate $f(0.05)$ using the following data and the Newton forward-difference formula:

x	0.0	0.2	0.4	0.6	0.8
$f(x)$	1.00000	1.22140	1.49182	1.82212	2.22554

   b. Use the Newton backward-difference formula to approximate $f(0.65)$.
   c. Use Stirling's formula to approximate $f(0.43)$.
10. Show that the polynomial interpolating the following data has degree 3.

x	−2	−1	0	1	2	3
$f(x)$	1	4	11	16	13	−4

11. a. Show that the cubic polynomials

$$P(x) = 3 - 2(x+1) + 0(x+1)(x) + (x+1)(x)(x-1)$$

and

$$Q(x) = -1 + 4(x+2) - 3(x+2)(x+1) + (x+2)(x+1)(x)$$

both interpolate the data

x	−2	−1	0	1	2
$f(x)$	−1	3	1	−1	3

   b. Why does part (a) not violate the uniqueness property of interpolating polynomials?
12. A fourth-degree polynomial $P(x)$ satisfies $\Delta^4 P(0) = 24$, $\Delta^3 P(0) = 6$, and $\Delta^2 P(0) = 0$, where $\Delta P(x) = P(x+1) - P(x)$. Compute $\Delta^2 P(10)$.

13. The following data are given for a polynomial $P(x)$ of unknown degree.

x	0	1	2
$P(x)$	2	−1	4

Determine the coefficient of x^2 in $P(x)$ if all third-order forward differences are 1.

14. The following data are given for a polynomial $P(x)$ of unknown degree.

x	0	1	2	3
$P(x)$	4	9	15	18

Determine the coefficient of x^3 in $P(x)$ if all fourth-order forward differences are 1.

15. The Newton forward-difference formula is used to approximate $f(0.3)$ given the following data.

x	0.0	0.2	0.4	0.6
$f(x)$	15.0	21.0	30.0	51.0

Suppose it is discovered that $f(0.4)$ was understated by 10 and $f(0.6)$ was overstated by 5. By what amount should the approximation to $f(0.3)$ be changed?

16. For a function f, the Newton divided-difference formula gives the interpolating polynomial

$$P_3(x) = 1 + 4x + 4x(x - 0.25) + \frac{16}{3}x(x - 0.25)(x - 0.5),$$

on the nodes $x_0 = 0$, $x_1 = 0.25$, $x_2 = 0.5$ and $x_3 = 0.75$. Find $f(0.75)$.

17. For a function f, the forward-divided differences are given by

$x_0 = 0.0$	$f[x_0]$		
		$f[x_0, x_1]$	
$x_1 = 0.4$	$f[x_1]$		$f[x_0, x_1, x_2] = \frac{50}{7}$
		$f[x_1, x_2] = 10$	
$x_2 = 0.7$	$f[x_2] = 6$		

Determine the missing entries in the table.

18. **a.** The introduction to this chapter included a table listing the population of the United States from 1950 to 2000. Use appropriate divided differences to approximate the population in the years 1940, 1975, and 2020.

b. The population in 1940 was approximately 132,165,000. How accurate do you think your 1975 and 2020 figures are?

19. Given

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + a_2(x - x_0)(x - x_1)$$
$$+ a_3(x - x_0)(x - x_1)(x - x_2) + \cdots$$
$$+ a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}),$$

use $P_n(x_2)$ to show that $a_2 = f[x_0, x_1, x_2]$.

20. Show that

$$f[x_0, x_1, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi(x))}{(n+1)!},$$

for some $\xi(x)$. [*Hint:* From Eq. (3.3),

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

Considering the interpolation polynomial of degree $n+1$ on $x_0, x_1, \ldots, x_n, x$, we have

$$f(x) = P_{n+1}(x) = P_n(x) + f[x_0, x_1, \ldots, x_n, x](x - x_0)\cdots(x - x_n).]$$

21. Let $i_0, i_1, \ldots, i_n$ be a rearrangement of the integers $0, 1, \ldots, n$. Show that $f[x_{i_0}, x_{i_1}, \ldots, x_{i_n}] = f[x_0, x_1, \ldots, x_n]$. [*Hint:* Consider the leading coefficient of the nth Lagrange polynomial on the data $\{x_0, x_1, \ldots, x_n\} = \{x_{i_0}, x_{i_1}, \ldots, x_{i_n}\}$.]

3.4 Hermite Interpolation

The Latin word *osculum*, literally a "small mouth" or "kiss", when applied to a curve indicates that it just touches and has the same shape. Hermite interpolation has this osculating property. It matches a given curve, and its derivative forces the interpolating curve to "kiss" the given curve.

Osculating polynomials generalize both the Taylor polynomials and the Lagrange polynomials. Suppose that we are given $n+1$ distinct numbers $x_0, x_1, \ldots, x_n$ in $[a, b]$ and nonnegative integers $m_0, m_1, \ldots, m_n$, and $m = \max\{m_0, m_1, \ldots, m_n\}$. The osculating polynomial approximating a function $f \in C^m[a, b]$ at x_i, for each $i = 0, \ldots, n$, is the polynomial of least degree that has the same values as the function f and all its derivatives of order less than or equal to m_i at each x_i. The degree of this osculating polynomial is at most

$$M = \sum_{i=0}^{n} m_i + n$$

because the number of conditions to be satisfied is $\sum_{i=0}^{n} m_i + (n+1)$, and a polynomial of degree M has $M+1$ coefficients that can be used to satisfy these conditions.

Definition 3.8 Let $x_0, x_1, \ldots, x_n$ be $n+1$ distinct numbers in $[a, b]$ and for $i = 0, 1, \ldots, n$ let m_i be a nonnegative integer. Suppose that $f \in C^m[a, b]$, where $m = \max_{0 \le i \le n} m_i$.

The **osculating polynomial** approximating f is the polynomial $P(x)$ of least degree such that

$$\frac{d^k P(x_i)}{dx^k} = \frac{d^k f(x_i)}{dx^k}, \quad \text{for each } i = 0, 1, \ldots, n \quad \text{and} \quad k = 0, 1, \ldots, m_i. \qquad \blacksquare$$

Charles Hermite (1822–1901) made significant mathematical discoveries throughout his life in areas such as complex analysis and number theory, particularly involving the theory of equations. He is perhaps best known for proving in 1873 that e is transcendental, that is, it is not the solution to any algebraic equation having integer coefficients. This lead in 1882 to Lindemann's proof that π is also transcendental, which demonstrated that it is impossible to use the standard geometry tools of Euclid to construct a square that has the same area as a unit circle.

Note that when $n = 0$, the osculating polynomial approximating f is the m_0th Taylor polynomial for f at x_0. When $m_i = 0$ for each i, the osculating polynomial is the nth Lagrange polynomial interpolating f on $x_0, x_1, \ldots, x_n$.

Hermite Polynomials

The case when $m_i = 1$, for each $i = 0, 1, \ldots, n$, gives the **Hermite polynomials**. For a given function f, these polynomials agree with f at $x_0, x_1, \ldots, x_n$. In addition, since their first derivatives agree with those of f, they have the same "shape" as the function at $(x_i, f(x_i))$ in the sense that the *tangent lines* to the polynomial and the function agree. We will restrict our study of osculating polynomials to this situation and consider first a theorem that describes precisely the form of the Hermite polynomials.

Theorem 3.9 If $f \in C^1[a, b]$ and $x_0, \ldots, x_n \in [a, b]$ are distinct, the unique polynomial of least degree agreeing with f and f' at $x_0, \ldots, x_n$ is the Hermite polynomial of degree at most $2n+1$ given by

$$H_{2n+1}(x) = \sum_{j=0}^{n} f(x_j)H_{n,j}(x) + \sum_{j=0}^{n} f'(x_j)\hat{H}_{n,j}(x),$$

Hermite gave a description of a general osculatory polynomial in a letter to Carl W. Borchardt in 1878, to whom he regularly sent his new results. His demonstration is an interesting application of the use of complex integration techniques to solve a real-valued problem.

where, for $L_{n,j}(x)$ denoting the jth Lagrange coefficient polynomial of degree n, we have

$$H_{n,j}(x) = [1 - 2(x - x_j)L'_{n,j}(x_j)]L^2_{n,j}(x) \quad \text{and} \quad \hat{H}_{n,j}(x) = (x - x_j)L^2_{n,j}(x).$$

Moreover, if $f \in C^{2n+2}[a, b]$, then

$$f(x) = H_{2n+1}(x) + \frac{(x - x_0)^2 \ldots (x - x_n)^2}{(2n + 2)!} f^{(2n+2)}(\xi(x)),$$

for some (generally unknown) $\xi(x)$ in the interval (a, b). ■

Proof First recall that

$$L_{n,j}(x_i) = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j. \end{cases}$$

Hence when $i \neq j$,

$$H_{n,j}(x_i) = 0 \qquad \text{and} \qquad \hat{H}_{n,j}(x_i) = 0,$$

whereas, for each i,

$$H_{n,i}(x_i) = [1 - 2(x_i - x_i)L'_{n,i}(x_i)] \cdot 1 = 1 \qquad \text{and} \qquad \hat{H}_{n,i}(x_i) = (x_i - x_i) \cdot 1^2 = 0.$$

As a consequence

$$H_{2n+1}(x_i) = \sum_{\substack{j=0 \\ j\neq i}}^{n} f(x_j) \cdot 0 + f(x_i) \cdot 1 + \sum_{j=0}^{n} f'(x_j) \cdot 0 = f(x_i),$$

so H_{2n+1} agrees with f at $x_0, x_1, \ldots, x_n$.

To show the agreement of H'_{2n+1} with f' at the nodes, first note that $L_{n,j}(x)$ is a factor of $H'_{n,j}(x)$, so $H'_{n,j}(x_i) = 0$ when $i \neq j$. In addition, when $i = j$ we have $L_{n,i}(x_i) = 1$, so

$$\begin{aligned} H'_{n,i}(x_i) &= -2L'_{n,i}(x_i) \cdot L^2_{n,i}(x_i) + [1 - 2(x_i - x_i)L'_{n,i}(x_i)]2L_{n,i}(x_i)L'_{n,i}(x_i) \\ &= -2L'_{n,i}(x_i) + 2L'_{n,i}(x_i) = 0. \end{aligned}$$

Hence, $H'_{n,j}(x_i) = 0$ for all i and j.

Finally,

$$\begin{aligned} \hat{H}'_{n,j}(x_i) &= L^2_{n,j}(x_i) + (x_i - x_j)2L_{n,j}(x_i)L'_{n,j}(x_i) \\ &= L_{n,j}(x_i)[L_{n,j}(x_i) + 2(x_i - x_j)L'_{n,j}(x_i)], \end{aligned}$$

so $\hat{H}'_{n,j}(x_i) = 0$ if $i \neq j$ and $\hat{H}'_{n,i}(x_i) = 1$. Combining these facts, we have

$$H'_{2n+1}(x_i) = \sum_{j=0}^{n} f(x_j) \cdot 0 + \sum_{\substack{j=0 \\ j\neq i}}^{n} f'(x_j) \cdot 0 + f'(x_i) \cdot 1 = f'(x_i).$$

Therefore, H_{2n+1} agrees with f and H'_{2n+1} with f' at $x_0, x_1, \ldots, x_n$.

The uniqueness of this polynomial and the error formula are considered in Exercise 11. ■ ■ ■

Example 1 Use the Hermite polynomial that agrees with the data listed in Table 3.15 to find an approximation of $f(1.5)$.

Table 3.15

k	x_k	$f(x_k)$	$f'(x_k)$
0	1.3	0.6200860	−0.5220232
1	1.6	0.4554022	−0.5698959
2	1.9	0.2818186	−0.5811571

Solution We first compute the Lagrange polynomials and their derivatives. This gives

$$L_{2,0}(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{50}{9}x^2 - \frac{175}{9}x + \frac{152}{9}, \qquad L'_{2,0}(x) = \frac{100}{9}x - \frac{175}{9};$$

$$L_{2,1}(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{-100}{9}x^2 + \frac{320}{9}x - \frac{247}{9}, \qquad L'_{2,1}(x) = \frac{-200}{9}x + \frac{320}{9};$$

and

$$L_{2,2} = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{50}{9}x^2 - \frac{145}{9}x + \frac{104}{9}, \qquad L'_{2,2}(x) = \frac{100}{9}x - \frac{145}{9}.$$

The polynomials $H_{2,j}(x)$ and $\hat{H}_{2,j}(x)$ are then

$$H_{2,0}(x) = [1 - 2(x-1.3)(-5)]\left(\frac{50}{9}x^2 - \frac{175}{9}x + \frac{152}{9}\right)^2$$

$$= (10x - 12)\left(\frac{50}{9}x^2 - \frac{175}{9}x + \frac{152}{9}\right)^2,$$

$$H_{2,1}(x) = 1 \cdot \left(\frac{-100}{9}x^2 + \frac{320}{9}x - \frac{247}{9}\right)^2,$$

$$H_{2,2}(x) = 10(2-x)\left(\frac{50}{9}x^2 - \frac{145}{9}x + \frac{104}{9}\right)^2,$$

$$\hat{H}_{2,0}(x) = (x-1.3)\left(\frac{50}{9}x^2 - \frac{175}{9}x + \frac{152}{9}\right)^2,$$

$$\hat{H}_{2,1}(x) = (x-1.6)\left(\frac{-100}{9}x^2 + \frac{320}{9}x - \frac{247}{9}\right)^2,$$

and

$$\hat{H}_{2,2}(x) = (x-1.9)\left(\frac{50}{9}x^2 - \frac{145}{9}x + \frac{104}{9}\right)^2.$$

Finally

$$H_5(x) = 0.6200860H_{2,0}(x) + 0.4554022H_{2,1}(x) + 0.2818186H_{2,2}(x)$$
$$- 0.5220232\hat{H}_{2,0}(x) - 0.5698959\hat{H}_{2,1}(x) - 0.5811571\hat{H}_{2,2}(x)$$

and

$$\begin{aligned} H_5(1.5) &= 0.6200860\left(\frac{4}{27}\right) + 0.4554022\left(\frac{64}{81}\right) + 0.2818186\left(\frac{5}{81}\right) \\ &\quad - 0.5220232\left(\frac{4}{405}\right) - 0.5698959\left(\frac{-32}{405}\right) - 0.5811571\left(\frac{-2}{405}\right) \\ &= 0.5118277, \end{aligned}$$

a result that is accurate to the places listed. ■

Although Theorem 3.9 provides a complete description of the Hermite polynomials, it is clear from Example 1 that the need to determine and evaluate the Lagrange polynomials and their derivatives makes the procedure tedious even for small values of n.

Hermite Polynomials Using Divided Differences

There is an alternative method for generating Hermite approximations that has as its basis the Newton interpolatory divided-difference formula (3.10) at $x_0, x_1, \ldots, x_n$, that is,

$$P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)\cdots(x - x_{k-1}).$$

The alternative method uses the connection between the nth divided difference and the nth derivative of f, as outlined in Theorem 3.6 in Section 3.3.

Suppose that the distinct numbers $x_0, x_1, \ldots, x_n$ are given together with the values of f and f' at these numbers. Define a new sequence $z_0, z_1, \ldots, z_{2n+1}$ by

$$z_{2i} = z_{2i+1} = x_i, \quad \text{for each } i = 0, 1, \ldots, n,$$

and construct the divided difference table in the form of Table 3.9 that uses $z_0, z_1, \ldots, z_{2n+1}$.

Since $z_{2i} = z_{2i+1} = x_i$ for each i, we cannot define $f[z_{2i}, z_{2i+1}]$ by the divided difference formula. However, if we assume, based on Theorem 3.6, that the reasonable substitution in this situation is $f[z_{2i}, z_{2i+1}] = f'(z_{2i}) = f'(x_i)$, we can use the entries

$$f'(x_0), f'(x_1), \ldots, f'(x_n)$$

in place of the undefined first divided differences

$$f[z_0, z_1], f[z_2, z_3], \ldots, f[z_{2n}, z_{2n+1}].$$

The remaining divided differences are produced as usual, and the appropriate divided differences are employed in Newton's interpolatory divided-difference formula. Table 3.16 shows the entries that are used for the first three divided-difference columns when determining the Hermite polynomial $H_5(x)$ for x_0, x_1, and x_2. The remaining entries are generated in the same manner as in Table 3.9. The Hermite polynomial is then given by

$$H_{2n+1}(x) = f[z_0] + \sum_{k=1}^{2n+1} f[z_0, \ldots, z_k](x - z_0)(x - z_1)\cdots(x - z_{k-1}).$$

A proof of this fact can be found in [Pow], p. 56.

Table 3.16

z	$f(z)$	First divided differences	Second divided differences
$z_0 = x_0$	$f[z_0] = f(x_0)$		
		$f[z_0, z_1] = f'(x_0)$	
$z_1 = x_0$	$f[z_1] = f(x_0)$		$f[z_0, z_1, z_2] = \dfrac{f[z_1, z_2] - f[z_0, z_1]}{z_2 - z_0}$
		$f[z_1, z_2] = \dfrac{f[z_2] - f[z_1]}{z_2 - z_1}$	
$z_2 = x_1$	$f[z_2] = f(x_1)$		$f[z_1, z_2, z_3] = \dfrac{f[z_2, z_3] - f[z_1, z_2]}{z_3 - z_1}$
		$f[z_2, z_3] = f'(x_1)$	
$z_3 = x_1$	$f[z_3] = f(x_1)$		$f[z_2, z_3, z_4] = \dfrac{f[z_3, z_4] - f[z_2, z_3]}{z_4 - z_2}$
		$f[z_3, z_4] = \dfrac{f[z_4] - f[z_3]}{z_4 - z_3}$	
$z_4 = x_2$	$f[z_4] = f(x_2)$		$f[z_3, z_4, z_5] = \dfrac{f[z_4, z_5] - f[z_3, z_4]}{z_5 - z_3}$
		$f[z_4, z_5] = f'(x_2)$	
$z_5 = x_2$	$f[z_5] = f(x_2)$		

Example 2 Use the data given in Example 1 and the divided difference method to determine the Hermite polynomial approximation at $x = 1.5$.

Solution The underlined entries in the first three columns of Table 3.17 are the data given in Example 1. The remaining entries in this table are generated by the standard divided-difference formula (3.9).

For example, for the second entry in the third column we use the second 1.3 entry in the second column and the first 1.6 entry in that column to obtain

$$\frac{0.4554022 - 0.6200860}{1.6 - 1.3} = -0.5489460.$$

For the first entry in the fourth column we use the first 1.3 entry in the third column and the first 1.6 entry in that column to obtain

$$\frac{-0.5489460 - (-0.5220232)}{1.6 - 1.3} = -0.0897427.$$

The value of the Hermite polynomial at 1.5 is

$$\begin{aligned} H_5(1.5) = {} & f[1.3] + f'(1.3)(1.5 - 1.3) + f[1.3, 1.3, 1.6](1.5 - 1.3)^2 \\ & + f[1.3, 1.3, 1.6, 1.6](1.5 - 1.3)^2(1.5 - 1.6) \\ & + f[1.3, 1.3, 1.6, 1.6, 1.9](1.5 - 1.3)^2(1.5 - 1.6)^2 \\ & + f[1.3, 1.3, 1.6, 1.6, 1.9, 1.9](1.5 - 1.3)^2(1.5 - 1.6)^2(1.5 - 1.9) \\ = {} & 0.6200860 + (-0.5220232)(0.2) + (-0.0897427)(0.2)^2 \\ & + 0.0663657(0.2)^2(-0.1) + 0.0026663(0.2)^2(-0.1)^2 \\ & + (-0.0027738)(0.2)^2(-0.1)^2(-0.4) \\ = {} & 0.5118277. \end{aligned}$$

■

Table 3.17

1.3	0.6200860					
		−0.5220232				
1.3	0.6200860		−0.0897427			
		−0.5489460		0.0663657		
1.6	0.4554022		−0.0698330		0.0026663	
		−0.5698959		0.0679655		−0.0027738
1.6	0.4554022		−0.0290537		0.0010020	
		−0.5786120		0.0685667		
1.9	0.2818186		−0.0084837			
		−0.5811571				
1.9	0.2818186					

The technique used in Algorithm 3.3 can be extended for use in determining other osculating polynomials. A concise discussion of the procedures can be found in [Pow], pp. 53–57.

ALGORITHM 3.3

Hermite Interpolation

To obtain the coefficients of the Hermite interpolating polynomial $H(x)$ on the $(n+1)$ distinct numbers $x_0, \dots, x_n$ for the function f:

INPUT numbers $x_0, x_1, \dots, x_n$; values $f(x_0), \dots, f(x_n)$ and $f'(x_0), \dots, f'(x_n)$.

OUTPUT the numbers $Q_{0,0}, Q_{1,1}, \dots, Q_{2n+1,2n+1}$ where

$$\begin{aligned} H(x) = {} & Q_{0,0} + Q_{1,1}(x-x_0) + Q_{2,2}(x-x_0)^2 + Q_{3,3}(x-x_0)^2(x-x_1) \\ & + Q_{4,4}(x-x_0)^2(x-x_1)^2 + \cdots \\ & + Q_{2n+1,2n+1}(x-x_0)^2(x-x_1)^2 \cdots (x-x_{n-1})^2(x-x_n). \end{aligned}$$

Step 1 For $i = 0, 1, \dots, n$ do Steps 2 and 3.

Step 2 Set $z_{2i} = x_i$;
$z_{2i+1} = x_i$;
$Q_{2i,0} = f(x_i)$;
$Q_{2i+1,0} = f(x_i)$;
$Q_{2i+1,1} = f'(x_i)$.

Step 3 If $i \neq 0$ then set

$$Q_{2i,1} = \frac{Q_{2i,0} - Q_{2i-1,0}}{z_{2i} - z_{2i-1}}.$$

Step 4 For $i = 2, 3, \dots, 2n+1$

for $j = 2, 3, \dots, i$ set $Q_{i,j} = \dfrac{Q_{i,j-1} - Q_{i-1,j-1}}{z_i - z_{i-j}}$.

Step 5 OUTPUT $(Q_{0,0}, Q_{1,1}, \dots, Q_{2n+1,2n+1})$;
STOP ■

The *NumericalAnalysis* package in Maple can be used to construct the Hermite coefficients. We first need to load the package and to define the data that is being used, in this case, x_i, $f(x_i)$, and $f'(x_i)$ for $i = 0, 1, \dots, n$. This is done by presenting the data in the form $[x_i, f(x_i), f'(x_i)]$. For example, the data for Example 2 is entered as

xy := [[1.3, 0.6200860, −0.5220232], [1.6, 0.4554022, −0.5698959],
[1.9, 0.2818186, −0.5811571]]

Then the command

$h5 := PolynomialInterpolation(xy, method = hermite, independentvar = 'x')$

produces an array whose nonzero entries correspond to the values in Table 3.17. The Hermite interpolating polynomial is created with the command

$Interpolant(h5))$

This gives the polynomial in (almost) Newton forward-difference form

$$1.29871616 - 0.5220232x - 0.08974266667(x - 1.3)^2 + 0.06636555557(x - 1.3)^2(x - 1.6) + 0.002666666633(x - 1.3)^2(x - 1.6)^2 - 0.002774691277(x - 1.3)^2(x - 1.6)^2(x - 1.9)$$

If a standard representation of the polynomial is needed, it is found with

$expand(Interpolant(h5))$

giving the Maple response

$$1.001944063 - 0.0082292208x - 0.2352161732x^2 - 0.01455607812x^3 + 0.02403178946x^4 - 0.002774691277x^5$$

EXERCISE SET 3.4

1. Use Theorem 3.9 or Algorithm 3.3 to construct an approximating polynomial for the following data.

a.

x	$f(x)$	$f'(x)$
8.3	17.56492	3.116256
8.6	18.50515	3.151762

b.

x	$f(x)$	$f'(x)$
0.8	0.22363362	2.1691753
1.0	0.65809197	2.0466965

c.

x	$f(x)$	$f'(x)$
−0.5	−0.0247500	0.7510000
−0.25	0.3349375	2.1890000
0	1.1010000	4.0020000

d.

x	$f(x)$	$f'(x)$
0.1	−0.62049958	3.58502082
0.2	−0.28398668	3.14033271
0.3	0.00660095	2.66668043
0.4	0.24842440	2.16529366

2. Use Theorem 3.9 or Algorithm 3.3 to construct an approximating polynomial for the following data.

a.

x	$f(x)$	$f'(x)$
0	1.00000	2.00000
0.5	2.71828	5.43656

b.

x	$f(x)$	$f'(x)$
−0.25	1.33203	0.437500
0.25	0.800781	−0.625000

c.

x	$f(x)$	$f'(x)$
0.1	−0.29004996	−2.8019975
0.2	−0.56079734	−2.6159201
0.3	−0.81401972	−2.9734038

d.

x	$f(x)$	$f'(x)$
−1	0.86199480	0.15536240
−0.5	0.95802009	0.23269654
0	1.0986123	0.33333333
0.5	1.2943767	0.45186776

3. The data in Exercise 1 were generated using the following functions. Use the polynomials constructed in Exercise 1 for the given value of x to approximate $f(x)$, and calculate the absolute error.

a. $f(x) = x \ln x$; approximate $f(8.4)$.

b. $f(x) = \sin(e^x - 2)$; approximate $f(0.9)$.

c. $f(x) = x^3 + 4.001x^2 + 4.002x + 1.101$; approximate $f(-1/3)$.

d. $f(x) = x \cos x - 2x^2 + 3x - 1$; approximate $f(0.25)$.

4. The data in Exercise 2 were generated using the following functions. Use the polynomials constructed in Exercise 2 for the given value of x to approximate $f(x)$, and calculate the absolute error.

a. $f(x) = e^{2x}$; approximate $f(0.43)$.

b. $f(x) = x^4 - x^3 + x^2 - x + 1$; approximate $f(0)$.

c. $f(x) = x^2 \cos x - 3x$; approximate $f(0.18)$.

d. $f(x) = \ln(e^x + 2)$; approximate $f(0.25)$.

5. **a.** Use the following values and five-digit rounding arithmetic to construct the Hermite interpolating polynomial to approximate $\sin 0.34$.

x	$\sin x$	$D_x \sin x = \cos x$
0.30	0.29552	0.95534
0.32	0.31457	0.94924
0.35	0.34290	0.93937

b. Determine an error bound for the approximation in part (a), and compare it to the actual error.

c. Add $\sin 0.33 = 0.32404$ and $\cos 0.33 = 0.94604$ to the data, and redo the calculations.

6. Let $f(x) = 3xe^x - e^{2x}$.

a. Approximate $f(1.03)$ by the Hermite interpolating polynomial of degree at most three using $x_0 = 1$ and $x_1 = 1.05$. Compare the actual error to the error bound.

b. Repeat (a) with the Hermite interpolating polynomial of degree at most five, using $x_0 = 1$, $x_1 = 1.05$, and $x_2 = 1.07$.

7. Use the error formula and Maple to find a bound for the errors in the approximations of $f(x)$ in parts (a) and (c) of Exercise 3.

8. Use the error formula and Maple to find a bound for the errors in the approximations of $f(x)$ in parts (a) and (c) of Exercise 4.

9. The following table lists data for the function described by $f(x) = e^{0.1x^2}$. Approximate $f(1.25)$ by using $H_5(1.25)$ and $H_3(1.25)$, where H_5 uses the nodes $x_0 = 1$, $x_1 = 2$, and $x_2 = 3$; and H_3 uses the nodes $\bar{x}_0 = 1$ and $\bar{x}_1 = 1.5$. Find error bounds for these approximations.

x	$f(x) = e^{0.1x^2}$	$f'(x) = 0.2xe^{0.1x^2}$
$x_0 = \bar{x}_0 = 1$	1.105170918	0.2210341836
$\bar{x}_1 = 1.5$	1.252322716	0.3756968148
$x_1 = 2$	1.491824698	0.5967298792
$x_2 = 3$	2.459603111	1.475761867

10. A car traveling along a straight road is clocked at a number of points. The data from the observations are given in the following table, where the time is in seconds, the distance is in feet, and the speed is in feet per second.

Time	0	3	5	8	13
Distance	0	225	383	623	993
Speed	75	77	80	74	72

a. Use a Hermite polynomial to predict the position of the car and its speed when $t = 10$ s.

b. Use the derivative of the Hermite polynomial to determine whether the car ever exceeds a 55 mi/h speed limit on the road. If so, what is the first time the car exceeds this speed?

c. What is the predicted maximum speed for the car?

11. **a.** Show that $H_{2n+1}(x)$ is the unique polynomial of least degree agreeing with f and f' at $x_0, \ldots, x_n$. [*Hint:* Assume that $P(x)$ is another such polynomial and consider $D = H_{2n+1} - P$ and D' at $x_0, x_1, \ldots, x_n$.]

b. Derive the error term in Theorem 3.9. [*Hint:* Use the same method as in the Lagrange error derivation, Theorem 3.3, defining

$$g(t) = f(t) - H_{2n+1}(t) - \frac{(t-x_0)^2 \cdots (t-x_n)^2}{(x-x_0)^2 \cdots (x-x_n)^2}[f(x) - H_{2n+1}(x)]$$

and using the fact that $g'(t)$ has $(2n+2)$ *distinct* zeros in $[a, b]$.]

12. Let $z_0 = x_0$, $z_1 = x_0$, $z_2 = x_1$, and $z_3 = x_1$. Form the following divided-difference table.

$z_0 = x_0$	$f[z_0] = f(x_0)$			
		$f[z_0, z_1] = f'(x_0)$		
$z_1 = x_0$	$f[z_1] = f(x_0)$		$f[z_0, z_1, z_2]$	
		$f[z_1, z_2]$		$f[z_0, z_1, z_2, z_3]$
$z_2 = x_1$	$f[z_2] = f(x_1)$		$f[z_1, z_2, z_3]$	
		$f[z_2, z_3] = f'(x_1)$		
$z_3 = x_1$	$f[z_3] = f(x_1)$			

Show that the cubic Hermite polynomial $H_3(x)$ can also be written as $f[z_0] + f[z_0, z_1](x - x_0) + f[z_0, z_1, z_2](x - x_0)^2 + f[z_0, z_1, z_2, z_3](x - x_0)^2(x - x_1)$.

3.5 Cubic Spline Interpolation[1]

The previous sections concerned the approximation of arbitrary functions on closed intervals using a single polynomial. However, high-degree polynomials can oscillate erratically, that is, a minor fluctuation over a small portion of the interval can induce large fluctuations over the entire range. We will see a good example of this in Figure 3.14 at the end of this section.

An alternative approach is to divide the approximation interval into a collection of subintervals and construct a (generally) different approximating polynomial on each subinterval. This is called **piecewise-polynomial approximation**.

Piecewise-Polynomial Approximation

The simplest piecewise-polynomial approximation is **piecewise-linear** interpolation, which consists of joining a set of data points

$$\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$$

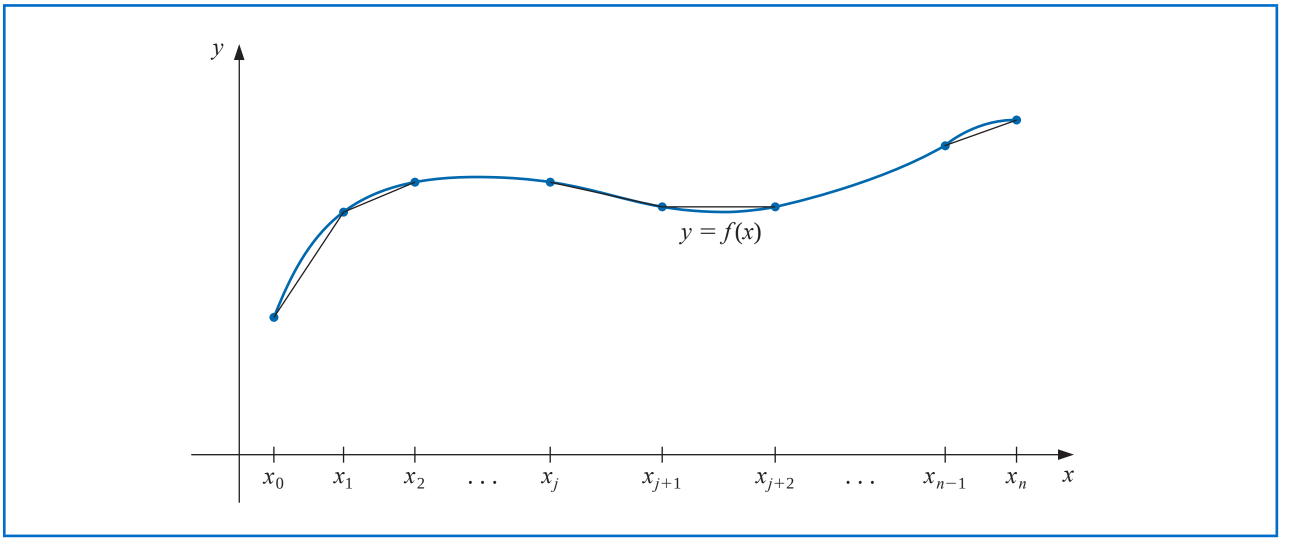
by a series of straight lines, as shown in Figure 3.7.

A disadvantage of linear function approximation is that there is likely no differentiability at the endpoints of the subintervals, which, in a geometrical context, means that the interpolating function is not "smooth." Often it is clear from physical conditions that smoothness is required, so the approximating function must be continuously differentiable.

An alternative procedure is to use a piecewise polynomial of Hermite type. For example, if the values of f and of f' are known at each of the points $x_0 < x_1 < \cdots < x_n$, a cubic Hermite polynomial can be used on each of the subintervals $[x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n]$ to obtain a function that has a continuous derivative on the interval $[x_0, x_n]$.

[1]The proofs of the theorems in this section rely on results in Chapter 6.

Figure 3.7



Isaac Jacob Schoenberg (1903–1990) developed his work on splines during World War II while on leave from the University of Pennsylvania to work at the Army's Ballistic Research Laboratory in Aberdeen, Maryland. His original work involved numerical procedures for solving differential equations. The much broader application of splines to the areas of data fitting and computer-aided geometric design became evident with the widespread availability of computers in the 1960s.

To determine the appropriate Hermite cubic polynomial on a given interval is simply a matter of computing $H_3(x)$ for that interval. The Lagrange interpolating polynomials needed to determine H_3 are of first degree, so this can be accomplished without great difficulty. However, to use Hermite piecewise polynomials for general interpolation, we need to know the derivative of the function being approximated, and this is frequently unavailable.

The remainder of this section considers approximation using piecewise polynomials that require no specific derivative information, except perhaps at the endpoints of the interval on which the function is being approximated.

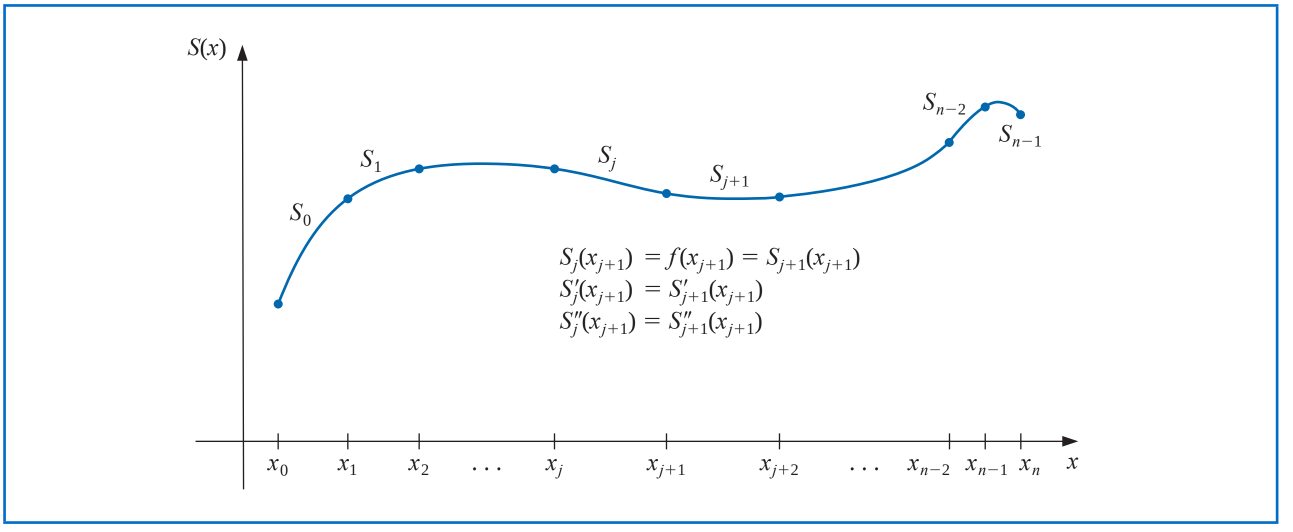
The simplest type of differentiable piecewise-polynomial function on an entire interval $[x_0, x_n]$ is the function obtained by fitting one quadratic polynomial between each successive pair of nodes. This is done by constructing a quadratic on $[x_0, x_1]$ agreeing with the function at x_0 and x_1, another quadratic on $[x_1, x_2]$ agreeing with the function at x_1 and x_2, and so on. A general quadratic polynomial has three arbitrary constants—the constant term, the coefficient of x, and the coefficient of x^2—and only two conditions are required to fit the data at the endpoints of each subinterval. So flexibility exists that permits the quadratics to be chosen so that the interpolant has a continuous derivative on $[x_0, x_n]$. The difficulty arises because we generally need to specify conditions about the derivative of the interpolant at the endpoints x_0 and x_n. There is not a sufficient number of constants to ensure that the conditions will be satisfied. (See Exercise 26.)

The root of the word "spline" is the same as that of splint. It was originally a small strip of wood that could be used to join two boards. Later the word was used to refer to a long flexible strip, generally of metal, that could be used to draw continuous smooth curves by forcing the strip to pass through specified points and tracing along the curve.

Cubic Splines

The most common piecewise-polynomial approximation uses cubic polynomials between each successive pair of nodes and is called **cubic spline interpolation**. A general cubic polynomial involves four constants, so there is sufficient flexibility in the cubic spline procedure to ensure that the interpolant is not only continuously differentiable on the interval, but also has a continuous second derivative. The construction of the cubic spline does not, however, assume that the derivatives of the interpolant agree with those of the function it is approximating, even at the nodes. (See Figure 3.8.)

Figure 3.8



Definition 3.10 Given a function f defined on $[a, b]$ and a set of nodes $a = x_0 < x_1 < \cdots < x_n = b$, a **cubic spline interpolant** S for f is a function that satisfies the following conditions:

(a) $S(x)$ is a cubic polynomial, denoted $S_j(x)$, on the subinterval $[x_j, x_{j+1}]$ for each $j = 0, 1, \ldots, n-1$;

(b) $S_j(x_j) = f(x_j)$ and $S_j(x_{j+1}) = f(x_{j+1})$ for each $j = 0, 1, \ldots, n-1$;

(c) $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$; (*Implied by* **(b)**.)

(d) $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$;

(e) $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for each $j = 0, 1, \ldots, n-2$;

(f) One of the following sets of boundary conditions is satisfied:

(i) $S''(x_0) = S''(x_n) = 0$ (**natural** *(or free)* **boundary**);

(ii) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (**clamped boundary**). ■

A natural spline has no conditions imposed for the direction at its endpoints, so the curve takes the shape of a straight line after it passes through the interpolation points nearest its endpoints. The name derives from the fact that this is the natural shape a flexible strip assumes if forced to pass through specified interpolation points with no additional constraints. (See Figure 3.9.)

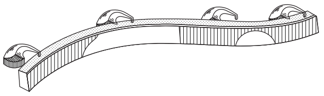


Figure 3.9

Although cubic splines are defined with other boundary conditions, the conditions given in (**f**) are sufficient for our purposes. When the free boundary conditions occur, the spline is called a **natural spline**, and its graph approximates the shape that a long flexible rod would assume if forced to go through the data points $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$.

In general, clamped boundary conditions lead to more accurate approximations because they include more information about the function. However, for this type of boundary condition to hold, it is necessary to have either the values of the derivative at the endpoints or an accurate approximation to those values.

Example 1 Construct a natural cubic spline that passes through the points $(1, 2)$, $(2, 3)$, and $(3, 5)$.

Solution This spline consists of two cubics. The first for the interval $[1, 2]$, denoted

$$S_0(x) = a_0 + b_0(x-1) + c_0(x-1)^2 + d_0(x-1)^3,$$

and the other for [2, 3], denoted

$$S_1(x) = a_1 + b_1(x-2) + c_1(x-2)^2 + d_1(x-2)^3.$$

There are 8 constants to be determined, which requires 8 conditions. Four conditions come from the fact that the splines must agree with the data at the nodes. Hence

$$2 = f(1) = a_0, \quad 3 = f(2) = a_0 + b_0 + c_0 + d_0, \quad 3 = f(2) = a_1, \quad \text{and}$$
$$5 = f(3) = a_1 + b_1 + c_1 + d_1.$$

Two more come from the fact that $S_0'(2) = S_1'(2)$ and $S_0''(2) = S_1''(2)$. These are

$$S_0'(2) = S_1'(2): \quad b_0 + 2c_0 + 3d_0 = b_1 \qquad \text{and} \qquad S_0''(2) = S_1''(2): \quad 2c_0 + 6d_0 = 2c_1$$

The final two come from the natural boundary conditions:

$$S_0''(1) = 0: \quad 2c_0 = 0 \qquad \text{and} \qquad S_1''(3) = 0: \quad 2c_1 + 6d_1 = 0.$$

Solving this system of equations gives the spline

$$S(x) = \begin{cases} 2 + \frac{3}{4}(x-1) + \frac{1}{4}(x-1)^3, & \text{for } x \in [1,2] \\ 3 + \frac{3}{2}(x-2) + \frac{3}{4}(x-2)^2 - \frac{1}{4}(x-2)^3, & \text{for } x \in [2,3] \end{cases}$$

■

Construction of a Cubic Spline

As the preceding example demonstrates, a spline defined on an interval that is divided into n subintervals will require determining $4n$ constants. To construct the cubic spline interpolant for a given function f, the conditions in the definition are applied to the cubic polynomials

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3,$$

Clamping a spline indicates that the ends of the flexible strip are fixed so that it is forced to take a specific direction at each of its endpoints. This is important, for example, when two spline functions should match at their endpoints. This is done mathematically by specifying the values of the derivative of the curve at the endpoints of the spline.

for each $j = 0, 1, \ldots, n-1$. Since $S_j(x_j) = a_j = f(x_j)$, condition **(c)** can be applied to obtain

$$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j(x_{j+1} - x_j) + c_j(x_{j+1} - x_j)^2 + d_j(x_{j+1} - x_j)^3,$$

for each $j = 0, 1, \ldots, n-2$.

The terms $x_{j+1} - x_j$ are used repeatedly in this development, so it is convenient to introduce the simpler notation

$$h_j = x_{j+1} - x_j,$$

for each $j = 0, 1, \ldots, n-1$. If we also define $a_n = f(x_n)$, then the equation

$$a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3 \tag{3.15}$$

holds for each $j = 0, 1, \ldots, n-1$.

In a similar manner, define $b_n = S'(x_n)$ and observe that

$$S_j'(x) = b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2$$

implies $S_j'(x_j) = b_j$, for each $j = 0, 1, \ldots, n-1$. Applying condition **(d)** gives

$$b_{j+1} = b_j + 2c_jh_j + 3d_jh_j^2, \tag{3.16}$$

for each $j = 0, 1, \ldots, n-1$.

Another relationship between the coefficients of S_j is obtained by defining $c_n = S''(x_n)/2$ and applying condition **(e)**. Then, for each $j = 0, 1, \ldots, n-1$,

$$c_{j+1} = c_j + 3d_jh_j. \tag{3.17}$$

Solving for d_j in Eq. (3.17) and substituting this value into Eqs. (3.15) and (3.16) gives, for each $j = 0, 1, \ldots, n-1$, the new equations

$$a_{j+1} = a_j + b_jh_j + \frac{h_j^2}{3}(2c_j + c_{j+1}) \tag{3.18}$$

and

$$b_{j+1} = b_j + h_j(c_j + c_{j+1}). \tag{3.19}$$

The final relationship involving the coefficients is obtained by solving the appropriate equation in the form of equation (3.18), first for b_j,

$$b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1}), \tag{3.20}$$

and then, with a reduction of the index, for b_{j-1}. This gives

$$b_{j-1} = \frac{1}{h_{j-1}}(a_j - a_{j-1}) - \frac{h_{j-1}}{3}(2c_{j-1} + c_j).$$

Substituting these values into the equation derived from Eq. (3.19), with the index reduced by one, gives the linear system of equations

$$h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_jc_{j+1} = \frac{3}{h_j}(a_{j+1} - a_j) - \frac{3}{h_{j-1}}(a_j - a_{j-1}), \tag{3.21}$$

for each $j = 1, 2, \ldots, n-1$. This system involves only the $\{c_j\}_{j=0}^{n}$ as unknowns. The values of $\{h_j\}_{j=0}^{n-1}$ and $\{a_j\}_{j=0}^{n}$ are given, respectively, by the spacing of the nodes $\{x_j\}_{j=0}^{n}$ and the values of f at the nodes. So once the values of $\{c_j\}_{j=0}^{n}$ are determined, it is a simple matter to find the remainder of the constants $\{b_j\}_{j=0}^{n-1}$ from Eq. (3.20) and $\{d_j\}_{j=0}^{n-1}$ from Eq. (3.17). Then we can construct the cubic polynomials $\{S_j(x)\}_{j=0}^{n-1}$.

The major question that arises in connection with this construction is whether the values of $\{c_j\}_{j=0}^{n}$ can be found using the system of equations given in (3.21) and, if so, whether these values are unique. The following theorems indicate that this is the case when either of the boundary conditions given in part **(f)** of the definition are imposed. The proofs of these theorems require material from linear algebra, which is discussed in Chapter 6.

Natural Splines

Theorem 3.11 If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$, then f has a unique natural spline interpolant S on the nodes $x_0, x_1, \ldots, x_n$; that is, a spline interpolant that satisfies the natural boundary conditions $S''(a) = 0$ and $S''(b) = 0$. ■

Proof The boundary conditions in this case imply that $c_n = S''(x_n)/2 = 0$ and that

$$0 = S''(x_0) = 2c_0 + 6d_0(x_0 - x_0),$$

so $c_0 = 0$. The two equations $c_0 = 0$ and $c_n = 0$ together with the equations in (3.21) produce a linear system described by the vector equation $A\mathbf{x} = \mathbf{b}$, where A is the $(n+1) \times (n+1)$ matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \ddots & & \vdots \\ 0 & h_1 & 2(h_1 + h_2) & h_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ \vdots & & h_{n-2} & 2(h_{n-2} + h_{n-1}) & & h_{n-1} \\ 0 & \cdots & 0 & 0 & & 1 \end{bmatrix},$$

and $\mathbf{b}$ and $\mathbf{x}$ are the vectors

$$\mathbf{b} = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}.$$

The matrix A is strictly diagonally dominant, that is, in each row the magnitude of the diagonal entry exceeds the sum of the magnitudes of all the other entries in the row. A linear system with a matrix of this form will be shown by Theorem 6.21 in Section 6.6 to have a unique solution for $c_0, c_1, \ldots, c_n$. ■ ■ ■

The solution to the cubic spline problem with the boundary conditions $S''(x_0) = S''(x_n) = 0$ can be obtained by applying Algorithm 3.4.

ALGORITHM 3.4

Natural Cubic Spline

To construct the cubic spline interpolant S for the function f, defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S''(x_0) = S''(x_n) = 0$:

INPUT $n; x_0, x_1, \ldots, x_n; a_0 = f(x_0), a_1 = f(x_1), \ldots, a_n = f(x_n)$.

OUTPUT a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$.

(*Note:* $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ *for* $x_j \le x \le x_{j+1}$.)

Step 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{i+1} - x_i$.

Step 2 For $i = 1, 2, \ldots, n-1$ set

$$\alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}).$$

Step 3 Set $l_0 = 1$; (*Steps 3, 4, 5, and part of Step 6 solve a tridiagonal linear system using a method described in Algorithm 6.7.*)
$\mu_0 = 0$;
$z_0 = 0$.

Step 4 For $i = 1, 2, \ldots, n-1$
set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;
$\mu_i = h_i / l_i$;
$z_i = (\alpha_i - h_{i-1}z_{i-1})/l_i$.

Step 5 Set $l_n = 1$;
$z_n = 0$;
$c_n = 0$.

Step 6 For $j = n-1, n-2, \ldots, 0$
set $c_j = z_j - \mu_j c_{j+1}$;
$b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$;
$d_j = (c_{j+1} - c_j)/(3h_j)$.

Step 7 OUTPUT $(a_j, b_j, c_j, d_j$ for $j = 0, 1, \ldots, n-1)$;
STOP. ■

Example 2 At the beginning of Chapter 3 we gave some Taylor polynomials to approximate the exponential $f(x) = e^x$. Use the data points $(0, 1)$, $(1, e)$, $(2, e^2)$, and $(3, e^3)$ to form a natural spline $S(x)$ that approximates $f(x) = e^x$.

Solution We have $n = 3$, $h_0 = h_1 = h_2 = 1$, $a_0 = 1$, $a_1 = e$, $a_2 = e^2$, and $a_3 = e^3$. So the matrix A and the vectors $\mathbf{b}$ and $\mathbf{x}$ given in Theorem 3.11 have the forms

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 3(e^2 - 2e + 1) \\ 3(e^3 - 2e^2 + e) \\ 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}.$$

The vector-matrix equation $A\mathbf{x} = \mathbf{b}$ is equivalent to the system of equations

$$c_0 = 0,$$
$$c_0 + 4c_1 + c_2 = 3(e^2 - 2e + 1),$$
$$c_1 + 4c_2 + c_3 = 3(e^3 - 2e^2 + e),$$
$$c_3 = 0.$$

This system has the solution $c_0 = c_3 = 0$, and to 5 decimal places,

$$c_1 = \frac{1}{5}(-e^3 + 6e^2 - 9e + 4) \approx 0.75685, \quad \text{and} \quad c_2 = \frac{1}{5}(4e^3 - 9e^2 + 6e - 1) \approx 5.83007.$$

Solving for the remaining constants gives

$$b_0 = \frac{1}{h_0}(a_1 - a_0) - \frac{h_0}{3}(c_1 + 2c_0)$$
$$= (e - 1) - \frac{1}{15}(-e^3 + 6e^2 - 9e + 4) \approx 1.46600,$$
$$b_1 = \frac{1}{h_1}(a_2 - a_1) - \frac{h_1}{3}(c_2 + 2c_1)$$
$$= (e^2 - e) - \frac{1}{15}(2e^3 + 3e^2 - 12e + 7) \approx 2.22285,$$
$$b_2 = \frac{1}{h_2}(a_3 - a_2) - \frac{h_2}{3}(c_3 + 2c_2)$$
$$= (e^3 - e^2) - \frac{1}{15}(8e^3 - 18e^2 + 12e - 2) \approx 8.80977,$$
$$d_0 = \frac{1}{3h_0}(c_1 - c_0) = \frac{1}{15}(-e^3 + 6e^2 - 9e + 4) \approx 0.25228,$$
$$d_1 = \frac{1}{3h_1}(c_2 - c_1) = \frac{1}{3}(e^3 - 3e^2 + 3e - 1) \approx 1.69107,$$

and

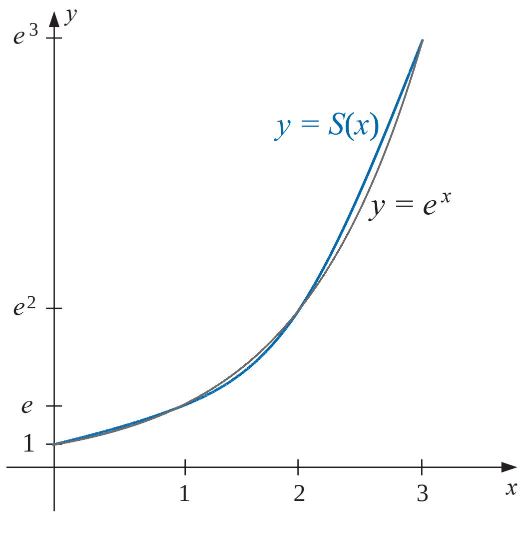
$$d_2 = \frac{1}{3h_2}(c_3 - c_1) = \frac{1}{15}(-4e^3 + 9e^2 - 6e + 1) \approx -1.94336.$$

The natural cubic spine is described piecewise by

$$S(x) = \begin{cases} 1 + 1.46600x + 0.25228x^3, & \text{for } x \in [0, 1], \\ 2.71828 + 2.22285(x-1) + 0.75685(x-1)^2 + 1.69107(x-1)^3, & \text{for } x \in [1, 2], \\ 7.38906 + 8.80977(x-2) + 5.83007(x-2)^2 - 1.94336(x-2)^3, & \text{for } x \in [2, 3]. \end{cases}$$

The spline and its agreement with $f(x) = e^x$ are shown in Figure 3.10. ■

Figure 3.10



The *NumericalAnalysis* package can be used to create a cubic spline in a manner similar to other constructions in this chapter. However, the *CurveFitting* Package in Maple can also be used, and since this has not been discussed previously we will use it to create the natural spline in Example 2. First we load the package with the command

$with(CurveFitting)$

and define the function being approximated with

$f := x \to e^x$

To create a spline we need to specify the nodes, variable, the degree, and the natural endpoints. This is done with

$sn := t \to Spline([[0., 1.0], [1.0, f(1.0)], [2.0, f(2.0)], [3.0, f(3.0)]], t, degree = 3,$ $endpoints = \text{'}natural\text{'})$

Maple returns

$$t \to CurveFitting{:}\text{-}Spline([[0., 1.0], [1.0, f(1.0)], [2.0, f(2.0)], [3.0, f(3.0)]], t, degree = 3, endpoints = \text{'}natural\text{'})$$

The form of the natural spline is seen with the command

$sn(t)$

which produces

$$\begin{cases} 1. + 1.465998t^2 + 0.2522848t^3 & t < 1.0 \\ 0.495432 + 2.22285t + 0.756853(t-1.0)^2 + 1.691071(t-1.0)^3 & t < 2.0 \\ -10.230483 + 8.809770t + 5.830067(t-2.0)^2 - 1.943356(t-2.0)^3 & otherwise \end{cases}$$

Once we have determined a spline approximation for a function we can use it to approximate other properties of the function. The next illustration involves the integral of the spline we found in the previous example.

Illustration To approximate the integral of $f(x) = e^x$ on $[0, 3]$, which has the value

$$\int_0^3 e^x\,dx = e^3 - 1 \approx 20.08553692 - 1 = 19.08553692,$$

we can piecewise integrate the spline that approximates f on this integral. This gives

$$\int_0^3 S(x) = \int_0^1 1 + 1.46600x + 0.25228x^3\,dx$$
$$+ \int_1^2 2.71828 + 2.22285(x-1) + 0.75685(x-1)^2 + 1.69107(x-1)^3\,dx$$
$$+ \int_2^3 7.38906 + 8.80977(x-2) + 5.83007(x-2)^2 - 1.94336(x-2)^3\,dx.$$

Integrating and collecting values from like powers gives

$$\int_0^3 S(x) = \left[x + 1.46600\frac{x^2}{2} + 0.25228\frac{x^4}{4}\right]_0^1$$

$$+ \left[2.71828(x-1) + 2.22285\frac{(x-1)^2}{2} + 0.75685\frac{(x-1)^3}{3} + 1.69107\frac{(x-1)^4}{4}\right]_1^2$$

$$+ \left[7.38906(x-2) + 8.80977\frac{(x-2)^2}{2} + 5.83007\frac{(x-2)^3}{3} - 1.94336\frac{(x-2)^4}{4}\right]_2^3$$

$$= (1 + 2.71828 + 7.38906) + \frac{1}{2}(1.46600 + 2.22285 + 8.80977)$$

$$+ \frac{1}{3}(0.75685 + 5.83007) + \frac{1}{4}(0.25228 + 1.69107 - 1.94336)$$

$$= 19.55229.$$

Because the nodes are equally spaced in this example the integral approximation is simply

$$\int_0^3 S(x)\,dx = (a_0+a_1+a_2)+\frac{1}{2}(b_0+b_1+b_2)+\frac{1}{3}(c_0+c_1+c_2)+\frac{1}{4}(d_0+d_1+d_2). \quad (3.22)$$

□

If we create the natural spline using Maple as described after Example 2, we can then use Maple's integration command to find the value in the Illustration. Simply enter

$int(sn(t), t = 0\,..\,3)$

19.55228648

Clamped Splines

Example 3 In Example 1 we found a natural spline S that passes through the points $(1, 2)$, $(2, 3)$, and $(3, 5)$. Construct a clamped spline s through these points that has $s'(1) = 2$ and $s'(3) = 1$.

Solution Let

$$s_0(x) = a_0 + b_0(x-1) + c_0(x-1)^2 + d_0(x-1)^3,$$

be the cubic on $[1, 2]$ and the cubic on $[2, 3]$ be

$$s_1(x) = a_1 + b_1(x-2) + c_1(x-2)^2 + d_1(x-2)^3.$$

Then most of the conditions to determine the 8 constants are the same as those in Example 1. That is,

$$2 = f(1) = a_0, \quad 3 = f(2) = a_0 + b_0 + c_0 + d_0, \quad 3 = f(2) = a_1, \quad \text{and}$$

$$5 = f(3) = a_1 + b_1 + c_1 + d_1.$$

$$s_0'(2) = s_1'(2): \quad b_0 + 2c_0 + 3d_0 = b_1 \quad \text{and} \quad s_0''(2) = s_1''(2): \quad 2c_0 + 6d_0 = 2c_1$$

However, the boundary conditions are now

$$s_0'(1) = 2: \quad b_0 = 2 \quad \text{and} \quad s_1'(3) = 1: \quad b_1 + 2c_1 + 3d_1 = 1.$$

Solving this system of equations gives the spline as

$$s(x) = \begin{cases} 2 + 2(x-1) - \frac{5}{2}(x-1)^2 + \frac{3}{2}(x-1)^3, & \text{for } x \in [1,2] \\ 3 + \frac{3}{2}(x-2) + 2(x-2)^2 - \frac{3}{2}(x-2)^3, & \text{for } x \in [2,3] \end{cases}$$ ■

In the case of general clamped boundary conditions we have a result that is similar to the theorem for natural boundary conditions described in Theorem 3.11.

Theorem 3.12 If f is defined at $a = x_0 < x_1 < \cdots < x_n = b$ and differentiable at a and b, then f has a unique clamped spline interpolant S on the nodes $x_0, x_1, \ldots, x_n$; that is, a spline interpolant that satisfies the clamped boundary conditions $S'(a) = f'(a)$ and $S'(b) = f'(b)$. ■

Proof Since $f'(a) = S'(a) = S'(x_0) = b_0$, Eq. (3.20) with $j = 0$ implies

$$f'(a) = \frac{1}{h_0}(a_1 - a_0) - \frac{h_0}{3}(2c_0 + c_1).$$

Consequently,

$$2h_0c_0 + h_0c_1 = \frac{3}{h_0}(a_1 - a_0) - 3f'(a).$$

Similarly,

$$f'(b) = b_n = b_{n-1} + h_{n-1}(c_{n-1} + c_n),$$

so Eq. (3.20) with $j = n - 1$ implies that

$$\begin{aligned} f'(b) &= \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{h_{n-1}}{3}(2c_{n-1} + c_n) + h_{n-1}(c_{n-1} + c_n) \\ &= \frac{a_n - a_{n-1}}{h_{n-1}} + \frac{h_{n-1}}{3}(c_{n-1} + 2c_n), \end{aligned}$$

and

$$h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}).$$

Equations (3.21) together with the equations

$$2h_0c_0 + h_0c_1 = \frac{3}{h_0}(a_1 - a_0) - 3f'(a)$$

and

$$h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1})$$

determine the linear system $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{bmatrix} 2h_0 & h_0 & 0 & \cdots & \cdots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & \ddots & & \vdots \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & \cdots & \cdots & 0 & h_{n-1} & 2h_{n-1} \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} \frac{3}{h_0}(a_1 - a_0) - 3f'(a) \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 3f'(b) - \frac{3}{h_{n-1}}(a_n - a_{n-1}) \end{bmatrix}, \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}.$$

This matrix A is also strictly diagonally dominant, so it satisfies the conditions of Theorem 6.21 in Section 6.6. Therefore, the linear system has a unique solution for $c_0, c_1, \ldots, c_n$. ■ ■ ■

The solution to the cubic spline problem with the boundary conditions $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ can be obtained by applying Algorithm 3.5.

ALGORITHM 3.5

Clamped Cubic Spline

To construct the cubic spline interpolant S for the function f defined at the numbers $x_0 < x_1 < \cdots < x_n$, satisfying $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$:

INPUT n; $x_0, x_1, \ldots, x_n$; $a_0 = f(x_0)$, $a_1 = f(x_1), \ldots, a_n = f(x_n)$; $FPO = f'(x_0)$; $FPN = f'(x_n)$.

OUTPUT a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$.

(Note: $S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$ for $x_j \le x \le x_{j+1}$.)

Step 1 For $i = 0, 1, \ldots, n-1$ set $h_i = x_{i+1} - x_i$.

Step 2 Set $\alpha_0 = 3(a_1 - a_0)/h_0 - 3FPO$;
$\alpha_n = 3FPN - 3(a_n - a_{n-1})/h_{n-1}$.

Step 3 For $i = 1, 2, \ldots, n-1$

$$\text{set } \alpha_i = \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}).$$

Step 4 Set $l_0 = 2h_0$; *(Steps 4,5,6, and part of Step 7 solve a tridiagonal linear system using a method described in Algorithm 6.7.)*
$\mu_0 = 0.5$;
$z_0 = \alpha_0/l_0$.

Step 5 For $i = 1, 2, \ldots, n-1$
set $l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1}\mu_{i-1}$;
$\mu_i = h_i/l_i$;
$z_i = (\alpha_i - h_{i-1}z_{i-1})/l_i$.

Step 6 Set $l_n = h_{n-1}(2 - \mu_{n-1})$;
$z_n = (\alpha_n - h_{n-1}z_{n-1})/l_n$;
$c_n = z_n$.

Step 7 For $j = n-1, n-2, \ldots, 0$
set $c_j = z_j - \mu_j c_{j+1}$;
$b_j = (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$;
$d_j = (c_{j+1} - c_j)/(3h_j)$.

Step 8 OUTPUT (a_j, b_j, c_j, d_j for $j = 0, 1, \ldots, n-1$);
STOP.

■

Example 4 Example 2 used a natural spline and the data points $(0, 1)$, $(1, e)$, $(2, e^2)$, and $(3, e^3)$ to form a new approximating function $S(x)$. Determine the clamped spline $s(x)$ that uses this data and the additional information that, since $f'(x) = e^x$, so $f'(0) = 1$ and $f'(3) = e^3$.

Solution As in Example 2, we have $n = 3$, $h_0 = h_1 = h_2 = 1$, $a_0 = 0$, $a_1 = e$, $a_2 = e^2$, and $a_3 = e^3$. This together with the information that $f'(0) = 1$ and $f'(3) = e^3$ gives the the matrix A and the vectors $\mathbf{b}$ and $\mathbf{x}$ with the forms

$$A = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3(e-2) \\ 3(e^2 - 2e + 1) \\ 3(e^3 - 2e^2 + e) \\ 3e^2 \end{bmatrix}, \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}.$$

The vector-matrix equation $A\mathbf{x} = \mathbf{b}$ is equivalent to the system of equations

$$2c_0 + c_1 = 3(e-2),$$

$$c_0 + 4c_1 + c_2 = 3(e^2 - 2e + 1),$$

$$c_1 + 4c_2 + c_3 = 3(e^3 - 2e^2 + e),$$

$$c_2 + 2c_3 = 3e^2.$$

Solving this system simultaneously for c_0, c_1, c_2 and c_3 gives, to 5 decimal places,

$$c_0 = \frac{1}{15}(2e^3 - 12e^2 + 42e - 59) = 0.44468,$$

$$c_1 = \frac{1}{15}(-4e^3 + 24e^2 - 39e + 28) = 1.26548,$$

$$c_2 = \frac{1}{15}(14e^3 - 39e^2 + 24e - 8) = 3.35087,$$

$$c_3 = \frac{1}{15}(-7e^3 + 42e^2 - 12e + 4) = 9.40815.$$

Solving for the remaining constants in the same manner as Example 2 gives

$$b_0 = 1.00000, \quad b_1 = 2.71016, \quad b_2 = 7.32652,$$

and

$$d_0 = 0.27360, \quad d_1 = 0.69513, \quad d_2 = 2.01909.$$

This gives the clamped cubic spine

$$s(x) = \begin{cases} 1 + x + 0.44468x^2 + 0.27360x^3, & \text{if } 0 \le x < 1, \\ 2.71828 + 2.71016(x-1) + 1.26548(x-1)^2 + 0.69513(x-1)^3, & \text{if } 1 \le x < 2, \\ 7.38906 + 7.32652(x-2) + 3.35087(x-2)^2 + 2.01909(x-2)^3, & \text{if } 2 \le x \le 3. \end{cases}$$

The graph of the clamped spline and $f(x) = e^x$ are so similar that no difference can be seen. ■

We can create the clamped cubic spline in Example 4 with the same commands we used for the natural spline, the only change that is needed is to specify the derivative at the endpoints. In this case we use

$sn := t \to Spline\,([[0., 1.0], [1.0, f(1.0)], [2.0, f(2.0)], [3.0, f(3.0)]], t, degree = 3,$
$endpoints = \left[1.0, e^{3.0}\right])$

giving essentially the same results as in the example.

We can also approximate the integral of f on $[0, 3]$, by integrating the clamped spline. The exact value of the integral is

$$\int_0^3 e^x\, dx = e^3 - 1 \approx 20.08554 - 1 = 19.08554.$$

Because the data is equally spaced, piecewise integrating the clamped spline results in the same formula as in (3.22), that is,

$$\int_0^3 s(x)\, dx = (a_0 + a_1 + a_2) + \frac{1}{2}(b_0 + b_1 + b_2) + \frac{1}{3}(c_0 + c_1 + c_2) + \frac{1}{4}(d_0 + d_1 + d_2).$$

Hence the integral approximation is

$$\begin{aligned} \int_0^3 s(x)\, dx &= (1 + 2.71828 + 7.38906) + \frac{1}{2}(1 + 2.71016 + 7.32652) \\ &\quad + \frac{1}{3}(0.44468 + 1.26548 + 3.35087) + \frac{1}{4}(0.27360 + 0.69513 + 2.01909) \\ &= 19.05965. \end{aligned}$$

The absolute error in the integral approximation using the clamped and natural splines are

$$\text{Natural} : |19.08554 - 19.55229| = 0.46675$$

and

$$\text{Clamped} : |19.08554 - 19.05965| = 0.02589.$$

For integration purposes the clamped spline is vastly superior. This should be no surprise since the boundary conditions for the clamped spline are exact, whereas for the natural spline we are essentially assuming that, since $f''(x) = e^x$,

$$0 = S''(0) \approx f''(0) = e^1 = 1 \quad \text{and} \quad 0 = S''(3) \approx f''(3) = e^3 \approx 20.$$

The next illustration uses a spine to approximate a curve that has no given functional representation.

Illustration Figure 3.11 shows a ruddy duck in flight. To approximate the top profile of the duck, we have chosen points along the curve through which we want the approximating curve to pass. Table 3.18 lists the coordinates of 21 data points relative to the superimposed coordinate system shown in Figure 3.12. Notice that more points are used when the curve is changing rapidly than when it is changing more slowly.

Figure 3.11

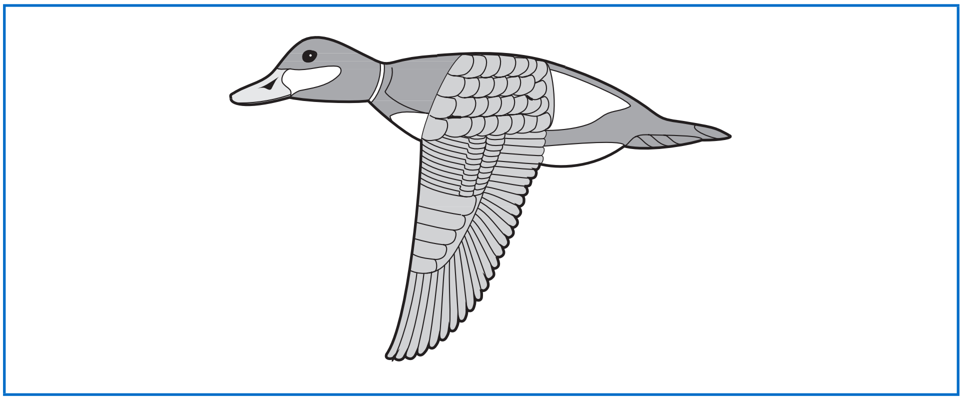
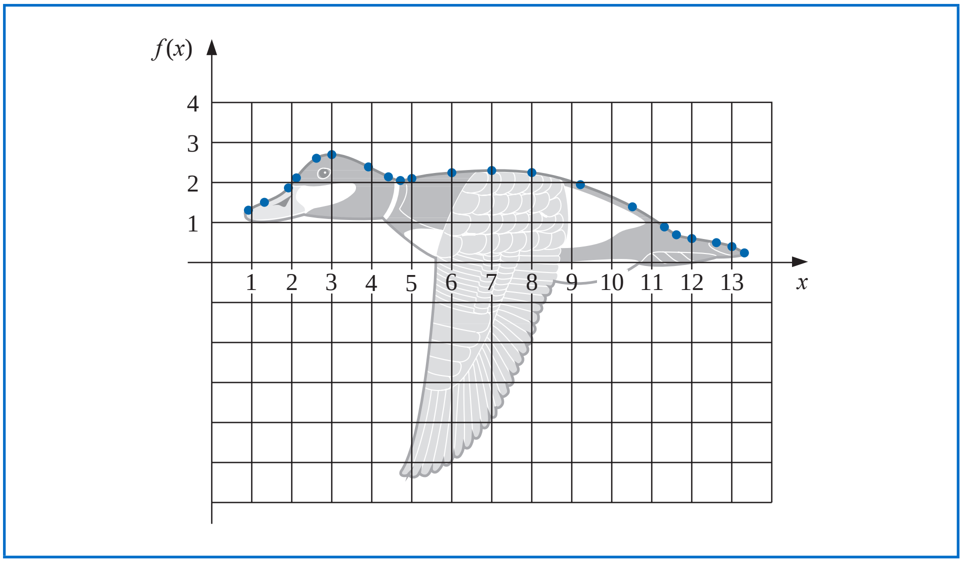


Table 3.18

x	0.9	1.3	1.9	2.1	2.6	3.0	3.9	4.4	4.7	5.0	6.0	7.0	8.0	9.2	10.5	11.3	11.6	12.0	12.6	13.0	13.3
$f(x)$	1.3	1.5	1.85	2.1	2.6	2.7	2.4	2.15	2.05	2.1	2.25	2.3	2.25	1.95	1.4	0.9	0.7	0.6	0.5	0.4	0.25

Figure 3.12

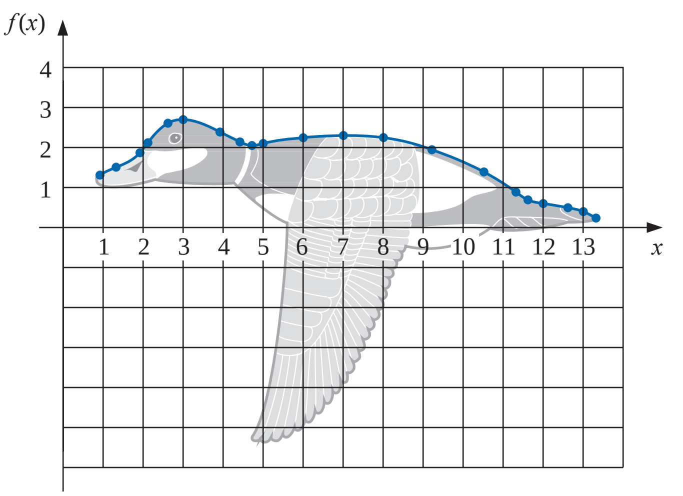


Using Algorithm 3.4 to generate the natural cubic spline for this data produces the coefficients shown in Table 3.19. This spline curve is nearly identical to the profile, as shown in Figure 3.13.

Table 3.19

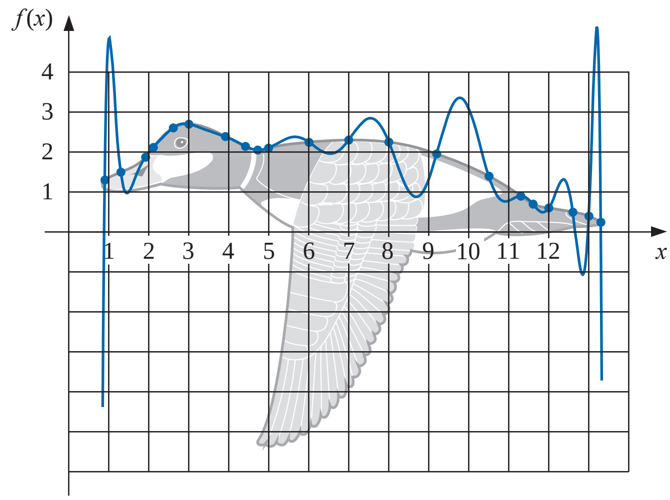
j	x_j	a_j	b_j	c_j	d_j
0	0.9	1.3	5.40	0.00	−0.25
1	1.3	1.5	0.42	−0.30	0.95
2	1.9	1.85	1.09	1.41	−2.96
3	2.1	2.1	1.29	−0.37	−0.45
4	2.6	2.6	0.59	−1.04	0.45
5	3.0	2.7	−0.02	−0.50	0.17
6	3.9	2.4	−0.50	−0.03	0.08
7	4.4	2.15	−0.48	0.08	1.31
8	4.7	2.05	−0.07	1.27	−1.58
9	5.0	2.1	0.26	−0.16	0.04
10	6.0	2.25	0.08	−0.03	0.00
11	7.0	2.3	0.01	−0.04	−0.02
12	8.0	2.25	−0.14	−0.11	0.02
13	9.2	1.95	−0.34	−0.05	−0.01
14	10.5	1.4	−0.53	−0.10	−0.02
15	11.3	0.9	−0.73	−0.15	1.21
16	11.6	0.7	−0.49	0.94	−0.84
17	12.0	0.6	−0.14	−0.06	0.04
18	12.6	0.5	−0.18	0.00	−0.45
19	13.0	0.4	−0.39	−0.54	0.60
20	13.3	0.25			

Figure 3.13



For comparison purposes, Figure 3.14 gives an illustration of the curve that is generated using a Lagrange interpolating polynomial to fit the data given in Table 3.18. The interpolating polynomial in this case is of degree 20 and oscillates wildly. It produces a very strange illustration of the back of a duck, in flight or otherwise.

Figure 3.14



To use a clamped spline to approximate this curve we would need derivative approximations for the endpoints. Even if these approximations were available, we could expect little improvement because of the close agreement of the natural cubic spline to the curve of the top profile. □

Constructing a cubic spline to approximate the lower profile of the ruddy duck would be more difficult since the curve for this portion cannot be expressed as a function of x, and at certain points the curve does not appear to be smooth. These problems can be resolved by using separate splines to represent various portions of the curve, but a more effective approach to approximating curves of this type is considered in the next section.

The clamped boundary conditions are generally preferred when approximating functions by cubic splines, so the derivative of the function must be known or approximated at the endpoints of the interval. When the nodes are equally spaced near both endpoints, approximations can be obtained by any of the appropriate formulas given in Sections 4.1 and 4.2. When the nodes are unequally spaced, the problem is considerably more difficult.

To conclude this section, we list an error-bound formula for the cubic spline with clamped boundary conditions. The proof of this result can be found in [Schul], pp. 57–58.

Theorem 3.13 Let $f \in C^4[a,b]$ with $\max_{a\le x\le b}|f^{(4)}(x)| = M$. If S is the unique clamped cubic spline interpolant to f with respect to the nodes $a = x_0 < x_1 < \cdots < x_n = b$, then for all x in $[a,b]$,

$$|f(x) - S(x)| \le \frac{5M}{384} \max_{0\le j\le n-1} (x_{j+1} - x_j)^4.$$ ■

A fourth-order error-bound result also holds in the case of natural boundary conditions, but it is more difficult to express. (See [BD], pp. 827–835.)

The natural boundary conditions will generally give less accurate results than the clamped conditions near the ends of the interval $[x_0, x_n]$ unless the function f happens to nearly satisfy $f''(x_0) = f''(x_n) = 0$. An alternative to the natural boundary condition that does not require knowledge of the derivative of f is the *not-a-knot* condition, (see [Deb2], pp. 55–56). This condition requires that $S'''(x)$ be continuous at x_1 and at x_{n-1}.

EXERCISE SET 3.5

1. Determine the natural cubic spline S that interpolates the data $f(0) = 0$, $f(1) = 1$, and $f(2) = 2$.

2. Determine the clamped cubic spline s that interpolates the data $f(0) = 0$, $f(1) = 1$, $f(2) = 2$ and satisfies $s'(0) = s'(2) = 1$.

3. Construct the natural cubic spline for the following data.

a.

x	$f(x)$
8.3	17.56492
8.6	18.50515

b.

x	$f(x)$
0.8	0.22363362
1.0	0.65809197

c.

x	$f(x)$
−0.5	−0.0247500
−0.25	0.3349375
0	1.1010000

d.

x	$f(x)$
0.1	−0.62049958
0.2	−0.28398668
0.3	0.00660095
0.4	0.24842440

4. Construct the natural cubic spline for the following data.

a.

x	$f(x)$
0	1.00000
0.5	2.71828

b.

x	$f(x)$
−0.25	1.33203
0.25	0.800781

c.

x	$f(x)$
0.1	−0.29004996
0.2	−0.56079734
0.3	−0.81401972

d.

x	$f(x)$
−1	0.86199480
−0.5	0.95802009
0	1.0986123
0.5	1.2943767

5. The data in Exercise 3 were generated using the following functions. Use the cubic splines constructed in Exercise 3 for the given value of x to approximate $f(x)$ and $f'(x)$, and calculate the actual error.

a. $f(x) = x \ln x$; approximate $f(8.4)$ and $f'(8.4)$.

b. $f(x) = \sin(e^x - 2)$; approximate $f(0.9)$ and $f'(0.9)$.

c. $f(x) = x^3 + 4.001x^2 + 4.002x + 1.101$; approximate $f(-\frac{1}{3})$ and $f'(-\frac{1}{3})$.

d. $f(x) = x \cos x - 2x^2 + 3x - 1$; approximate $f(0.25)$ and $f'(0.25)$.

6. The data in Exercise 4 were generated using the following functions. Use the cubic splines constructed in Exercise 4 for the given value of x to approximate $f(x)$ and $f'(x)$, and calculate the actual error.

a. $f(x) = e^{2x}$; approximate $f(0.43)$ and $f'(0.43)$.

b. $f(x) = x^4 - x^3 + x^2 - x + 1$; approximate $f(0)$ and $f'(0)$.

c. $f(x) = x^2 \cos x - 3x$; approximate $f(0.18)$ and $f'(0.18)$.

d. $f(x) = \ln(e^x + 2)$; approximate $f(0.25)$ and $f'(0.25)$.

7. Construct the clamped cubic spline using the data of Exercise 3 and the fact that

a. $f'(8.3) = 3.116256$ and $f'(8.6) = 3.151762$

b. $f'(0.8) = 2.1691753$ and $f'(1.0) = 2.0466965$

c. $f'(-0.5) = 0.7510000$ and $f'(0) = 4.0020000$

d. $f'(0.1) = 3.58502082$ and $f'(0.4) = 2.16529366$

8. Construct the clamped cubic spline using the data of Exercise 4 and the fact that

a. $f'(0) = 2$ and $f'(0.5) = 5.43656$

b. $f'(-0.25) = 0.437500$ and $f'(0.25) = -0.625000$

c. $f'(0.1) = -2.8004996$ and $f'(0) = -2.9734038$

d. $f'(-1) = 0.15536240$ and $f'(0.5) = 0.45186276$

9. Repeat Exercise 5 using the clamped cubic splines constructed in Exercise 7.

10. Repeat Exercise 6 using the clamped cubic splines constructed in Exercise 8.

11. A natural cubic spline S on $[0, 2]$ is defined by

$$S(x) = \begin{cases} S_0(x) = 1 + 2x - x^3, & \text{if } 0 \le x < 1, \\ S_1(x) = 2 + b(x-1) + c(x-1)^2 + d(x-1)^3, & \text{if } 1 \le x \le 2. \end{cases}$$

Find b, c, and d.

12. A clamped cubic spline s for a function f is defined on $[1, 3]$ by

$$s(x) = \begin{cases} s_0(x) = 3(x-1) + 2(x-1)^2 - (x-1)^3, & \text{if } 1 \le x < 2, \\ s_1(x) = a + b(x-2) + c(x-2)^2 + d(x-2)^3, & \text{if } 2 \le x \le 3. \end{cases}$$

Given $f'(1) = f'(3)$, find a, b, c, and d.

13. A natural cubic spline S is defined by

$$S(x) = \begin{cases} S_0(x) = 1 + B(x-1) - D(x-1)^3, & \text{if } 1 \le x < 2, \\ S_1(x) = 1 + b(x-2) - \frac{3}{4}(x-2)^2 + d(x-2)^3, & \text{if } 2 \le x \le 3. \end{cases}$$

If S interpolates the data $(1, 1)$, $(2, 1)$, and $(3, 0)$, find B, D, b, and d.

14. A clamped cubic spline s for a function f is defined by

$$s(x) = \begin{cases} s_0(x) = 1 + Bx + 2x^2 - 2x^3, & \text{if } 0 \le x < 1, \\ s_1(x) = 1 + b(x-1) - 4(x-1)^2 + 7(x-1)^3, & \text{if } 1 \le x \le 2. \end{cases}$$

Find $f'(0)$ and $f'(2)$.

15. Construct a natural cubic spline to approximate $f(x) = \cos \pi x$ by using the values given by $f(x)$ at $x = 0, 0.25, 0.5, 0.75$, and 1.0. Integrate the spline over $[0, 1]$, and compare the result to $\int_0^1 \cos \pi x \, dx = 0$. Use the derivatives of the spline to approximate $f'(0.5)$ and $f''(0.5)$. Compare these approximations to the actual values.

16. Construct a natural cubic spline to approximate $f(x) = e^{-x}$ by using the values given by $f(x)$ at $x = 0$, 0.25, 0.75, and 1.0. Integrate the spline over $[0, 1]$, and compare the result to $\int_0^1 e^{-x} \, dx = 1 - 1/e$. Use the derivatives of the spline to approximate $f'(0.5)$ and $f''(0.5)$. Compare the approximations to the actual values.

17. Repeat Exercise 15, constructing instead the clamped cubic spline with $f'(0) = f'(1) = 0$.

18. Repeat Exercise 16, constructing instead the clamped cubic spline with $f'(0) = -1$, $f'(1) = -e^{-1}$.

19. Suppose that $f(x)$ is a polynomial of degree 3. Show that $f(x)$ is its own clamped cubic spline, but that it cannot be its own natural cubic spline.

20. Suppose the data $\{x_i, f(x_i)\}_{i=1}^n$ lie on a straight line. What can be said about the natural and clamped cubic splines for the function f? [*Hint:* Take a cue from the results of Exercises 1 and 2.]

21. Given the partition $x_0 = 0$, $x_1 = 0.05$, and $x_2 = 0.1$ of $[0, 0.1]$, find the piecewise linear interpolating function F for $f(x) = e^{2x}$. Approximate $\int_0^{0.1} e^{2x} \, dx$ with $\int_0^{0.1} F(x) \, dx$, and compare the results to the actual value.

22. Let $f \in C^2[a, b]$, and let the nodes $a = x_0 < x_1 < \cdots < x_n = b$ be given. Derive an error estimate similar to that in Theorem 3.13 for the piecewise linear interpolating function F. Use this estimate to derive error bounds for Exercise 21.

23. Extend Algorithms 3.4 and 3.5 to include as output the first and second derivatives of the spline at the nodes.

24. Extend Algorithms 3.4 and 3.5 to include as output the integral of the spline over the interval $[x_0, x_n]$.

25. Given the partition $x_0 = 0$, $x_1 = 0.05$, $x_2 = 0.1$ of $[0, 0.1]$ and $f(x) = e^{2x}$:

a. Find the cubic spline s with clamped boundary conditions that interpolates f.

b. Find an approximation for $\int_0^{0.1} e^{2x} \, dx$ by evaluating $\int_0^{0.1} s(x) \, dx$.

c. Use Theorem 3.13 to estimate $\max_{0\leq x\leq 0.1}|f(x)-s(x)|$ and

$$\left|\int_0^{0.1} f(x)\,dx - \int_0^{0.1} s(x)\,dx\right|.$$

d. Determine the cubic spline S with natural boundary conditions, and compare $S(0.02)$, $s(0.02)$, and $e^{0.04} = 1.04081077$.

26. Let f be defined on $[a, b]$, and let the nodes $a = x_0 < x_1 < x_2 = b$ be given. A quadratic spline interpolating function S consists of the quadratic polynomial

$$S_0(x) = a_0 + b_0(x - x_0) + c_0(x - x_0)^2 \quad \text{on } [x_0, x_1]$$

and the quadratic polynomial

$$S_1(x) = a_1 + b_1(x - x_1) + c_1(x - x_1)^2 \quad \text{on } [x_1, x_2],$$

such that

i. $S(x_0) = f(x_0), S(x_1) = f(x_1)$, and $S(x_2) = f(x_2)$,

ii. $S \in C^1[x_0, x_2]$.

Show that conditions (i) and (ii) lead to five equations in the six unknowns a_0, b_0, c_0, a_1, b_1, and c_1. The problem is to decide what additional condition to impose to make the solution unique. Does the condition $S \in C^2[x_0, x_2]$ lead to a meaningful solution?

27. Determine a quadratic spline s that interpolates the data $f(0) = 0$, $f(1) = 1$, $f(2) = 2$ and satisfies $s'(0) = 2$.

28. a. The introduction to this chapter included a table listing the population of the United States from 1950 to 2000. Use natural cubic spline interpolation to approximate the population in the years 1940, 1975, and 2020.

b. The population in 1940 was approximately 132,165,000. How accurate do you think your 1975 and 2020 figures are?

29. A car traveling along a straight road is clocked at a number of points. The data from the observations are given in the following table, where the time is in seconds, the distance is in feet, and the speed is in feet per second.

Time	0	3	5	8	13
Distance	0	225	383	623	993
Speed	75	77	80	74	72

a. Use a clamped cubic spline to predict the position of the car and its speed when $t = 10$ s.

b. Use the derivative of the spline to determine whether the car ever exceeds a 55-mi/h speed limit on the road; if so, what is the first time the car exceeds this speed?

c. What is the predicted maximum speed for the car?

30. The 2009 Kentucky Derby was won by a horse named Mine That Bird (at more than 50:1 odds) in a time of 2:02.66 (2 minutes and 2.66 seconds) for the $1\frac{1}{4}$-mile race. Times at the quarter-mile, half-mile, and mile poles were 0:22.98, 0:47.23, and 1:37.49.

a. Use these values together with the starting time to construct a natural cubic spline for Mine That Bird's race.

b. Use the spline to predict the time at the three-quarter-mile pole, and compare this to the actual time of 1:12.09.

c. Use the spline to approximate Mine That Bird's starting speed and speed at the finish line.

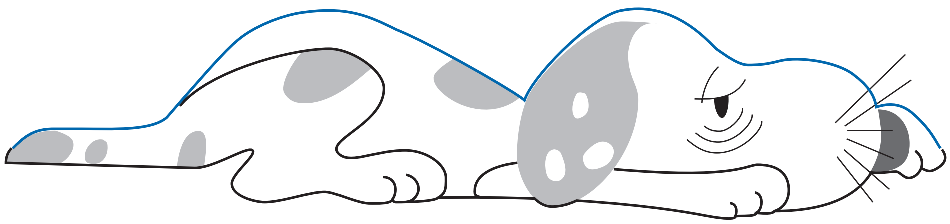
31. It is suspected that the high amounts of tannin in mature oak leaves inhibit the growth of the winter moth (*Operophtera bromata L., Geometridae*) larvae that extensively damage these trees in certain years. The following table lists the average weight of two samples of larvae at times in the first 28 days after birth. The first sample was reared on young oak leaves, whereas the second sample was reared on mature leaves from the same tree.

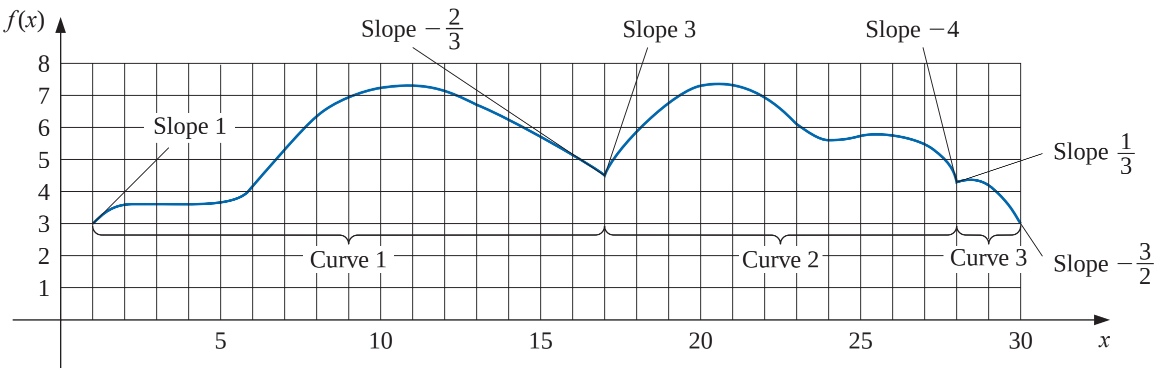
a. Use a natural cubic spline to approximate the average weight curve for each sample.

b. Find an approximate maximum average weight for each sample by determining the maximum of the spline.

Day	0	6	10	13	17	20	28
Sample 1 average weight (mg)	6.67	17.33	42.67	37.33	30.10	29.31	28.74
Sample 2 average weight (mg)	6.67	16.11	18.89	15.00	10.56	9.44	8.89

32. The upper portion of this noble beast is to be approximated using clamped cubic spline interpolants. The curve is drawn on a grid from which the table is constructed. Use Algorithm 3.5 to construct the three clamped cubic splines.





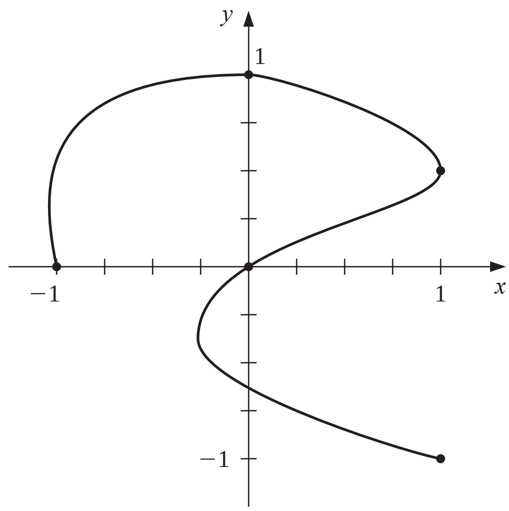
Curve 1				Curve 2				Curve 3			
i	x_i	$f(x_i)$	$f'(x_i)$	i	x_i	$f(x_i)$	$f'(x_i)$	i	x_i	$f(x_i)$	$f'(x_i)$
0	1	3.0	1.0	0	17	4.5	3.0	0	27.7	4.1	0.33
1	2	3.7		1	20	7.0		1	28	4.3	
2	5	3.9		2	23	6.1		2	29	4.1	
3	6	4.2		3	24	5.6		3	30	3.0	−1.5
4	7	5.7		4	25	5.8					
5	8	6.6		5	27	5.2					
6	10	7.1		6	27.7	4.1	−4.0				
7	13	6.7									
8	17	4.5	−0.67								

33. Repeat Exercise 32, constructing three natural splines using Algorithm 3.4.

3.6 Parametric Curves

None of the techniques developed in this chapter can be used to generate curves of the form shown in Figure 3.15 because this curve cannot be expressed as a function of one coordinate variable in terms of the other. In this section we will see how to represent general curves by using a parameter to express both the x- and y-coordinate variables. Any good book on computer graphics will show how this technique can be extended to represent general curves and surfaces in space. (See, for example, [FVFH].)

Figure 3.15



A straightforward parametric technique for determining a polynomial or piecewise polynomial to connect the points (x_0, y_0), (x_1, y_1), ..., (x_n, y_n) in the order given is to use a parameter t on an interval $[t_0, t_n]$, with $t_0 < t_1 < \cdots < t_n$, and construct approximation functions with

$$x_i = x(t_i) \quad \text{and} \quad y_i = y(t_i), \quad \text{for each } i = 0, 1, \ldots, n.$$

The following example demonstrates the technique in the case where both approximating functions are Lagrange interpolating polynomials.

Example 1 Construct a pair of Lagrange polynomials to approximate the curve shown in Figure 3.15, using the data points shown on the curve.

Solution There is flexibility in choosing the parameter, and we will choose the points $\{t_i\}_{i=0}^4$ equally spaced in [0,1], which gives the data in Table 3.20.

Table 3.20

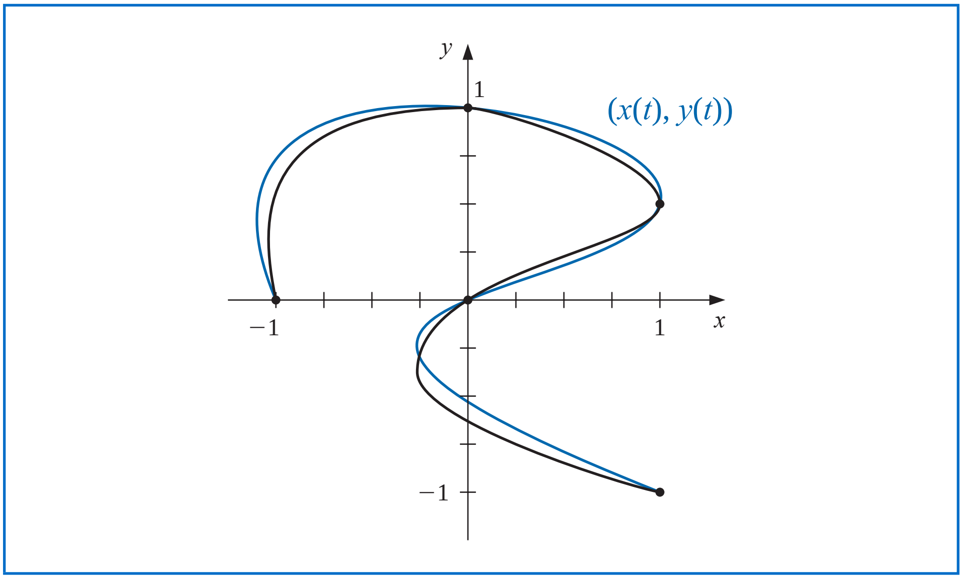
i	0	1	2	3	4
t_i	0	0.25	0.5	0.75	1
x_i	−1	0	1	0	1
y_i	0	1	0.5	0	−1

This produces the interpolating polynomials

$$x(t) = \left(\left(\left(64t - \frac{352}{3}\right)t + 60\right)t - \frac{14}{3}\right)t - 1 \quad \text{and} \quad y(t) = \left(\left(\left(-\frac{64}{3}t + 48\right)t - \frac{116}{3}\right)t + 11\right)t.$$

Plotting this parametric system produces the graph shown in blue in Figure 3.16. Although it passes through the required points and has the same basic shape, it is quite a crude approximation to the original curve. A more accurate approximation would require additional nodes, with the accompanying increase in computation. ■

Figure 3.16



Parametric Hermite and spline curves can be generated in a similar manner, but these also require extensive computational effort.

Applications in computer graphics require the rapid generation of smooth curves that can be easily and quickly modified. For both aesthetic and computational reasons, changing one portion of these curves should have little or no effect on other portions of the curves. This eliminates the use of interpolating polynomials and splines since changing one portion of these curves affects the whole curve.

A successful computer design system needs to be based on a formal mathematical theory so that the results are predictable, but this theory should be performed in the background so that the artist can base the design on aesthetics.

The choice of curve for use in computer graphics is generally a form of the piecewise cubic Hermite polynomial. Each portion of a cubic Hermite polynomial is completely determined by specifying its endpoints and the derivatives at these endpoints. As a consequence, one portion of the curve can be changed while leaving most of the curve the same. Only the adjacent portions need to be modified to ensure smoothness at the endpoints. The computations can be performed quickly, and the curve can be modified a section at a time.

The problem with Hermite interpolation is the need to specify the derivatives at the endpoints of each section of the curve. Suppose the curve has $n+1$ data points $(x(t_0), y(t_0)), \ldots, (x(t_n), y(t_n))$, and we wish to parameterize the cubic to allow complex features. Then we must specify $x'(t_i)$ and $y'(t_i)$, for each $i = 0, 1, \ldots, n$. This is not as difficult as it would first appear, since each portion is generated independently. We must ensure only that the derivatives at the endpoints of each portion match those in the adjacent portion. Essentially, then, we can simplify the process to one of determining a pair of cubic Hermite polynomials in the parameter t, where $t_0 = 0$ and $t_1 = 1$, given the endpoint data $(x(0), y(0))$ and $(x(1), y(1))$ and the derivatives dy/dx (at $t = 0$) and dy/dx (at $t = 1$).

Notice, however, that we are specifying only six conditions, and the cubic polynomials in $x(t)$ and $y(t)$ each have four parameters, for a total of eight. This provides flexibility in choosing the pair of cubic Hermite polynomials to satisfy the conditions, because the natural form for determining $x(t)$ and $y(t)$ requires that we specify $x'(0)$, $x'(1)$, $y'(0)$, and $y'(1)$. The explicit Hermite curve in x and y requires specifying only the quotients

$$\frac{dy}{dx}(t=0) = \frac{y'(0)}{x'(0)} \quad \text{and} \quad \frac{dy}{dx}(t=1) = \frac{y'(1)}{x'(1)}.$$

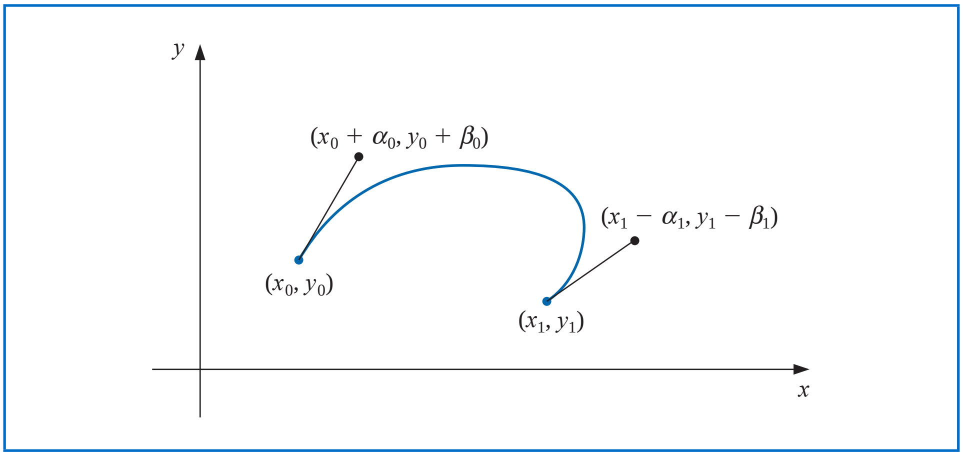
By multiplying $x'(0)$ and $y'(0)$ by a common scaling factor, the tangent line to the curve at $(x(0), y(0))$ remains the same, but the shape of the curve varies. The larger the scaling factor, the closer the curve comes to approximating the tangent line near $(x(0), y(0))$. A similar situation exists at the other endpoint $(x(1), y(1))$.

To further simplify the process in interactive computer graphics, the derivative at an endpoint is specified by using a second point, called a *guidepoint*, on the desired tangent line. The farther the guidepoint is from the node, the more closely the curve approximates the tangent line near the node.

In Figure 3.17, the nodes occur at (x_0, y_0) and (x_1, y_1), the guidepoint for (x_0, y_0) is $(x_0 + \alpha_0, y_0 + \beta_0)$, and the guidepoint for (x_1, y_1) is $(x_1 - \alpha_1, y_1 - \beta_1)$. The cubic Hermite polynomial $x(t)$ on $[0, 1]$ satisfies

$$x(0) = x_0, \quad x(1) = x_1, \quad x'(0) = \alpha_0, \quad \text{and} \quad x'(1) = \alpha_1.$$

Figure 3.17



The unique cubic polynomial satisfying these conditions is

$$x(t) = [2(x_0 - x_1) + (\alpha_0 + \alpha_1)]t^3 + [3(x_1 - x_0) - (\alpha_1 + 2\alpha_0)]t^2 + \alpha_0 t + x_0. \tag{3.23}$$

In a similar manner, the unique cubic polynomial satisfying

$$y(0) = y_0, \quad y(1) = y_1, \quad y'(0) = \beta_0, \quad \text{and} \quad y'(1) = \beta_1$$

is

$$y(t) = [2(y_0 - y_1) + (\beta_0 + \beta_1)]t^3 + [3(y_1 - y_0) - (\beta_1 + 2\beta_0)]t^2 + \beta_0 t + y_0. \tag{3.24}$$

Example 2 Determine the graph of the parametric curve generated Eq. (3.23) and (3.24) when the end points are $(x_0, y_0) = (0, 0)$ and $(x_1, y_1) = (1, 0)$, and respective guide points, as shown in Figure 3.18 are $(1, 1)$ and $(0, 1)$.

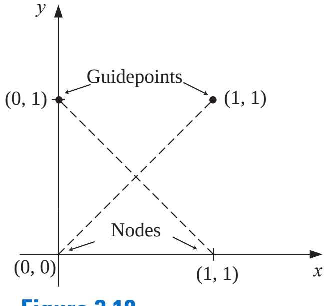


Figure 3.18

Solution The endpoint information implies that $x_0 = 0$, $x_1 = 1$, $y_0 = 0$, and $y_1 = 0$, and the guide points at $(1, 1)$ and $(0, 1)$ imply that $\alpha_0 = 1$, $\alpha_1 = 1$, $\beta_0 = 1$, and $\beta_1 = -1$. Note that the slopes of the guide lines at $(0, 0)$ and $(1, 0)$ are, respectively

$$\frac{\beta_0}{\alpha_0} = \frac{1}{1} = 1 \quad \text{and} \quad \frac{\beta_1}{\alpha_1} = \frac{-1}{1} = -1.$$

Equations (3.23) and (3.24) imply that for $t \in [0, 1]$ we have

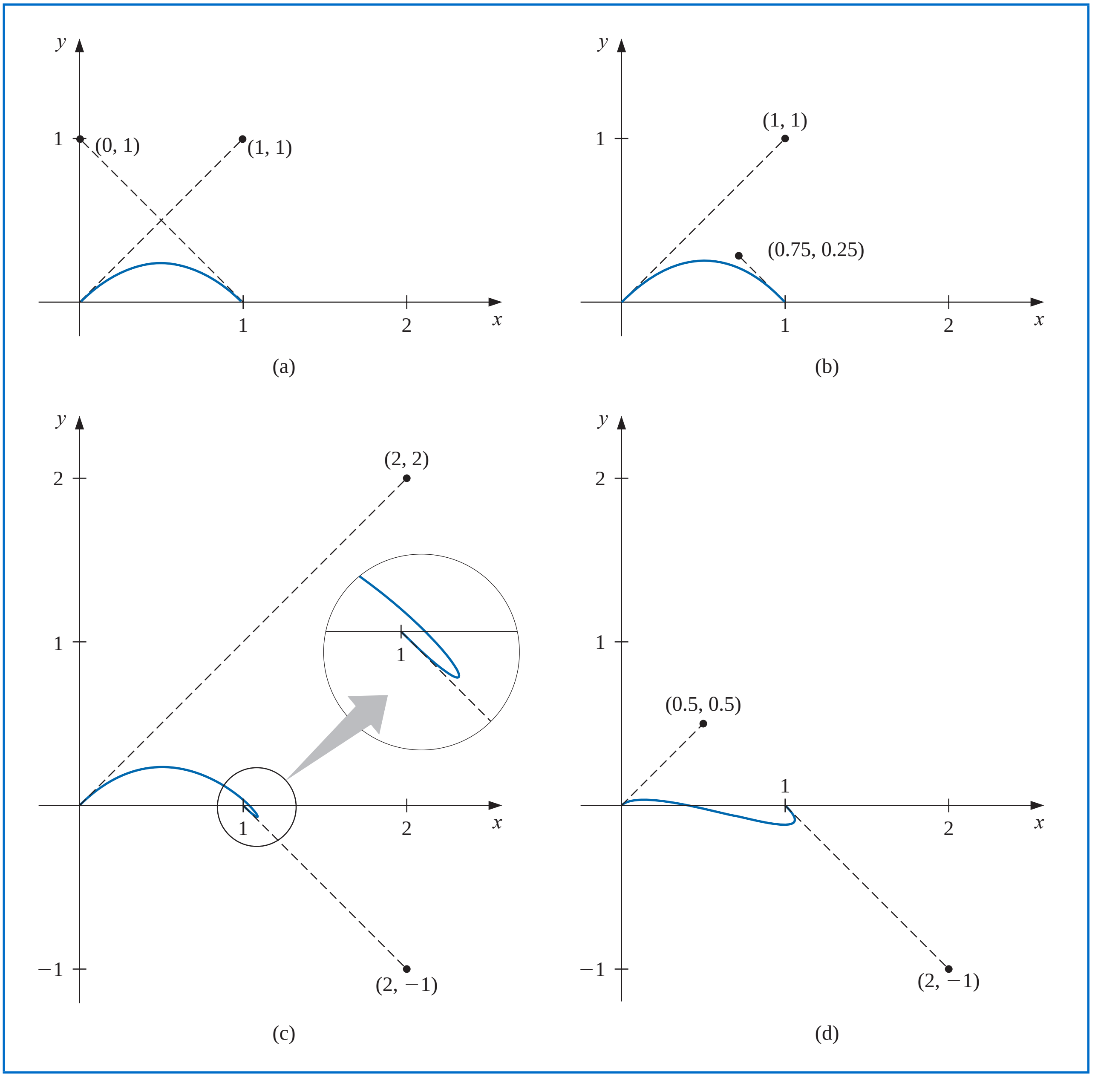
$$x(t) = [2(0 - 1) + (1 + 1)]t^3 + [3(0 - 0) - (1 + 2 \cdot 1)]t^2 + 1 \cdot t + 0 = t$$

and

$$y(t) = [2(0-0) + (1+(-1))]t^3 + [3(0-0) - (-1+2\cdot 1)]t^2 + 1\cdot t + 0 = -t^2 + t.$$

This graph is shown as (a) in Figure 3.19, together with some other possibilities of curves produced by Eqs. (3.23) and (3.24) when the nodes are $(0,0)$ and $(1,0)$ and the slopes at these nodes are 1 and -1, respectively. ■

Figure 3.19



Pierre Etienne Bézier (1910–1999) was head of design and production for Renault motorcars for most of his professional life. He began his research into computer-aided design and manufacturing in 1960, developing interactive tools for curve and surface design, and initiated computer-generated milling for automobile modeling.

The Bézier curves that bear his name have the advantage of being based on a rigorous mathematical theory that does not need to be explicitly recognized by the practitioner who simply wants to make an aesthetically pleasing curve or surface. These are the curves that are the basis of the powerful Adobe Postscript system, and produce the freehand curves that are generated in most sufficiently powerful computer graphics packages.

The standard procedure for determining curves in an interactive graphics mode is to first use a mouse or touchpad to set the nodes and guidepoints to generate a first approximation to the curve. These can be set manually, but most graphics systems permit you to use your input device to draw the curve on the screen freehand and will select appropriate nodes and guidepoints for your freehand curve.

The nodes and guidepoints can then be manipulated into a position that produces an aesthetically pleasing curve. Since the computation is minimal, the curve can be determined so quickly that the resulting change is seen immediately. Moreover, all the data needed to compute the curves are imbedded in the coordinates of the nodes and guidepoints, so no analytical knowledge is required of the user.

Popular graphics programs use this type of system for their freehand graphic representations in a slightly modified form. The Hermite cubics are described as **Bézier polynomials**, which incorporate a scaling factor of 3 when computing the derivatives at the endpoints. This modifies the parametric equations to

$$x(t) = [2(x_0 - x_1) + 3(\alpha_0 + \alpha_1)]t^3 + [3(x_1 - x_0) - 3(\alpha_1 + 2\alpha_0)]t^2 + 3\alpha_0 t + x_0, \quad (3.25)$$

and

$$y(t) = [2(y_0 - y_1) + 3(\beta_0 + \beta_1)]t^3 + [3(y_1 - y_0) - 3(\beta_1 + 2\beta_0)]t^2 + 3\beta_0 t + y_0, \quad (3.26)$$

for $0 \le t \le 1$, but this change is transparent to the user of the system.

Algorithm 3.6 constructs a set of Bézier curves based on the parametric equations in Eqs. (3.25) and (3.26).

ALGORITHM **3.6**

Bézier Curve

To construct the cubic Bézier curves $C_0, \dots, C_{n-1}$ in parametric form, where C_i is represented by

$$(x_i(t), y_i(t)) = (a_0^{(i)} + a_1^{(i)}t + a_2^{(i)}t^2 + a_3^{(i)}t^3, b_0^{(i)} + b_1^{(i)}t + b_2^{(i)}t^2 + b_3^{(i)}t^3),$$

for $0 \le t \le 1$, as determined by the left endpoint (x_i, y_i), left guidepoint (x_i^+, y_i^+), right endpoint (x_{i+1}, y_{i+1}), and right guidepoint (x_{i+1}^-, y_{i+1}^-) for each $i = 0, 1, \dots, n-1$:

INPUT n; $(x_0, y_0), \dots, (x_n, y_n)$; $(x_0^+, y_0^+), \dots, (x_{n-1}^+, y_{n-1}^+)$; $(x_1^-, y_1^-), \dots, (x_n^-, y_n^-)$.

OUTPUT coefficients $\{a_0^{(i)}, a_1^{(i)}, a_2^{(i)}, a_3^{(i)}, b_0^{(i)}, b_1^{(i)}, b_2^{(i)}, b_3^{(i)}$, for $0 \le i \le n-1\}$.

Step 1 For each $i = 0, 1, \dots, n-1$ do Steps 2 and 3.

Step 2 Set $a_0^{(i)} = x_i$;
$b_0^{(i)} = y_i$;
$a_1^{(i)} = 3(x_i^+ - x_i)$;
$b_1^{(i)} = 3(y_i^+ - y_i)$;
$a_2^{(i)} = 3(x_i + x_{i+1}^- - 2x_i^+)$;
$b_2^{(i)} = 3(y_i + y_{i+1}^- - 2y_i^+)$;
$a_3^{(i)} = x_{i+1} - x_i + 3x_i^+ - 3x_{i+1}^-$;
$b_3^{(i)} = y_{i+1} - y_i + 3y_i^+ - 3y_{i+1}^-$;

Step 3 OUTPUT $(a_0^{(i)}, a_1^{(i)}, a_2^{(i)}, a_3^{(i)}, b_0^{(i)}, b_1^{(i)}, b_2^{(i)}, b_3^{(i)})$.

Step 4 STOP. ■

Three-dimensional curves are generated in a similar manner by additionally specifying third components z_0 and z_1 for the nodes and $z_0+\gamma_0$ and $z_1-\gamma_1$ for the guidepoints. The more difficult problem involving the representation of three-dimensional curves concerns the loss of the third dimension when the curve is projected onto a two-dimensional computer screen. Various projection techniques are used, but this topic lies within the realm of computer graphics. For an introduction to this topic and ways that the technique can be modified for surface representations, see one of the many books on computer graphics methods, such as [FVFH].

EXERCISE SET 3.6

1. Let $(x_0, y_0) = (0, 0)$ and $(x_1, y_1) = (5, 2)$ be the endpoints of a curve. Use the given guidepoints to construct parametric cubic Hermite approximations $(x(t), y(t))$ to the curve, and graph the approximations.

a. $(1, 1)$ and $(6, 1)$

b. $(0.5, 0.5)$ and $(5.5, 1.5)$

c. $(1, 1)$ and $(6, 3)$

d. $(2, 2)$ and $(7, 0)$

2. Repeat Exercise 1 using cubic Bézier polynomials.

3. Construct and graph the cubic Bézier polynomials given the following points and guidepoints.

a. Point $(1, 1)$ with guidepoint $(1.5, 1.25)$ to point $(6, 2)$ with guidepoint $(7, 3)$

b. Point $(1, 1)$ with guidepoint $(1.25, 1.5)$ to point $(6, 2)$ with guidepoint $(5, 3)$

c. Point $(0, 0)$ with guidepoint $(0.5, 0.5)$ to point $(4, 6)$ with entering guidepoint $(3.5, 7)$ and exiting guidepoint $(4.5, 5)$ to point $(6, 1)$ with guidepoint $(7, 2)$

d. Point $(0, 0)$ with guidepoint $(0.5, 0.5)$ to point $(2, 1)$ with entering guidepoint $(3, 1)$ and exiting guidepoint $(3, 1)$ to point $(4, 0)$ with entering guidepoint $(5, 1)$ and exiting guidepoint $(3, -1)$ to point $(6, -1)$ with guidepoint $(6.5, -0.25)$

4. Use the data in the following table and Algorithm 3.6 to approximate the shape of the letter $\mathcal{N}$.

i	x_i	y_i	α_i	β_i	α'_i	β'_i
0	3	6	3.3	6.5		
1	2	2	2.8	3.0	2.5	2.5
2	6	6	5.8	5.0	5.0	5.8
3	5	2	5.5	2.2	4.5	2.5
4	6.5	3			6.4	2.8

5. Suppose a cubic Bézier polynomial is placed through (u_0, v_0) and (u_3, v_3) with guidepoints (u_1, v_1) and (u_2, v_2), respectively.

a. Derive the parametric equations for $u(t)$ and $v(t)$ assuming that

$$u(0) = u_0, \quad u(1) = u_3, \quad u'(0) = u_1 - u_0, \quad u'(1) = u_3 - u_2$$

and

$$v(0) = v_0, \quad v(1) = v_3, \quad v'(0) = v_1 - v_0, \quad v'(1) = v_3 - v_2.$$

b. Let $f(i/3) = u_i$, for $i = 0, 1, 2, 3$ and $g(i/3) = v_i$, for $i = 0, 1, 2, 3$. Show that the Bernstein polynomial of degree 3 in t for f is $u(t)$ and the Bernstein polynomial of degree three in t for g is $v(t)$. (See Exercise 23 of Section 3.1.)

3.7 Survey of Methods and Software

In this chapter we have considered approximating a function using polynomials and piecewise polynomials. The function can be specified by a given defining equation or by providing points in the plane through which the graph of the function passes. A set of nodes $x_0, x_1, \ldots, x_n$ is given in each case, and more information, such as the value of various derivatives, may also be required. We need to find an approximating function that satisfies the conditions specified by these data.

The interpolating polynomial $P(x)$ is the polynomial of least degree that satisfies, for a function f,

$$P(x_i) = f(x_i), \quad \text{for each } i = 0, 1, \ldots, n.$$

Although this interpolating polynomial is unique, it can take many different forms. The Lagrange form is most often used for interpolating tables when n is small and for deriving formulas for approximating derivatives and integrals. Neville's method is used for evaluating several interpolating polynomials at the same value of x. Newton's forms of the polynomial are more appropriate for computation and are also used extensively for deriving formulas for solving differential equations. However, polynomial interpolation has the inherent weaknesses of oscillation, particularly if the number of nodes is large. In this case there are other methods that can be better applied.

The Hermite polynomials interpolate a function and its derivative at the nodes. They can be very accurate but require more information about the function being approximated. When there are a large number of nodes, the Hermite polynomials also exhibit oscillation weaknesses.

The most commonly used form of interpolation is piecewise-polynomial interpolation. If function and derivative values are available, piecewise cubic Hermite interpolation is recommended. This is the preferred method for interpolating values of a function that is the solution to a differential equation. When only the function values are available, natural cubic spline interpolation can be used. This spline forces the second derivative of the spline to be zero at the endpoints. Other cubic splines require additional data. For example, the clamped cubic spline needs values of the derivative of the function at the endpoints of the interval.

Other methods of interpolation are commonly used. Trigonometric interpolation, in particular the Fast Fourier Transform discussed in Chapter 8, is used with large amounts of data when the function is assumed to have a periodic nature. Interpolation by rational functions is also used.

If the data are suspected to be inaccurate, smoothing techniques can be applied, and some form of least squares fit of data is recommended. Polynomials, trigonometric functions, rational functions, and splines can be used in least squares fitting of data. We consider these topics in Chapter 8.

Interpolation routines included in the IMSL Library are based on the book *A Practical Guide to Splines* by Carl de Boor [Deb] and use interpolation by cubic splines. There are cubic splines to minimize oscillations and to preserve concavity. Methods for two-dimensional interpolation by bicubic splines are also included.

The NAG library contains subroutines for polynomial and Hermite interpolation, for cubic spline interpolation, and for piecewise cubic Hermite interpolation. NAG also contains subroutines for interpolating functions of two variables.

The netlib library contains the subroutines to compute the cubic spline with various endpoint conditions. One package produces the Newton's divided difference coefficients for a discrete set of data points, and there are various routines for evaluating Hermite piecewise polynomials.

MATLAB can be used to interpolate a discrete set of data points, using either nearest neighbor interpolation, linear interpolation, cubic spline interpolation, or cubic interpolation. Cubic splines can also be produced.

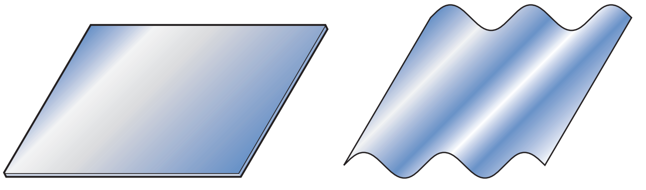
General references to the methods in this chapter are the books by Powell [Pow] and by Davis [Da]. The seminal paper on splines is due to Schoenberg [Scho]. Important books on splines are by Schultz [Schul], De Boor [Deb2], Dierckx [Di], and Schumaker [Schum].

CHAPTER

4 Numerical Differentiation and Integration

Introduction

A sheet of corrugated roofing is constructed by pressing a flat sheet of aluminum into one whose cross section has the form of a sine wave.



A corrugated sheet 4 ft long is needed, the height of each wave is 1 in. from the center line, and each wave has a period of approximately 2π in. The problem of finding the length of the initial flat sheet is one of determining the length of the curve given by $f(x) = \sin x$ from $x = 0$ in. to $x = 48$ in. From calculus we know that this length is

$$L = \int_0^{48} \sqrt{1 + (f'(x))^2}\, dx = \int_0^{48} \sqrt{1 + (\cos x)^2}\, dx,$$

so the problem reduces to evaluating this integral. Although the sine function is one of the most common mathematical functions, the calculation of its length involves an elliptic integral of the second kind, which cannot be evaluated explicitly. Methods are developed in this chapter to approximate the solution to problems of this type. This particular problem is considered in Exercise 25 of Section 4.4 and Exercise 12 of Section 4.5.

We mentioned in the introduction to Chapter 3 that one reason for using algebraic polynomials to approximate an arbitrary set of data is that, given any continuous function defined on a closed interval, there exists a polynomial that is arbitrarily close to the function at every point in the interval. Also, the derivatives and integrals of polynomials are easily obtained and evaluated. It should not be surprising, then, that many procedures for approximating derivatives and integrals use the polynomials that approximate the function.

4.1 Numerical Differentiation

The derivative of the function f at x_0 is

$$f'(x_0) = \lim_{h\to 0} \frac{f(x_0+h) - f(x_0)}{h}.$$

This formula gives an obvious way to generate an approximation to $f'(x_0)$; simply compute

$$\frac{f(x_0+h) - f(x_0)}{h}$$

for small values of h. Although this may be obvious, it is not very successful, due to our old nemesis round-off error. But it is certainly a place to start.

To approximate $f'(x_0)$, suppose first that $x_0 \in (a, b)$, where $f \in C^2[a, b]$, and that $x_1 = x_0 + h$ for some $h \neq 0$ that is sufficiently small to ensure that $x_1 \in [a, b]$. We construct the first Lagrange polynomial $P_{0,1}(x)$ for f determined by x_0 and x_1, with its error term:

$$\begin{aligned} f(x) &= P_{0,1}(x) + \frac{(x-x_0)(x-x_1)}{2!} f''(\xi(x)) \\ &= \frac{f(x_0)(x-x_0-h)}{-h} + \frac{f(x_0+h)(x-x_0)}{h} + \frac{(x-x_0)(x-x_0-h)}{2} f''(\xi(x)), \end{aligned}$$

for some $\xi(x)$ between x_0 and x_1. Differentiating gives

$$\begin{aligned} f'(x) &= \frac{f(x_0+h) - f(x_0)}{h} + D_x\left[\frac{(x-x_0)(x-x_0-h)}{2} f''(\xi(x))\right] \\ &= \frac{f(x_0+h) - f(x_0)}{h} + \frac{2(x-x_0)-h}{2} f''(\xi(x)) \\ &\quad + \frac{(x-x_0)(x-x_0-h)}{2} D_x(f''(\xi(x))). \end{aligned}$$

Deleting the terms involving $\xi(x)$ gives

$$f'(x) \approx \frac{f(x_0+h) - f(x_0)}{h}.$$

Difference equations were used and popularized by Isaac Newton in the last quarter of the 17th century, but many of these techniques had previously been developed by Thomas Harriot (1561–1621) and Henry Briggs (1561–1630). Harriot made significant advances in navigation techniques, and Briggs was the person most responsible for the acceptance of logarithms as an aid to computation.

One difficulty with this formula is that we have no information about $D_x f''(\xi(x))$, so the truncation error cannot be estimated. When x is x_0, however, the coefficient of $D_x f''(\xi(x))$ is 0, and the formula simplifies to

$$f'(x_0) = \frac{f(x_0+h) - f(x_0)}{h} - \frac{h}{2} f''(\xi). \tag{4.1}$$

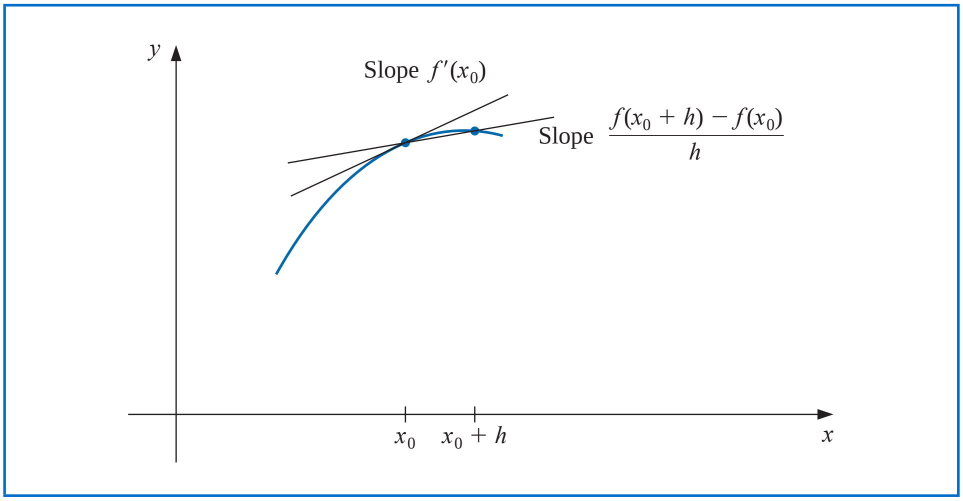
For small values of h, the difference quotient $[f(x_0+h) - f(x_0)]/h$ can be used to approximate $f'(x_0)$ with an error bounded by $M|h|/2$, where M is a bound on $|f''(x)|$ for x between x_0 and $x_0 + h$. This formula is known as the **forward-difference formula** if $h > 0$ (see Figure 4.1) and the **backward-difference formula** if $h < 0$.

Example 1 Use the forward-difference formula to approximate the derivative of $f(x) = \ln x$ at $x_0 = 1.8$ using $h = 0.1$, $h = 0.05$, and $h = 0.01$, and determine bounds for the approximation errors.

Solution The forward-difference formula

$$\frac{f(1.8+h) - f(1.8)}{h}$$

Figure 4.1



with $h = 0.1$ gives

$$\frac{\ln 1.9 - \ln 1.8}{0.1} = \frac{0.64185389 - 0.58778667}{0.1} = 0.5406722.$$

Because $f''(x) = -1/x^2$ and $1.8 < \xi < 1.9$, a bound for this approximation error is

$$\frac{|hf''(\xi)|}{2} = \frac{|h|}{2\xi^2} < \frac{0.1}{2(1.8)^2} = 0.0154321.$$

The approximation and error bounds when $h = 0.05$ and $h = 0.01$ are found in a similar manner and the results are shown in Table 4.1.

Table 4.1

h	$f(1.8+h)$	$\frac{f(1.8+h)-f(1.8)}{h}$	$\frac{\|h\|}{2(1.8)^2}$
0.1	0.64185389	0.5406722	0.0154321
0.05	0.61518564	0.5479795	0.0077160
0.01	0.59332685	0.5540180	0.0015432

Since $f'(x) = 1/x$, the exact value of $f'(1.8)$ is $0.55\overline{5}$, and in this case the error bounds are quite close to the true approximation error. ■

To obtain general derivative approximation formulas, suppose that $\{x_0, x_1, \ldots, x_n\}$ are $(n+1)$ distinct numbers in some interval I and that $f \in C^{n+1}(I)$. From Theorem 3.3 on page 112,

$$f(x) = \sum_{k=0}^{n} f(x_k)L_k(x) + \frac{(x-x_0)\cdots(x-x_n)}{(n+1)!} f^{(n+1)}(\xi(x)),$$

for some $\xi(x)$ in I, where $L_k(x)$ denotes the kth Lagrange coefficient polynomial for f at $x_0, x_1, \ldots, x_n$. Differentiating this expression gives

$$f'(x) = \sum_{k=0}^{n} f(x_k)L_k'(x) + D_x\left[\frac{(x-x_0)\cdots(x-x_n)}{(n+1!)}\right] f^{(n+1)}(\xi(x))$$
$$+ \frac{(x-x_0)\cdots(x-x_n)}{(n+1)!} D_x[f^{(n+1)}(\xi(x))].$$

We again have a problem estimating the truncation error unless x is one of the numbers x_j. In this case, the term multiplying $D_x[f^{(n+1)}(\xi(x))]$ is 0, and the formula becomes

$$f'(x_j) = \sum_{k=0}^{n} f(x_k)L_k'(x_j) + \frac{f^{(n+1)}(\xi(x_j))}{(n+1)!}\prod_{\substack{k=0\\k\neq j}}^{n}(x_j - x_k), \tag{4.2}$$

which is called an **$(n+1)$-point formula** to approximate $f'(x_j)$.

In general, using more evaluation points in Eq. (4.2) produces greater accuracy, although the number of functional evaluations and growth of round-off error discourages this somewhat. The most common formulas are those involving three and five evaluation points.

We first derive some useful three-point formulas and consider aspects of their errors. Because

$$L_0(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}, \quad \text{we have} \quad L_0'(x) = \frac{2x - x_1 - x_2}{(x_0-x_1)(x_0-x_2)}.$$

Similarly,

$$L_1'(x) = \frac{2x - x_0 - x_2}{(x_1-x_0)(x_1-x_2)} \quad \text{and} \quad L_2'(x) = \frac{2x - x_0 - x_1}{(x_2-x_0)(x_2-x_1)}.$$

Hence, from Eq. (4.2),

$$f'(x_j) = f(x_0)\left[\frac{2x_j - x_1 - x_2}{(x_0-x_1)(x_0-x_2)}\right] + f(x_1)\left[\frac{2x_j - x_0 - x_2}{(x_1-x_0)(x_1-x_2)}\right]$$
$$+ f(x_2)\left[\frac{2x_j - x_0 - x_1}{(x_2-x_0)(x_2-x_1)}\right] + \frac{1}{6}f^{(3)}(\xi_j)\prod_{\substack{k=0\\k\neq j}}^{2}(x_j - x_k), \tag{4.3}$$

for each $j = 0, 1, 2$, where the notation ξ_j indicates that this point depends on x_j.

Three-Point Formulas

The formulas from Eq. (4.3) become especially useful if the nodes are equally spaced, that is, when

$$x_1 = x_0 + h \quad \text{and} \quad x_2 = x_0 + 2h, \quad \text{for some } h \neq 0.$$

We will assume equally-spaced nodes throughout the remainder of this section.

Using Eq. (4.3) with $x_j = x_0, x_1 = x_0 + h$, and $x_2 = x_0 + 2h$ gives

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_1) - \frac{1}{2}f(x_2)\right] + \frac{h^2}{3}f^{(3)}(\xi_0).$$

Doing the same for $x_j = x_1$ gives

$$f'(x_1) = \frac{1}{h}\left[-\frac{1}{2}f(x_0) + \frac{1}{2}f(x_2)\right] - \frac{h^2}{6}f^{(3)}(\xi_1),$$

and for $x_j = x_2$,

$$f'(x_2) = \frac{1}{h}\left[\frac{1}{2}f(x_0) - 2f(x_1) + \frac{3}{2}f(x_2)\right] + \frac{h^2}{3}f^{(3)}(\xi_2).$$

Since $x_1 = x_0 + h$ and $x_2 = x_0 + 2h$, these formulas can also be expressed as

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_0+h) - \frac{1}{2}f(x_0+2h)\right] + \frac{h^2}{3}f^{(3)}(\xi_0),$$

$$f'(x_0+h) = \frac{1}{h}\left[-\frac{1}{2}f(x_0) + \frac{1}{2}f(x_0+2h)\right] - \frac{h^2}{6}f^{(3)}(\xi_1),$$

and

$$f'(x_0+2h) = \frac{1}{h}\left[\frac{1}{2}f(x_0) - 2f(x_0+h) + \frac{3}{2}f(x_0+2h)\right] + \frac{h^2}{3}f^{(3)}(\xi_2).$$

As a matter of convenience, the variable substitution x_0 for $x_0 + h$ is used in the middle equation to change this formula to an approximation for $f'(x_0)$. A similar change, x_0 for $x_0 + 2h$, is used in the last equation. This gives three formulas for approximating $f'(x_0)$:

$$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0+h) - f(x_0+2h)] + \frac{h^2}{3}f^{(3)}(\xi_0),$$

$$f'(x_0) = \frac{1}{2h}[-f(x_0-h) + f(x_0+h)] - \frac{h^2}{6}f^{(3)}(\xi_1),$$

and

$$f'(x_0) = \frac{1}{2h}[f(x_0-2h) - 4f(x_0-h) + 3f(x_0)] + \frac{h^2}{3}f^{(3)}(\xi_2).$$

Finally, note that the last of these equations can be obtained from the first by simply replacing h with $-h$, so there are actually only two formulas:

Three-Point Endpoint Formula

- $$f'(x_0) = \frac{1}{2h}[-3f(x_0) + 4f(x_0+h) - f(x_0+2h)] + \frac{h^2}{3}f^{(3)}(\xi_0), \tag{4.4}$$

where ξ_0 lies between x_0 and $x_0 + 2h$.

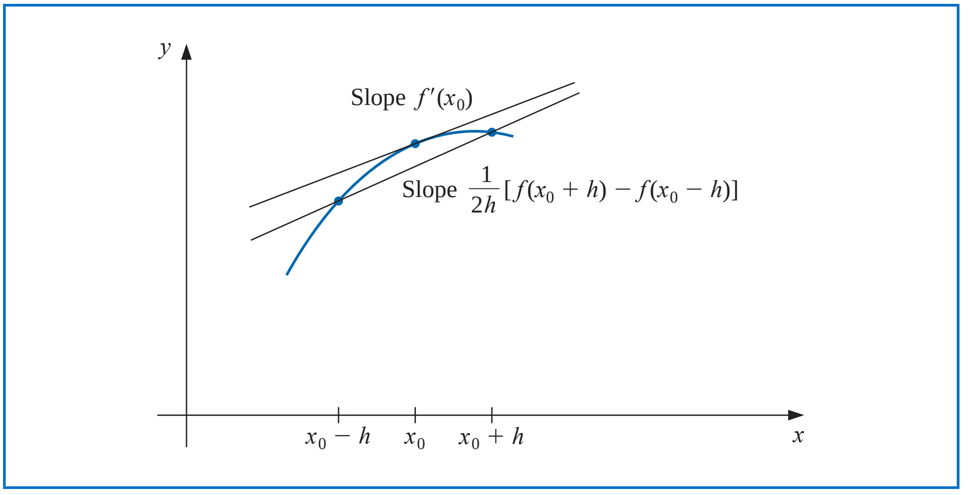
Three-Point Midpoint Formula

- $$f'(x_0) = \frac{1}{2h}[f(x_0+h) - f(x_0-h)] - \frac{h^2}{6}f^{(3)}(\xi_1), \tag{4.5}$$

where ξ_1 lies between $x_0 - h$ and $x_0 + h$.

Although the errors in both Eq. (4.4) and Eq. (4.5) are $O(h^2)$, the error in Eq. (4.5) is approximately half the error in Eq. (4.4). This is because Eq. (4.5) uses data on both sides of x_0 and Eq. (4.4) uses data on only one side. Note also that f needs to be evaluated at only two points in Eq. (4.5), whereas in Eq. (4.4) three evaluations are needed. Figure 4.2 on page 178 gives an illustration of the approximation produced from Eq. (4.5). The approximation in Eq. (4.4) is useful near the ends of an interval, because information about f outside the interval may not be available.

Figure 4.2



Five-Point Formulas

The methods presented in Eqs. (4.4) and (4.5) are called **three-point formulas** (even though the third point $f(x_0)$ does not appear in Eq. (4.5)). Similarly, there are **five-point formulas** that involve evaluating the function at two additional points. The error term for these formulas is $O(h^4)$. One common five-point formula is used to determine approximations for the derivative at the midpoint.

Five-Point Midpoint Formula

- $$f'(x_0) = \frac{1}{12h}[f(x_0-2h) - 8f(x_0-h) + 8f(x_0+h) - f(x_0+2h)] + \frac{h^4}{30}f^{(5)}(\xi), \tag{4.6}$$

  where ξ lies between $x_0 - 2h$ and $x_0 + 2h$.

The derivation of this formula is considered in Section 4.2. The other five-point formula is used for approximations at the endpoints.

Five-Point Endpoint Formula

- $$\begin{aligned} f'(x_0) = \frac{1}{12h}[&-25f(x_0) + 48f(x_0+h) - 36f(x_0+2h) \\ &+ 16f(x_0+3h) - 3f(x_0+4h)] + \frac{h^4}{5}f^{(5)}(\xi), \end{aligned} \tag{4.7}$$

  where ξ lies between x_0 and $x_0 + 4h$.

Left-endpoint approximations are found using this formula with $h > 0$ and right-endpoint approximations with $h < 0$. The five-point endpoint formula is particularly useful for the clamped cubic spline interpolation of Section 3.5.

Example 2 Values for $f(x) = xe^x$ are given in Table 4.2. Use all the applicable three-point and five-point formulas to approximate $f'(2.0)$.

Table 4.2

x	$f(x)$
1.8	10.889365
1.9	12.703199
2.0	14.778112
2.1	17.148957
2.2	19.855030

Solution The data in the table permit us to find four different three-point approximations. We can use the endpoint formula (4.4) with $h = 0.1$ or with $h = -0.1$, and we can use the midpoint formula (4.5) with $h = 0.1$ or with $h = 0.2$.

Using the endpoint formula (4.4) with $h = 0.1$ gives

$$\frac{1}{0.2}[-3f(2.0) + 4f(2.1) - f(2.2)] = 5[-3(14.778112) + 4(17.148957) - 19.855030)] = 22.032310,$$

and with $h = -0.1$ gives 22.054525.

Using the midpoint formula (4.5) with $h = 0.1$ gives

$$\frac{1}{0.2}[f(2.1) - f(1.9)] = 5(17.148957 - 12.7703199) = 22.228790,$$

and with $h = 0.2$ gives 22.414163.

The only five-point formula for which the table gives sufficient data is the midpoint formula (4.6) with $h = 0.1$. This gives

$$\begin{aligned}\frac{1}{1.2}[f(1.8) - 8f(1.9) + 8f(2.1) - f(2.2)] &= \frac{1}{1.2}[10.889365 - 8(12.703199) \\ &\quad + 8(17.148957) - 19.855030] \\ &= 22.166999\end{aligned}$$

If we had no other information we would accept the five-point midpoint approximation using $h = 0.1$ as the most accurate, and expect the true value to be between that approximation and the three-point mid-point approximation that is in the interval [22.166, 22.229].

The true value in this case is $f'(2.0) = (2 + 1)e^2 = 22.167168$, so the approximation errors are actually:

Three-point endpoint with $h = 0.1$: 1.35×10^{-1};

Three-point endpoint with $h = -0.1$: 1.13×10^{-1};

Three-point midpoint with $h = 0.1$: -6.16×10^{-2};

Three-point midpoint with $h = 0.2$: -2.47×10^{-1};

Five-point midpoint with $h = 0.1$: 1.69×10^{-4}. ■

Methods can also be derived to find approximations to higher derivatives of a function using only tabulated values of the function at various points. The derivation is algebraically tedious, however, so only a representative procedure will be presented.

Expand a function f in a third Taylor polynomial about a point x_0 and evaluate at $x_0 + h$ and $x_0 - h$. Then

$$f(x_0 + h) = f(x_0) + f'(x_0)h + \frac{1}{2}f''(x_0)h^2 + \frac{1}{6}f'''(x_0)h^3 + \frac{1}{24}f^{(4)}(\xi_1)h^4$$

and

$$f(x_0 - h) = f(x_0) - f'(x_0)h + \frac{1}{2}f''(x_0)h^2 - \frac{1}{6}f'''(x_0)h^3 + \frac{1}{24}f^{(4)}(\xi_{-1})h^4,$$

where $x_0 - h < \xi_{-1} < x_0 < \xi_1 < x_0 + h$.

If we add these equations, the terms involving $f'(x_0)$ and $-f'(x_0)$ cancel, so

$$f(x_0 + h) + f(x_0 - h) = 2f(x_0) + f''(x_0)h^2 + \frac{1}{24}[f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})]h^4.$$

Solving this equation for $f''(x_0)$ gives

$$f''(x_0) = \frac{1}{h^2}[f(x_0 - h) - 2f(x_0) + f(x_0 + h)] - \frac{h^2}{24}[f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})]. \quad (4.8)$$

Suppose $f^{(4)}$ is continuous on $[x_0 - h, x_0 + h]$. Since $\frac{1}{2}[f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})]$ is between $f^{(4)}(\xi_1)$ and $f^{(4)}(\xi_{-1})$, the Intermediate Value Theorem implies that a number ξ exists between ξ_1 and ξ_{-1}, and hence in $(x_0 - h, x_0 + h)$, with

$$f^{(4)}(\xi) = \frac{1}{2}\left[f^{(4)}(\xi_1) + f^{(4)}(\xi_{-1})\right].$$

This permits us to rewrite Eq. (4.8) in its final form.

Second Derivative Midpoint Formula

- $$f''(x_0) = \frac{1}{h^2}[f(x_0 - h) - 2f(x_0) + f(x_0 + h)] - \frac{h^2}{12}f^{(4)}(\xi), \quad (4.9)$$

  for some ξ, where $x_0 - h < \xi < x_0 + h$.

If $f^{(4)}$ is continuous on $[x_0 - h, x_0 + h]$ it is also bounded, and the approximation is $O(h^2)$.

Example 3 In Example 2 we used the data shown in Table 4.3 to approximate the first derivative of $f(x) = xe^x$ at $x = 2.0$. Use the second derivative formula (4.9) to approximate $f''(2.0)$.

Table 4.3

x	$f(x)$
1.8	10.889365
1.9	12.703199
2.0	14.778112
2.1	17.148957
2.2	19.855030

Solution The data permits us to determine two approximations for $f''(2.0)$. Using (4.9) with $h = 0.1$ gives

$$\frac{1}{0.01}[f(1.9) - 2f(2.0) + f(2.1)] = 100[12.703199 - 2(14.778112) + 17.148957]$$
$$= 29.593200,$$

and using (4.9) with $h = 0.2$ gives

$$\frac{1}{0.04}[f(1.8) - 2f(2.0) + f(2.2)] = 25[10.889365 - 2(14.778112) + 19.855030]$$
$$= 29.704275.$$

Because $f''(x) = (x + 2)e^x$, the exact value is $f''(2.0) = 29.556224$. Hence the actual errors are -3.70×10^{-2} and -1.48×10^{-1}, respectively. ■

Round-Off Error Instability

It is particularly important to pay attention to round-off error when approximating derivatives. To illustrate the situation, let us examine the three-point midpoint formula Eq. (4.5),

$$f'(x_0) = \frac{1}{2h}[f(x_0 + h) - f(x_0 - h)] - \frac{h^2}{6}f^{(3)}(\xi_1),$$

more closely. Suppose that in evaluating $f(x_0 + h)$ and $f(x_0 - h)$ we encounter round-off errors $e(x_0 + h)$ and $e(x_0 - h)$. Then our computations actually use the values $\tilde{f}(x_0 + h)$ and $\tilde{f}(x_0 - h)$, which are related to the true values $f(x_0 + h)$ and $f(x_0 - h)$ by

$$f(x_0 + h) = \tilde{f}(x_0 + h) + e(x_0 + h) \quad \text{and} \quad f(x_0 - h) = \tilde{f}(x_0 - h) + e(x_0 - h).$$

The total error in the approximation,

$$f'(x_0) - \frac{\tilde{f}(x_0 + h) - \tilde{f}(x_0 - h)}{2h} = \frac{e(x_0 + h) - e(x_0 - h)}{2h} - \frac{h^2}{6} f^{(3)}(\xi_1),$$

is due both to round-off error, the first part, and to truncation error. If we assume that the round-off errors $e(x_0 \pm h)$ are bounded by some number $\varepsilon > 0$ and that the third derivative of f is bounded by a number $M > 0$, then

$$\left| f'(x_0) - \frac{\tilde{f}(x_0 + h) - \tilde{f}(x_0 - h)}{2h} \right| \le \frac{\varepsilon}{h} + \frac{h^2}{6} M.$$

To reduce the truncation error, $h^2 M/6$, we need to reduce h. But as h is reduced, the round-off error ε/h grows. In practice, then, it is seldom advantageous to let h be too small, because in that case the round-off error will dominate the calculations.

Illustration Consider using the values in Table 4.4 to approximate $f'(0.900)$, where $f(x) = \sin x$. The true value is $\cos 0.900 = 0.62161$. The formula

$$f'(0.900) \approx \frac{f(0.900 + h) - f(0.900 - h)}{2h},$$

with different values of h, gives the approximations in Table 4.5.

Table 4.4

x	$\sin x$	x	$\sin x$
0.800	0.71736	0.901	0.78395
0.850	0.75128	0.902	0.78457
0.880	0.77074	0.905	0.78643
0.890	0.77707	0.910	0.78950
0.895	0.78021	0.920	0.79560
0.898	0.78208	0.950	0.81342
0.899	0.78270	1.000	0.84147

Table 4.5

h	Approximation to $f'(0.900)$	Error
0.001	0.62500	0.00339
0.002	0.62250	0.00089
0.005	0.62200	0.00039
0.010	0.62150	−0.00011
0.020	0.62150	−0.00011
0.050	0.62140	−0.00021
0.100	0.62055	−0.00106

The optimal choice for h appears to lie between 0.005 and 0.05. We can use calculus to verify (see Exercise 29) that a minimum for

$$e(h) = \frac{\varepsilon}{h} + \frac{h^2}{6} M,$$

occurs at $h = \sqrt[3]{3\varepsilon/M}$, where

$$M = \max_{x \in [0.800, 1.00]} |f'''(x)| = \max_{x \in [0.800, 1.00]} |\cos x| = \cos 0.8 \approx 0.69671.$$

Because values of f are given to five decimal places, we will assume that the round-off error is bounded by $\varepsilon = 5 \times 10^{-6}$. Therefore, the optimal choice of h is approximately

$$h = \sqrt[3]{\frac{3(0.000005)}{0.69671}} \approx 0.028,$$

which is consistent with the results in Table 4.6. □

In practice, we cannot compute an optimal h to use in approximating the derivative, since we have no knowledge of the third derivative of the function. But we must remain aware that reducing the step size will not always improve the approximation. □

We have considered only the round-off error problems that are presented by the three-point formula Eq. (4.5), but similar difficulties occur with all the differentiation formulas. The reason can be traced to the need to divide by a power of h. As we found in Section 1.2 (see, in particular, Example 3), division by small numbers tends to exaggerate round-off error, and this operation should be avoided if possible. In the case of numerical differentiation, we cannot avoid the problem entirely, although the higher-order methods reduce the difficulty.

Keep in mind that difference method approximations might be unstable.

As approximation methods, numerical differentiation is *unstable*, since the small values of h needed to reduce truncation error also cause the round-off error to grow. This is the first class of unstable methods we have encountered, and these techniques would be avoided if it were possible. However, in addition to being used for computational purposes, the formulas are needed for approximating the solutions of ordinary and partial-differential equations.

EXERCISE SET 4.1

1. Use the forward-difference formulas and backward-difference formulas to determine each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
0.5	0.4794	
0.6	0.5646	
0.7	0.6442	

b.

x	$f(x)$	$f'(x)$
0.0	0.00000	
0.2	0.74140	
0.4	1.3718	

2. Use the forward-difference formulas and backward-difference formulas to determine each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
−0.3	1.9507	
−0.2	2.0421	
−0.1	2.0601	

b.

x	$f(x)$	$f'(x)$
1.0	1.0000	
1.2	1.2625	
1.4	1.6595	

3. The data in Exercise 1 were taken from the following functions. Compute the actual errors in Exercise 1, and find error bounds using the error formulas.

a. $f(x) = \sin x$ **b.** $f(x) = e^x - 2x^2 + 3x - 1$

4. The data in Exercise 2 were taken from the following functions. Compute the actual errors in Exercise 2, and find error bounds using the error formulas.

a. $f(x) = 2\cos 2x - x$ **b.** $f(x) = x^2 \ln x + 1$

5. Use the most accurate three-point formula to determine each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
1.1	9.025013	
1.2	11.02318	
1.3	13.46374	
1.4	16.44465	

b.

x	$f(x)$	$f'(x)$
8.1	16.94410	
8.3	17.56492	
8.5	18.19056	
8.7	18.82091	

c.

x	$f(x)$	$f'(x)$
2.9	−4.827866	
3.0	−4.240058	
3.1	−3.496909	
3.2	−2.596792	

d.

x	$f(x)$	$f'(x)$
2.0	3.6887983	
2.1	3.6905701	
2.2	3.6688192	
2.3	3.6245909	

6. Use the most accurate three-point formula to determine each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
-0.3	-0.27652	
-0.2	-0.25074	
-0.1	-0.16134	
0	0	

b.

x	$f(x)$	$f'(x)$
7.4	-68.3193	
7.6	-71.6982	
7.8	-75.1576	
8.0	-78.6974	

c.

x	$f(x)$	$f'(x)$
1.1	1.52918	
1.2	1.64024	
1.3	1.70470	
1.4	1.71277	

d.

x	$f(x)$	$f'(x)$
-2.7	0.054797	
-2.5	0.11342	
-2.3	0.65536	
-2.1	0.98472	

7. The data in Exercise 5 were taken from the following functions. Compute the actual errors in Exercise 5, and find error bounds using the error formulas.

a. $f(x) = e^{2x}$
b. $f(x) = x \ln x$
c. $f(x) = x \cos x - x^2 \sin x$
d. $f(x) = 2(\ln x)^2 + 3 \sin x$

8. The data in Exercise 6 were taken from the following functions. Compute the actual errors in Exercise 6, and find error bounds using the error formulas.

a. $f(x) = e^{2x} - \cos 2x$
b. $f(x) = \ln(x + 2) - (x + 1)^2$
c. $f(x) = x \sin x + x^2 \cos x$
d. $f(x) = (\cos 3x)^2 - e^{2x}$

9. Use the formulas given in this section to determine, as accurately as possible, approximations for each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
2.1	-1.709847	
2.2	-1.373823	
2.3	-1.119214	
2.4	-0.9160143	
2.5	-0.7470223	
2.6	-0.6015966	

b.

x	$f(x)$	$f'(x)$
-3.0	9.367879	
-2.8	8.233241	
-2.6	7.180350	
-2.4	6.209329	
-2.2	5.320305	
-2.0	4.513417	

10. Use the formulas given in this section to determine, as accurately as possible, approximations for each missing entry in the following tables.

a.

x	$f(x)$	$f'(x)$
1.05	-1.709847	
1.10	-1.373823	
1.15	-1.119214	
1.20	-0.9160143	
1.25	-0.7470223	
1.30	-0.6015966	

b.

x	$f(x)$	$f'(x)$
-3.0	16.08554	
-2.8	12.64465	
-2.6	9.863738	
-2.4	7.623176	
-2.2	5.825013	
-2.0	4.389056	

11. The data in Exercise 9 were taken from the following functions. Compute the actual errors in Exercise 9, and find error bounds using the error formulas and Maple.

a. $f(x) = \tan x$
b. $f(x) = e^{x/3} + x^2$

12. The data in Exercise 10 were taken from the following functions. Compute the actual errors in Exercise 10, and find error bounds using the error formulas and Maple.

a. $f(x) = \tan 2x$
b. $f(x) = e^{-x} - 1 + x$

13. Use the following data and the knowledge that the first five derivatives of f are bounded on $[1, 5]$ by 2, 3, 6, 12 and 23, respectively, to approximate $f'(3)$ as accurately as possible. Find a bound for the error.

x	1	2	3	4	5
$f(x)$	2.4142	2.6734	2.8974	3.0976	3.2804

14. Repeat Exercise 13, assuming instead that the third derivative of f is bounded on $[1, 5]$ by 4.

15. Repeat Exercise 1 using four-digit rounding arithmetic, and compare the errors to those in Exercise 3.

16. Repeat Exercise 5 using four-digit chopping arithmetic, and compare the errors to those in Exercise 7.

17. Repeat Exercise 9 using four-digit rounding arithmetic, and compare the errors to those in Exercise 11.

18. Consider the following table of data:

x	0.2	0.4	0.6	0.8	1.0
$f(x)$	0.9798652	0.9177710	0.808038	0.6386093	0.3843735

a. Use all the appropriate formulas given in this section to approximate $f'(0.4)$ and $f''(0.4)$.

b. Use all the appropriate formulas given in this section to approximate $f'(0.6)$ and $f''(0.6)$.

19. Let $f(x) = \cos \pi x$. Use Eq. (4.9) and the values of $f(x)$ at $x = 0.25$, 0.5, and 0.75 to approximate $f''(0.5)$. Compare this result to the exact value and to the approximation found in Exercise 15 of Section 3.5. Explain why this method is particularly accurate for this problem, and find a bound for the error.

20. Let $f(x) = 3xe^x - \cos x$. Use the following data and Eq. (4.9) to approximate $f''(1.3)$ with $h = 0.1$ and with $h = 0.01$.

x	1.20	1.29	1.30	1.31	1.40
$f(x)$	11.59006	13.78176	14.04276	14.30741	16.86187

Compare your results to $f''(1.3)$.

21. Consider the following table of data:

x	0.2	0.4	0.6	0.8	1.0
$f(x)$	0.9798652	0.9177710	0.8080348	0.6386093	0.3843735

a. Use Eq. (4.7) to approximate $f'(0.2)$.

b. Use Eq. (4.7) to approximate $f'(1.0)$.

c. Use Eq. (4.6) to approximate $f'(0.6)$.

22. Derive an $O(h^4)$ five-point formula to approximate $f'(x_0)$ that uses $f(x_0 - h)$, $f(x_0)$, $f(x_0 + h)$, $f(x_0 + 2h)$, and $f(x_0 + 3h)$. [*Hint:* Consider the expression $Af(x_0 - h) + Bf(x_0 + h) + Cf(x_0 + 2h) + Df(x_0 + 3h)$. Expand in fourth Taylor polynomials, and choose A, B, C, and D appropriately.]

23. Use the formula derived in Exercise 22 and the data of Exercise 21 to approximate $f'(0.4)$ and $f'(0.8)$.

24. **a.** Analyze the round-off errors, as in Example 4, for the formula

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2} f''(\xi_0).$$

b. Find an optimal $h > 0$ for the function given in Example 2.

25. In Exercise 10 of Section 3.4 data were given describing a car traveling on a straight road. That problem asked to predict the position and speed of the car when $t = 10$ s. Use the following times and positions to predict the speed at each time listed.

Time	0	3	5	8	10	13
Distance	0	225	383	623	742	993

26. In a circuit with impressed voltage $\mathcal{E}(t)$ and inductance L, Kirchhoff's first law gives the relationship

$$\mathcal{E}(t) = L\frac{di}{dt} + Ri,$$

where R is the resistance in the circuit and i is the current. Suppose we measure the current for several values of t and obtain:

t	1.00	1.01	1.02	1.03	1.0
i	3.10	3.12	3.14	3.18	3.24

where t is measured in seconds, i is in amperes, the inductance L is a constant 0.98 henries, and the resistance is 0.142 ohms. Approximate the voltage $\mathcal{E}(t)$ when $t = 1.00$, 1.01, 1.02, 1.03, and 1.04.

27. All calculus students know that the derivative of a function f at x can be defined as

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}.$$

Choose your favorite function f, nonzero number x, and computer or calculator. Generate approximations $f_n'(x)$ to $f'(x)$ by

$$f_n'(x) = \frac{f(x+10^{-n}) - f(x)}{10^{-n}},$$

for $n = 1, 2, \ldots, 20$, and describe what happens.

28. Derive a method for approximating $f'''(x_0)$ whose error term is of order h^2 by expanding the function f in a fourth Taylor polynomial about x_0 and evaluating at $x_0 \pm h$ and $x_0 \pm 2h$.

29. Consider the function

$$e(h) = \frac{\varepsilon}{h} + \frac{h^2}{6}M,$$

where M is a bound for the third derivative of a function. Show that $e(h)$ has a minimum at $\sqrt[3]{3\varepsilon/M}$.

4.2 Richardson's Extrapolation

Richardson's extrapolation is used to generate high-accuracy results while using low-order formulas. Although the name attached to the method refers to a paper written by L. F. Richardson and J. A. Gaunt [RG] in 1927, the idea behind the technique is much older. An interesting article regarding the history and application of extrapolation can be found in [Joy].

Lewis Fry Richardson (1881–1953) was the first person to systematically apply mathematics to weather prediction while working in England for the Meteorological Office. As a conscientious objector during World War I, he wrote extensively about the economic futility of warfare, using systems of differential equations to model rational interactions between countries. The extrapolation technique that bears his name was the rediscovery of a technique with roots that are at least as old as Christiaan Hugyens (1629–1695), and possibly Archimedes (287–212 B.C.E.).

Extrapolation can be applied whenever it is known that an approximation technique has an error term with a predictable form, one that depends on a parameter, usually the step size h. Suppose that for each number $h \neq 0$ we have a formula $N_1(h)$ that approximates an unknown constant M, and that the truncation error involved with the approximation has the form

$$M - N_1(h) = K_1 h + K_2 h^2 + K_3 h^3 + \cdots,$$

for some collection of (unknown) constants $K_1, K_2, K_3, \ldots$.

The truncation error is $O(h)$, so unless there was a large variation in magnitude among the constants $K_1, K_2, K_3, \ldots$,

$$M - N_1(0.1) \approx 0.1K_1, \quad M - N_1(0.01) \approx 0.01K_1,$$

and, in general, $M - N_1(h) \approx K_1 h$.

The object of extrapolation is to find an easy way to combine these rather inaccurate $O(h)$ approximations in an appropriate way to produce formulas with a higher-order truncation error.

Suppose, for example, we can combine the $N_1(h)$ formulas to produce an $O(h^2)$ approximation formula, $N_2(h)$, for M with

$$M - N_2(h) = \hat{K}_2 h^2 + \hat{K}_3 h^3 + \cdots,$$

for some, again unknown, collection of constants $\hat{K}_2, \hat{K}_3, \ldots$. Then we would have

$$M - N_2(0.1) \approx 0.01\hat{K}_2, \quad M - N_2(0.01) \approx 0.0001\hat{K}_2,$$

and so on. If the constants K_1 and $\hat{K}_2$ are roughly of the same magnitude, then the $N_2(h)$ approximations would be much better than the corresponding $N_1(h)$ approximations. The extrapolation continues by combining the $N_2(h)$ approximations in a manner that produces formulas with $O(h^3)$ truncation error, and so on.

To see specifically how we can generate the extrapolation formulas, consider the $O(h)$ formula for approximating M

$$M = N_1(h) + K_1 h + K_2 h^2 + K_3 h^3 + \cdots. \tag{4.10}$$

The formula is assumed to hold for all positive h, so we replace the parameter h by half its value. Then we have a second $O(h)$ approximation formula

$$M = N_1\left(\frac{h}{2}\right) + K_1\frac{h}{2} + K_2\frac{h^2}{4} + K_3\frac{h^3}{8} + \cdots. \tag{4.11}$$

Subtracting Eq. (4.10) from twice Eq. (4.11) eliminates the term involving K_1 and gives

$$M = N_1\left(\frac{h}{2}\right) + \left[N_1\left(\frac{h}{2}\right) - N_1(h)\right] + K_2\left(\frac{h^2}{2} - h^2\right) + K_3\left(\frac{h^3}{4} - h^3\right) + \cdots. \tag{4.12}$$

Define

$$N_2(h) = N_1\left(\frac{h}{2}\right) + \left[N_1\left(\frac{h}{2}\right) - N_1(h)\right].$$

Then Eq. (4.12) is an $O(h^2)$ approximation formula for M:

$$M = N_2(h) - \frac{K_2}{2}h^2 - \frac{3K_3}{4}h^3 - \cdots. \tag{4.13}$$

Example 1 In Example 1 of Section 4.1 we use the forward-difference method with $h = 0.1$ and $h = 0.05$ to find approximations to $f'(1.8)$ for $f(x) = \ln(x)$. Assume that this formula has truncation error $O(h)$ and use extrapolation on these values to see if this results in a better approximation.

Solution In Example 1 of Section 4.1 we found that

$$\text{with } h = 0.1\text{: } f'(1.8) \approx 0.5406722, \quad \text{and} \quad \text{with } h = 0.05\text{: } f'(1.8) \approx 0.5479795.$$

This implies that

$$N_1(0.1) = 0.5406722 \quad \text{and} \quad N_1(0.05) = 0.5479795.$$

Extrapolating these results gives the new approximation

$$N_2(0.1) = N_1(0.05) + (N_1(0.05) - N_1(0.1)) = 0.5479795 + (0.5479795 - 0.5406722)$$
$$= 0.555287.$$

The $h = 0.1$ and $h = 0.05$ results were found to be accurate to within 1.5×10^{-2} and 7.7×10^{-3}, respectively. Because $f'(1.8) = 1/1.8 = 0.\overline{5}$, the extrapolated value is accurate to within 2.7×10^{-4}. ■

Extrapolation can be applied whenever the truncation error for a formula has the form

$$\sum_{j=1}^{m-1} K_j h^{\alpha_j} + O(h^{\alpha_m}),$$

for a collection of constants K_j and when $\alpha_1 < \alpha_2 < \alpha_3 < \cdots < \alpha_m$. Many formulas used for extrapolation have truncation errors that contain only even powers of h, that is, have the form

$$M = N_1(h) + K_1 h^2 + K_2 h^4 + K_3 h^6 + \cdots . \tag{4.14}$$

The extrapolation is much more effective than when all powers of h are present because the averaging process produces results with errors $O(h^2)$, $O(h^4)$, $O(h^6)$, $\ldots$, with essentially no increase in computation, over the results with errors, $O(h)$, $O(h^2)$, $O(h^3)$, $\ldots$.

Assume that approximation has the form of Eq. (4.14). Replacing h with $h/2$ gives the $O(h^2)$ approximation formula

$$M = N_1\left(\frac{h}{2}\right) + K_1\frac{h^2}{4} + K_2\frac{h^4}{16} + K_3\frac{h^6}{64} + \cdots .$$

Subtracting Eq. (4.14) from 4 times this equation eliminates the h^2 term,

$$3M = \left[4N_1\left(\frac{h}{2}\right) - N_1(h)\right] + K_2\left(\frac{h^4}{4} - h^4\right) + K_3\left(\frac{h^6}{16} - h^6\right) + \cdots .$$

Dividing this equation by 3 produces an $O(h^4)$ formula

$$M = \frac{1}{3}\left[4N_1\left(\frac{h}{2}\right) - N_1(h)\right] + \frac{K_2}{3}\left(\frac{h^4}{4} - h^4\right) + \frac{K_3}{3}\left(\frac{h^6}{16} - h^6\right) + \cdots .$$

Defining

$$N_2(h) = \frac{1}{3}\left[4N_1\left(\frac{h}{2}\right) - N_1(h)\right] = N_1\left(\frac{h}{2}\right) + \frac{1}{3}\left[N_1\left(\frac{h}{2}\right) - N_1(h)\right],$$

produces the approximation formula with truncation error $O(h^4)$:

$$M = N_2(h) - K_2\frac{h^4}{4} - K_3\frac{5h^6}{16} + \cdots . \tag{4.15}$$

Now replace h in Eq. (4.15) with $h/2$ to produce a second $O(h^4)$ formula

$$M = N_2\left(\frac{h}{2}\right) - K_2\frac{h^4}{64} - K_3\frac{5h^6}{1024} - \cdots .$$

Subtracting Eq. (4.15) from 16 times this equation eliminates the h^4 term and gives

$$15M = \left[16N_2\left(\frac{h}{2}\right) - N_2(h)\right] + K_3\frac{15h^6}{64} + \cdots .$$

Dividing this equation by 15 produces the new $O(h^6)$ formula

$$M = \frac{1}{15}\left[16N_2\left(\frac{h}{2}\right) - N_2(h)\right] + K_3\frac{h^6}{64} + \cdots .$$

We now have the $O(h^6)$ approximation formula

$$N_3(h) = \frac{1}{15}\left[16N_2\left(\frac{h}{2}\right) - N_2(h)\right] = N_2\left(\frac{h}{2}\right) + \frac{1}{15}\left[N_2\left(\frac{h}{2}\right) - N_2(h)\right].$$

Continuing this procedure gives, for each $j = 2, 3, \ldots,$ the $O(h^{2j})$ approximation

$$N_j(h) = N_{j-1}\left(\frac{h}{2}\right) + \frac{N_{j-1}(h/2) - N_{j-1}(h)}{4^{j-1} - 1}.$$

Table 4.6 shows the order in which the approximations are generated when

$$M = N_1(h) + K_1 h^2 + K_2 h^4 + K_3 h^6 + \cdots. \tag{4.16}$$

It is conservatively assumed that the true result is accurate at least to within the agreement of the bottom two results in the diagonal, in this case, to within $|N_3(h) - N_4(h)|$.

Table 4.6

$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$
1: $N_1(h)$			
2: $N_1(\frac{h}{2})$	**3:** $N_2(h)$		
4: $N_1(\frac{h}{4})$	**5:** $N_2(\frac{h}{2})$	**6:** $N_3(h)$	
7: $N_1(\frac{h}{8})$	**8:** $N_2(\frac{h}{4})$	**9:** $N_3(\frac{h}{2})$	**10:** $N_4(h)$

Example 2 Taylor's theorem can be used to show that centered-difference formula in Eq. (4.5) to approximate $f'(x_0)$ can be expressed with an error formula:

$$f'(x_0) = \frac{1}{2h}[f(x_0 + h) - f(x_0 - h)] - \frac{h^2}{6}f'''(x_0) - \frac{h^4}{120}f^{(5)}(x_0) - \cdots.$$

Find approximations of order $O(h^2)$, $O(h^4)$, and $O(h^6)$ for $f'(2.0)$ when $f(x) = xe^x$ and $h = 0.2$.

Solution The constants $K_1 = -f'''(x_0)/6$, $K_2 = -f^{(5)}(x_0)/120, \cdots$, are not likely to be known, but this is not important. We only need to know that these constants exist in order to apply extrapolation.

We have the $O(h^2)$ approximation

$$f'(x_0) = N_1(h) - \frac{h^2}{6}f'''(x_0) - \frac{h^4}{120}f^{(5)}(x_0) - \cdots, \tag{4.17}$$

where

$$N_1(h) = \frac{1}{2h}[f(x_0 + h) - f(x_0 - h)].$$

This gives us the first $O(h^2)$ approximations

$$N_1(0.2) = \frac{1}{0.4}[f(2.2) - f(1.8)] = 2.5(19.855030 - 10.889365) = 22.414160,$$

and

$$N_1(0.1) = \frac{1}{0.2}[f(2.1) - f(1.9)] = 5(17.148957 - 12.703199) = 22.228786.$$

Combining these to produce the first $O(h^4)$ approximation gives

$$N_2(0.2) = N_1(0.1) + \frac{1}{3}(N_1(0.1) - N_1(0.2))$$
$$= 22.228786 + \frac{1}{3}(22.228786 - 22.414160) = 22.166995.$$

To determine an $O(h^6)$ formula we need another $O(h^4)$ result, which requires us to find the third $O(h^2)$ approximation

$$N_1(0.05) = \frac{1}{0.1}[f(2.05) - f(1.95)] = 10(15.924197 - 13.705941) = 22.182564.$$

We can now find the $O(h^4)$ approximation

$$\begin{aligned} N_2(0.1) &= N_1(0.05) + \frac{1}{3}(N_1(0.05) - N_1(0.1)) \\ &= 22.182564 + \frac{1}{3}(22.182564 - 22.228786) = 22.167157. \end{aligned}$$

and finally the $O(h^6)$ approximation

$$\begin{aligned} N_3(0.2) &= N_2(0.1) + \frac{1}{15}(N_2(0.1) - N_1(0.2)) \\ &= 22.167157 + \frac{1}{15}(22.167157 - 22.166995) = 22.167168. \end{aligned}$$

We would expect the final approximation to be accurate to at least the value 22.167 because the $N_2(0.2)$ and $N_3(0.2)$ give this same value. In fact, $N_3(0.2)$ is accurate to all the listed digits. ■

Each column beyond the first in the extrapolation table is obtained by a simple averaging process, so the technique can produce high-order approximations with minimal computational cost. However, as k increases, the round-off error in $N_1(h/2^k)$ will generally increase because the instability of numerical differentiation is related to the step size $h/2^k$. Also, the higher-order formulas depend increasingly on the entry to their immediate left in the table, which is the reason we recommend comparing the final diagonal entries to ensure accuracy.

In Section 4.1, we discussed both three- and five-point methods for approximating $f'(x_0)$ given various functional values of f. The three-point methods were derived by differentiating a Lagrange interpolating polynomial for f. The five-point methods can be obtained in a similar manner, but the derivation is tedious. Extrapolation can be used to more easily derive these formulas, as illustrated below.

Illustration Suppose we expand the function f in a fourth Taylor polynomial about x_0. Then

$$\begin{aligned} f(x) =& f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 + \frac{1}{6}f'''(x_0)(x - x_0)^3 \\ &+ \frac{1}{24}f^{(4)}(x_0)(x - x_0)^4 + \frac{1}{120}f^{(5)}(\xi)(x - x_0)^5, \end{aligned}$$

for some number ξ between x and x_0. Evaluating f at $x_0 + h$ and $x_0 - h$ gives

$$\begin{aligned} f(x_0 + h) =& f(x_0) + f'(x_0)h + \frac{1}{2}f''(x_0)h^2 + \frac{1}{6}f'''(x_0)h^3 \\ &+ \frac{1}{24}f^{(4)}(x_0)h^4 + \frac{1}{120}f^{(5)}(\xi_1)h^5 \end{aligned} \tag{4.18}$$

and

$$\begin{aligned} f(x_0 - h) =& f(x_0) - f'(x_0)h + \frac{1}{2}f''(x_0)h^2 - \frac{1}{6}f'''(x_0)h^3 \\ &+ \frac{1}{24}f^{(4)}(x_0)h^4 - \frac{1}{120}f^{(5)}(\xi_2)h^5, \end{aligned} \tag{4.19}$$

where $x_0 - h < \xi_2 < x_0 < \xi_1 < x_0 + h$.

Subtracting Eq. (4.19) from Eq. (4.18) gives a new approximation for $f'(x)$.

$$f(x_0+h)-f(x_0-h)=2hf'(x_0)+\frac{h^3}{3}f'''(x_0)+\frac{h^5}{120}[f^{(5)}(\xi_1)+f^{(5)}(\xi_2)], \quad (4.20)$$

which implies that

$$f'(x_0)=\frac{1}{2h}[f(x_0+h)-f(x_0-h)]-\frac{h^2}{6}f'''(x_0)-\frac{h^4}{240}[f^{(5)}(\xi_1)+f^{(5)}(\xi_2)].$$

If $f^{(5)}$ is continuous on $[x_0-h, x_0+h]$, the Intermediate Value Theorem 1.11 implies that a number $\tilde{\xi}$ in (x_0-h, x_0+h) exists with

$$f^{(5)}(\tilde{\xi})=\frac{1}{2}\left[f^{(5)}(\xi_1)+f^{(5)}(\xi_2)\right].$$

As a consequence,we have the $O(h^2)$ approximation

$$f'(x_0)=\frac{1}{2h}[f(x_0+h)-f(x_0-h)]-\frac{h^2}{6}f'''(x_0)-\frac{h^4}{120}f^{(5)}(\tilde{\xi}). \quad (4.21)$$

Although the approximation in Eq. (4.21) is the same as that given in the three-point formula in Eq. (4.5), the unknown evaluation point occurs now in $f^{(5)}$, rather than in f'''. Extrapolation takes advantage of this by first replacing h in Eq. (4.21) with $2h$ to give the new formula

$$f'(x_0)=\frac{1}{4h}[f(x_0+2h)-f(x_0-2h)]-\frac{4h^2}{6}f'''(x_0)-\frac{16h^4}{120}f^{(5)}(\hat{\xi}), \quad (4.22)$$

where $\hat{\xi}$ is between x_0-2h and x_0+2h.

Multiplying Eq. (4.21) by 4 and subtracting Eq. (4.22) produces

$$\begin{aligned}3f'(x_0)=&\frac{2}{h}[f(x_0+h)-f(x_0-h)]-\frac{1}{4h}[f(x_0+2h)-f(x_0-2h)]\\&-\frac{h^4}{30}f^{(5)}(\tilde{\xi})+\frac{2h^4}{15}f^{(5)}(\hat{\xi}).\end{aligned}$$

Even if $f^{(5)}$ is continuous on $[x_0-2h, x_0+2h]$, the Intermediate Value Theorem 1.11 cannot be applied as we did to derive Eq. (4.21) because here we have the *difference* of terms involving $f^{(5)}$. However, an alternative method can be used to show that $f^{(5)}(\tilde{\xi})$ and $f^{(5)}(\hat{\xi})$ can still be replaced by a common value $f^{(5)}(\xi)$. Assuming this and dividing by 3 produces the five-point midpoint formula Eq. (4.6) that we saw in Section 4.1

$$f'(x_0)=\frac{1}{12h}[f(x_0-2h)-8f(x_0-h)+8f(x_0+h)-f(x_0+2h)]+\frac{h^4}{30}f^{(5)}(\xi). \quad \square$$

Other formulas for first and higher derivatives can be derived in a similar manner. See, for example, Exercise 8.

The technique of extrapolation is used throughout the text. The most prominent applications occur in approximating integrals in Section 4.5 and for determining approximate solutions to differential equations in Section 5.8.

EXERCISE SET 4.2

1. Apply the extrapolation process described in Example 1 to determine $N_3(h)$, an approximation to $f'(x_0)$, for the following functions and stepsizes.
   a. $f(x) = \ln x$, $x_0 = 1.0$, $h = 0.4$
   b. $f(x) = x + e^x$, $x_0 = 0.0$, $h = 0.4$
   c. $f(x) = 2^x \sin x$, $x_0 = 1.05$, $h = 0.4$
   d. $f(x) = x^3 \cos x$, $x_0 = 2.3$, $h = 0.4$
2. Add another line to the extrapolation table in Exercise 1 to obtain the approximation $N_4(h)$.
3. Repeat Exercise 1 using four-digit rounding arithmetic.
4. Repeat Exercise 2 using four-digit rounding arithmetic.
5. The following data give approximations to the integral

$$M = \int_0^{\pi} \sin x \, dx.$$

$$N_1(h) = 1.570796, \quad N_1\left(\frac{h}{2}\right) = 1.896119, \quad N_1\left(\frac{h}{4}\right) = 1.974232, \quad N_1\left(\frac{h}{8}\right) = 1.993570.$$

   Assuming $M = N_1(h) + K_1h^2 + K_2h^4 + K_3h^6 + K_4h^8 + O(h^{10})$, construct an extrapolation table to determine $N_4(h)$.
6. The following data can be used to approximate the integral

$$M = \int_0^{3\pi/2} \cos x \, dx.$$

$$N_1(h) = 2.356194, \qquad N_1\left(\frac{h}{2}\right) = -0.4879837,$$

$$N_1\left(\frac{h}{4}\right) = -0.8815732, \quad N_1\left(\frac{h}{8}\right) = -0.9709157.$$

   Assume a formula exists of the type given in Exercise 5 and determine $N_4(h)$.
7. Show that the five-point formula in Eq. (4.6) applied to $f(x) = xe^x$ at $x_0 = 2.0$ gives $N_2(0.2)$ in Table 4.6 when $h = 0.1$ and $N_2(0.1)$ when $h = 0.05$.
8. The forward-difference formula can be expressed as

$$f'(x_0) = \frac{1}{h}[f(x_0 + h) - f(x_0)] - \frac{h}{2}f''(x_0) - \frac{h^2}{6}f'''(x_0) + O(h^3).$$

   Use extrapolation to derive an $O(h^3)$ formula for $f'(x_0)$.
9. Suppose that $N(h)$ is an approximation to M for every $h > 0$ and that

$$M = N(h) + K_1h + K_2h^2 + K_3h^3 + \cdots,$$

   for some constants $K_1, K_2, K_3, \ldots$. Use the values $N(h)$, $N\left(\frac{h}{3}\right)$, and $N\left(\frac{h}{9}\right)$ to produce an $O(h^3)$ approximation to M.
10. Suppose that $N(h)$ is an approximation to M for every $h > 0$ and that

$$M = N(h) + K_1h^2 + K_2h^4 + K_3h^6 + \cdots,$$

   for some constants $K_1, K_2, K_3, \ldots$. Use the values $N(h)$, $N\left(\frac{h}{3}\right)$, and $N\left(\frac{h}{9}\right)$ to produce an $O(h^6)$ approximation to M.
11. In calculus, we learn that $e = \lim_{h\to 0}(1 + h)^{1/h}$.
    a. Determine approximations to e corresponding to $h = 0.04$, 0.02, and 0.01.
    b. Use extrapolation on the approximations, assuming that constants $K_1, K_2, \ldots$ exist with $e = (1 + h)^{1/h} + K_1h + K_2h^2 + K_3h^3 + \cdots$, to produce an $O(h^3)$ approximation to e, where $h = 0.04$.
    c. Do you think that the assumption in part (b) is correct?

12. **a.** Show that

$$\lim_{h\to 0}\left(\frac{2+h}{2-h}\right)^{1/h} = e.$$

b. Compute approximations to e using the formula $N(h) = \left(\frac{2+h}{2-h}\right)^{1/h}$, for $h = 0.04, 0.02$, and 0.01.

c. Assume that $e = N(h) + K_1h + K_2h^2 + K_3h^3 + \cdots$. Use extrapolation, with at least 16 digits of precision, to compute an $O(h^3)$ approximation to e with $h = 0.04$. Do you think the assumption is correct?

d. Show that $N(-h) = N(h)$.

e. Use part (d) to show that $K_1 = K_3 = K_5 = \cdots = 0$ in the formula

$$e = N(h) + K_1h + K_2h^2 + K_3h^3K_4h^4 + K_5h^5 + \cdots,$$

so that the formula reduces to

$$e = N(h) + K_2h^2 + K_4h^4 + K_6h^6 + \cdots.$$

f. Use the results of part (e) and extrapolation to compute an $O(h^6)$ approximation to e with $h = 0.04$.

13. Suppose the following extrapolation table has been constructed to approximate the number M with $M = N_1(h) + K_1h^2 + K_2h^4 + K_3h^6$:

$N_1(h)$		
$N_1\left(\frac{h}{2}\right)$	$N_2(h)$	
$N_1\left(\frac{h}{4}\right)$	$N_2\left(\frac{h}{2}\right)$	$N_3(h)$

a. Show that the linear interpolating polynomial $P_{0,1}(h)$ through $(h^2, N_1(h))$ and $(h^2/4, N_1(h/2))$ satisfies $P_{0,1}(0) = N_2(h)$. Similarly, show that $P_{1,2}(0) = N_2(h/2)$.

b. Show that the linear interpolating polynomial $P_{0,2}(h)$ through $(h^4, N_2(h))$ and $(h^4/16, N_2(h/2))$ satisfies $P_{0,2}(0) = N_3(h)$.

14. Suppose that $N_1(h)$ is a formula that produces $O(h)$ approximations to a number M and that

$$M = N_1(h) + K_1h + K_2h^2 + \cdots,$$

for a collection of positive constants $K_1, K_2, \ldots$. Then $N_1(h)$, $N_1(h/2)$, $N_1(h/4)$, ... are all lower bounds for M. What can be said about the extrapolated approximations $N_2(h), N_3(h), \ldots$?

15. The semiperimeters of regular polygons with k sides that inscribe and circumscribe the unit circle were used by Archimedes before 200 B.C.E. to approximate π, the circumference of a semicircle. Geometry can be used to show that the sequence of inscribed and circumscribed semiperimeters $\{p_k\}$ and $\{P_k\}$, respectively, satisfy

$$p_k = k\sin\left(\frac{\pi}{k}\right) \quad \text{and} \quad P_k = k\tan\left(\frac{\pi}{k}\right),$$

with $p_k < \pi < P_k$, whenever $k \geq 4$.

a. Show that $p_4 = 2\sqrt{2}$ and $P_4 = 4$.

b. Show that for $k \geq 4$, the sequences satisfy the recurrence relations

$$P_{2k} = \frac{2p_kP_k}{p_k + P_k} \quad \text{and} \quad p_{2k} = \sqrt{p_kP_{2k}}.$$

c. Approximate π to within 10^{-4} by computing p_k and P_k until $P_k - p_k < 10^{-4}$.

d. Use Taylor Series to show that

$$\pi = p_k + \frac{\pi^3}{3!}\left(\frac{1}{k}\right)^2 - \frac{\pi^5}{5!}\left(\frac{1}{k}\right)^4 + \cdots$$

and

$$\pi = P_k - \frac{\pi^3}{3}\left(\frac{1}{k}\right)^2 + \frac{2\pi^5}{15}\left(\frac{1}{k}\right)^4 - \cdots.$$

e. Use extrapolation with $h = 1/k$ to better approximate π.

4.3 Elements of Numerical Integration

The need often arises for evaluating the definite integral of a function that has no explicit antiderivative or whose antiderivative is not easy to obtain. The basic method involved in approximating $\int_a^b f(x)\,dx$ is called **numerical quadrature**. It uses a sum $\sum_{i=0}^n a_i f(x_i)$ to approximate $\int_a^b f(x)\,dx$.

The methods of quadrature in this section are based on the interpolation polynomials given in Chapter 3. The basic idea is to select a set of distinct nodes $\{x_0, \ldots, x_n\}$ from the interval $[a, b]$. Then integrate the Lagrange interpolating polynomial

$$P_n(x) = \sum_{i=0}^n f(x_i)L_i(x)$$

and its truncation error term over $[a, b]$ to obtain

$$\int_a^b f(x)\,dx = \int_a^b \sum_{i=0}^n f(x_i)L_i(x)\,dx + \int_a^b \prod_{i=0}^n (x - x_i)\frac{f^{(n+1)}(\xi(x))}{(n+1)!}\,dx$$
$$= \sum_{i=0}^n a_i f(x_i) + \frac{1}{(n+1)!}\int_a^b \prod_{i=0}^n (x - x_i) f^{(n+1)}(\xi(x))\,dx,$$

where $\xi(x)$ is in $[a, b]$ for each x and

$$a_i = \int_a^b L_i(x)\,dx, \quad \text{for each } i = 0, 1, \ldots, n.$$

The quadrature formula is, therefore,

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^n a_i f(x_i),$$

with error given by

$$E(f) = \frac{1}{(n+1)!}\int_a^b \prod_{i=0}^n (x - x_i) f^{(n+1)}(\xi(x))\,dx.$$

Before discussing the general situation of quadrature formulas, let us consider formulas produced by using first and second Lagrange polynomials with equally-spaced nodes. This gives the **Trapezoidal rule** and **Simpson's rule**, which are commonly introduced in calculus courses.

The Trapezoidal Rule

To derive the Trapezoidal rule for approximating $\int_a^b f(x)\,dx$, let $x_0 = a$, $x_1 = b$, $h = b - a$ and use the linear Lagrange polynomial:

$$P_1(x) = \frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1).$$

Then

$$\int_a^b f(x)\,dx = \int_{x_0}^{x_1} \left[\frac{(x - x_1)}{(x_0 - x_1)} f(x_0) + \frac{(x - x_0)}{(x_1 - x_0)} f(x_1) \right] dx + \frac{1}{2} \int_{x_0}^{x_1} f''(\xi(x))(x - x_0)(x - x_1)\,dx. \tag{4.23}$$

The product $(x - x_0)(x - x_1)$ does not change sign on $[x_0, x_1]$, so the Weighted Mean Value Theorem for Integrals 1.13 can be applied to the error term to give, for some ξ in (x_0, x_1),

$$\begin{aligned}\int_{x_0}^{x_1} f''(\xi(x))(x - x_0)(x - x_1)\,dx &= f''(\xi) \int_{x_0}^{x_1} (x - x_0)(x - x_1)\,dx \\ &= f''(\xi) \left[\frac{x^3}{3} - \frac{(x_1 + x_0)}{2} x^2 + x_0 x_1 x \right]_{x_0}^{x_1} \\ &= -\frac{h^3}{6} f''(\xi).\end{aligned}$$

When we use the term *trapezoid* we mean a four-sided figure that has at least two of its sides parallel. The European term for this figure is *trapezium*. To further confuse the issue, the European word trapezoidal refers to a four-sided figure with no sides equal, and the American word for this type of figure is trapezium.

Consequently, Eq. (4.23) implies that

$$\begin{aligned}\int_a^b f(x)\,dx &= \left[\frac{(x - x_1)^2}{2(x_0 - x_1)} f(x_0) + \frac{(x - x_0)^2}{2(x_1 - x_0)} f(x_1) \right]_{x_0}^{x_1} - \frac{h^3}{12} f''(\xi) \\ &= \frac{(x_1 - x_0)}{2} [f(x_0) + f(x_1)] - \frac{h^3}{12} f''(\xi).\end{aligned}$$

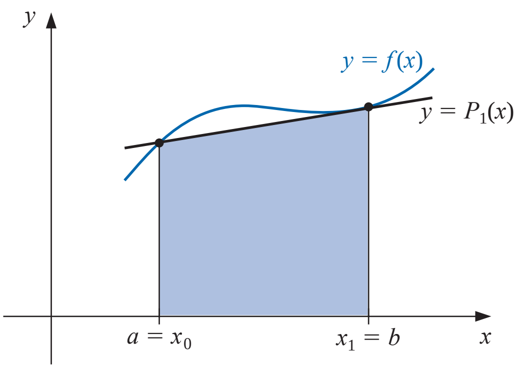
Using the notation $h = x_1 - x_0$ gives the following rule:

Trapezoidal Rule:

$$\int_a^b f(x)\,dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12} f''(\xi).$$

This is called the Trapezoidal rule because when f is a function with positive values, $\int_a^b f(x)\,dx$ is approximated by the area in a trapezoid, as shown in Figure 4.3.

Figure 4.3

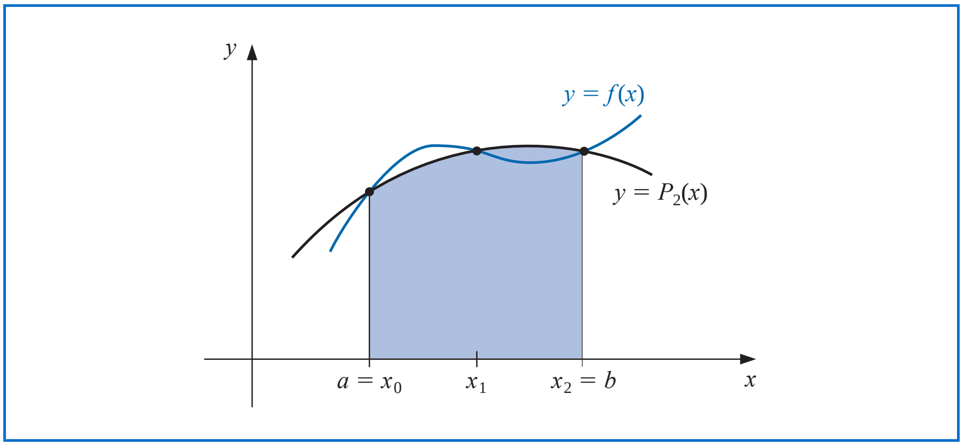


The error term for the Trapezoidal rule involves f'', so the rule gives the exact result when applied to any function whose second derivative is identically zero, that is, any polynomial of degree one or less.

Simpson's Rule

Simpson's rule results from integrating over $[a, b]$ the second Lagrange polynomial with equally-spaced nodes $x_0 = a$, $x_2 = b$, and $x_1 = a + h$, where $h = (b - a)/2$. (See Figure 4.4.)

Figure 4.4



Therefore

$$\int_a^b f(x)\,dx = \int_{x_0}^{x_2} \left[\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f(x_0) + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f(x_1) + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f(x_2) \right] dx + \int_{x_0}^{x_2} \frac{(x-x_0)(x-x_1)(x-x_2)}{6} f^{(3)}(\xi(x))\,dx.$$

Deriving Simpson's rule in this manner, however, provides only an $O(h^4)$ error term involving $f^{(3)}$. By approaching the problem in another way, a higher-order term involving $f^{(4)}$ can be derived.

To illustrate this alternative method, suppose that f is expanded in the third Taylor polynomial about x_1. Then for each x in $[x_0, x_2]$, a number $\xi(x)$ in (x_0, x_2) exists with

$$f(x) = f(x_1) + f'(x_1)(x-x_1) + \frac{f''(x_1)}{2}(x-x_1)^2 + \frac{f'''(x_1)}{6}(x-x_1)^3 + \frac{f^{(4)}(\xi(x))}{24}(x-x_1)^4$$

and

$$\int_{x_0}^{x_2} f(x)\,dx = \left[f(x_1)(x-x_1) + \frac{f'(x_1)}{2}(x-x_1)^2 + \frac{f''(x_1)}{6}(x-x_1)^3 + \frac{f'''(x_1)}{24}(x-x_1)^4 \right]_{x_0}^{x_2} + \frac{1}{24}\int_{x_0}^{x_2} f^{(4)}(\xi(x))(x-x_1)^4\,dx. \quad (4.24)$$

Because $(x - x_1)^4$ is never negative on $[x_0, x_2]$, the Weighted Mean Value Theorem for Integrals 1.13 implies that

$$\frac{1}{24}\int_{x_0}^{x_2} f^{(4)}(\xi(x))(x - x_1)^4\,dx = \frac{f^{(4)}(\xi_1)}{24}\int_{x_0}^{x_2}(x - x_1)^4\,dx = \frac{f^{(4)}(\xi_1)}{120}(x - x_1)^5\Big]_{x_0}^{x_2},$$

for some number ξ_1 in (x_0, x_2).

However, $h = x_2 - x_1 = x_1 - x_0$, so

$$(x_2 - x_1)^2 - (x_0 - x_1)^2 = (x_2 - x_1)^4 - (x_0 - x_1)^4 = 0,$$

whereas

$$(x_2 - x_1)^3 - (x_0 - x_1)^3 = 2h^3 \quad \text{and} \quad (x_2 - x_1)^5 - (x_0 - x_1)^5 = 2h^5.$$

Consequently, Eq. (4.24) can be rewritten as

$$\int_{x_0}^{x_2} f(x)\,dx = 2hf(x_1) + \frac{h^3}{3}f''(x_1) + \frac{f^{(4)}(\xi_1)}{60}h^5.$$

If we now replace $f''(x_1)$ by the approximation given in Eq. (4.9) of Section 4.1, we have

$$\int_{x_0}^{x_2} f(x)\,dx = 2hf(x_1) + \frac{h^3}{3}\left\{\frac{1}{h^2}[f(x_0) - 2f(x_1) + f(x_2)] - \frac{h^2}{12}f^{(4)}(\xi_2)\right\} + \frac{f^{(4)}(\xi_1)}{60}h^5$$

$$= \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{12}\left[\frac{1}{3}f^{(4)}(\xi_2) - \frac{1}{5}f^{(4)}(\xi_1)\right].$$

It can be shown by alternative methods (see Exercise 24) that the values ξ_1 and ξ_2 in this expression can be replaced by a common value ξ in (x_0, x_2). This gives Simpson's rule.

Thomas Simpson (1710–1761) was a self-taught mathematician who supported himself during his early years as a weaver. His primary interest was probability theory, although in 1750 he published a two-volume calculus book entitled *The Doctrine and Application of Fluxions*.

Simpson's Rule:

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90}f^{(4)}(\xi).$$

The error term in Simpson's rule involves the fourth derivative of f, so it gives exact results when applied to any polynomial of degree three or less.

Example 1 Compare the Trapezoidal rule and Simpson's rule approximations to $\int_0^2 f(x)\,dx$ when $f(x)$ is

(a) x^2 **(b)** x^4 **(c)** $(x+1)^{-1}$

(d) $\sqrt{1+x^2}$ **(e)** $\sin x$ **(f)** e^x

Solution On $[0, 2]$ the Trapezoidal and Simpson's rule have the forms

$$\text{Trapezoid:} \quad \int_0^2 f(x)\,dx \approx f(0) + f(2) \quad \text{and}$$

$$\text{Simpson's:} \quad \int_0^2 f(x)\,dx \approx \frac{1}{3}[f(0) + 4f(1) + f(2)].$$

When $f(x) = x^2$ they give

$$\text{Trapezoid:} \quad \int_0^2 f(x)\,dx \approx 0^2 + 2^2 = 4 \quad \text{and}$$

$$\text{Simpson's:} \quad \int_0^2 f(x)\,dx \approx \frac{1}{3}[(0^2) + 4 \cdot 1^2 + 2^2] = \frac{8}{3}.$$

The approximation from Simpson's rule is exact because its truncation error involves $f^{(4)}$, which is identically 0 when $f(x) = x^2$.

The results to three places for the functions are summarized in Table 4.7. Notice that in each instance Simpson's Rule is significantly superior. ■

Table 4.7

	(a)	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**
$f(x)$	x^2	x^4	$(x+1)^{-1}$	$\sqrt{1+x^2}$	$\sin x$	e^x
Exact value	2.667	6.400	1.099	2.958	1.416	6.389
Trapezoidal	4.000	16.000	1.333	3.326	0.909	8.389
Simpson's	2.667	6.667	1.111	2.964	1.425	6.421

Measuring Precision

The standard derivation of quadrature error formulas is based on determining the class of polynomials for which these formulas produce exact results. The next definition is used to facilitate the discussion of this derivation.

Definition 4.1 The **degree of accuracy**, or **precision**, of a quadrature formula is the largest positive integer n such that the formula is exact for x^k, for each $k = 0, 1, \ldots, n$. ■

The improved accuracy of Simpson's rule over the Trapezoidal rule is intuitively explained by the fact that Simpson's rule includes a midpoint evaluation that provides better balance to the approximation.

Definition 4.1 implies that the Trapezoidal and Simpson's rules have degrees of precision one and three, respectively.

Integration and summation are linear operations; that is,

$$\int_a^b (\alpha f(x) + \beta g(x))\,dx = \alpha \int_a^b f(x)\,dx + \beta \int_a^b g(x)\,dx$$

and

$$\sum_{i=0}^{n} (\alpha f(x_i) + \beta g(x_i)) = \alpha \sum_{i=0}^{n} f(x_i) + \beta \sum_{i=0}^{n} g(x_i),$$

for each pair of integrable functions f and g and each pair of real constants α and β. This implies (see Exercise 25) that:

The open and closed terminology for methods implies that the open methods use as nodes only points in the open interval, (a, b) to approximate $\int_a^b f(x)\,dx$. The closed methods include the points a and b of the closed interval $[a, b]$ as nodes.

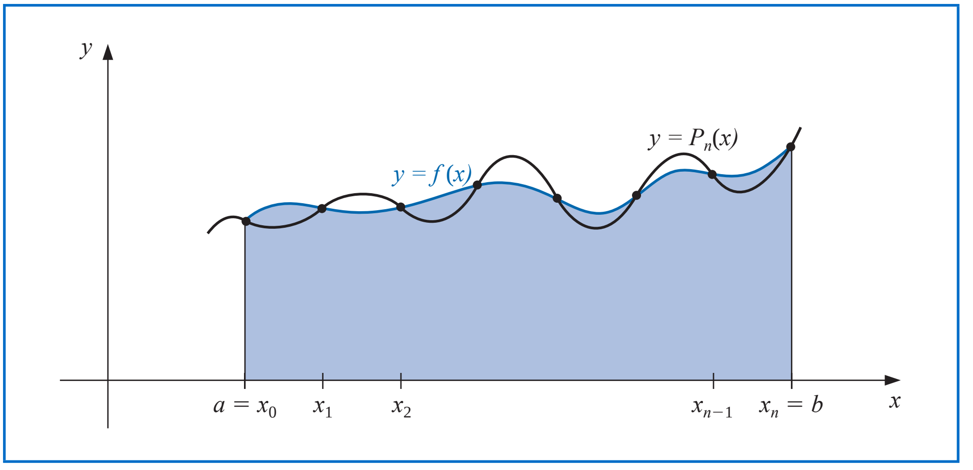
- The degree of precision of a quadrature formula is n if and only if the error is zero for all polynomials of degree $k = 0, 1, \ldots, n$, but is not zero for some polynomial of degree $n + 1$.

The Trapezoidal and Simpson's rules are examples of a class of methods known as Newton-Cotes formulas. There are two types of Newton-Cotes formulas, open and closed.

Closed Newton-Cotes Formulas

The $(n+1)$-*point closed Newton-Cotes formula* uses nodes $x_i = x_0 + ih$, for $i = 0, 1, \ldots, n$, where $x_0 = a$, $x_n = b$ and $h = (b-a)/n$. (See Figure 4.5.) It is called closed because the endpoints of the closed interval $[a, b]$ are included as nodes.

Figure 4.5



The formula assumes the form

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{n} a_i f(x_i),$$

where

$$a_i = \int_{x_0}^{x_n} L_i(x)\,dx = \int_{x_0}^{x_n} \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)}\,dx.$$

The following theorem details the error analysis associated with the closed Newton-Cotes formulas. For a proof of this theorem, see [IK], p. 313.

Theorem 4.2 Suppose that $\sum_{i=0}^{n} a_i f(x_i)$ denotes the $(n+1)$-point closed Newton-Cotes formula with $x_0 = a$, $x_n = b$, and $h = (b-a)/n$. There exists $\xi \in (a, b)$ for which

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_0^n t^2(t-1)\cdots(t-n)\,dt,$$

if n is even and $f \in C^{n+2}[a, b]$, and

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_0^n t(t-1)\cdots(t-n)\,dt,$$

if n is odd and $f \in C^{n+1}[a, b]$. ■

Roger Cotes (1682–1716) rose from a modest background to become, in 1704, the first Plumian Professor at Cambridge University. He made advances in numerous mathematical areas including numerical methods for interpolation and integration. Newton is reputed to have said of Cotes ...*if he had lived we might have known something.*

Note that when n is an even integer, the degree of precision is $n + 1$, although the interpolation polynomial is of degree at most n. When n is odd, the degree of precision is only n.

Some of the common **closed Newton-Cotes formulas** with their error terms are listed. Note that in each case the unknown value ξ lies in (a, b).

$n = 1$: Trapezoidal rule

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12} f''(\xi), \quad \text{where} \quad x_0 < \xi < x_1. \tag{4.25}$$

$n = 2$: Simpson's rule

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90} f^{(4)}(\xi), \quad \text{where} \quad x_0 < \xi < x_2. \tag{4.26}$$

$n = 3$: Simpson's Three-Eighths rule

$$\int_{x_0}^{x_3} f(x)\,dx = \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] - \frac{3h^5}{80} f^{(4)}(\xi), \tag{4.27}$$

$$\text{where} \quad x_0 < \xi < x_3.$$

$n = 4$:

$$\int_{x_0}^{x_4} f(x)\,dx = \frac{2h}{45}[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)] - \frac{8h^7}{945} f^{(6)}(\xi),$$

$$\text{where} \quad x_0 < \xi < x_4. \tag{4.28}$$

Open Newton-Cotes Formulas

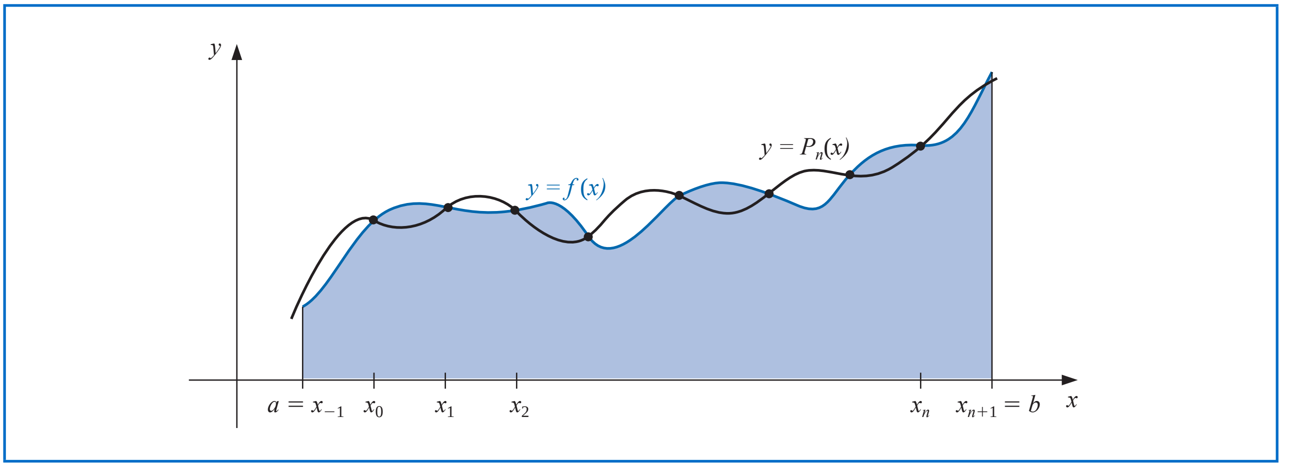
The *open Newton-Cotes formulas* do not include the endpoints of $[a, b]$ as nodes. They use the nodes $x_i = x_0 + ih$, for each $i = 0, 1, \ldots, n$, where $h = (b - a)/(n + 2)$ and $x_0 = a + h$. This implies that $x_n = b - h$, so we label the endpoints by setting $x_{-1} = a$ and $x_{n+1} = b$, as shown in Figure 4.6 on page 200. Open formulas contain all the nodes used for the approximation within the open interval (a, b). The formulas become

$$\int_a^b f(x)\,dx = \int_{x_{-1}}^{x_{n+1}} f(x)\,dx \approx \sum_{i=0}^{n} a_i f(x_i),$$

where

$$a_i = \int_a^b L_i(x)\,dx.$$

Figure 4.6



The following theorem is analogous to Theorem 4.2; its proof is contained in [IK], p. 314.

Theorem 4.3 Suppose that $\sum_{i=0}^{n} a_i f(x_i)$ denotes the $(n+1)$-point open Newton-Cotes formula with $x_{-1} = a$, $x_{n+1} = b$, and $h = (b-a)/(n+2)$. There exists $\xi \in (a, b)$ for which

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_{-1}^{n+1} t^2(t-1)\cdots(t-n)\,dt,$$

if n is even and $f \in C^{n+2}[a, b]$, and

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n} a_i f(x_i) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_{-1}^{n+1} t(t-1)\cdots(t-n)\,dt,$$

if n is odd and $f \in C^{n+1}[a, b]$. ■

Notice, as in the case of the closed methods, we have the degree of precision comparatively higher for the even methods than for the odd methods.

Some of the common **open Newton-Cotes** formulas with their error terms are as follows:

$n = 0$: Midpoint rule

$$\int_{x_{-1}}^{x_1} f(x)\,dx = 2h f(x_0) + \frac{h^3}{3} f''(\xi), \quad \text{where} \quad x_{-1} < \xi < x_1. \tag{4.29}$$

$n = 1$:

$$\int_{x_{-1}}^{x_2} f(x)\,dx = \frac{3h}{2}[f(x_0) + f(x_1)] + \frac{3h^3}{4} f''(\xi), \quad \text{where} \quad x_{-1} < \xi < x_2. \tag{4.30}$$

***n* = 2:**

$$\int_{x_{-1}}^{x_3} f(x)\,dx = \frac{4h}{3}[2f(x_0) - f(x_1) + 2f(x_2)] + \frac{14h^5}{45} f^{(4)}(\xi), \tag{4.31}$$

$$\text{where} \quad x_{-1} < \xi < x_3.$$

***n* = 3:**

$$\int_{x_{-1}}^{x_4} f(x)\,dx = \frac{5h}{24}[11f(x_0) + f(x_1) + f(x_2) + 11f(x_3)] + \frac{95}{144} h^5 f^{(4)}(\xi), \tag{4.32}$$

$$\text{where} \quad x_{-1} < \xi < x_4.$$

Example 2 Compare the results of the closed and open Newton-Cotes formulas listed as (4.25)–(4.28) and (4.29)–(4.32) when approximating

$$\int_0^{\pi/4} \sin x\,dx = 1 - \sqrt{2}/2 \approx 0.29289322.$$

Solution For the closed formulas we have

$$n = 1: \quad \frac{(\pi/4)}{2}\left[\sin 0 + \sin\frac{\pi}{4}\right] \approx 0.27768018$$

$$n = 2: \quad \frac{(\pi/8)}{3}\left[\sin 0 + 4\sin\frac{\pi}{8} + \sin\frac{\pi}{4}\right] \approx 0.29293264$$

$$n = 3: \quad \frac{3(\pi/12)}{8}\left[\sin 0 + 3\sin\frac{\pi}{12} + 3\sin\frac{\pi}{6} + \sin\frac{\pi}{4}\right] \approx 0.29291070$$

$$n = 4: \quad \frac{2(\pi/16)}{45}\left[7\sin 0 + 32\sin\frac{\pi}{16} + 12\sin\frac{\pi}{8} + 32\sin\frac{3\pi}{16} + 7\sin\frac{\pi}{4}\right] \approx 0.29289318$$

and for the open formulas we have

$$n = 0: \quad 2(\pi/8)\left[\sin\frac{\pi}{8}\right] \approx 0.30055887$$

$$n = 1: \quad \frac{3(\pi/12)}{2}\left[\sin\frac{\pi}{12} + \sin\frac{\pi}{6}\right] \approx 0.29798754$$

$$n = 2: \quad \frac{4(\pi/16)}{3}\left[2\sin\frac{\pi}{16} - \sin\frac{\pi}{8} + 2\sin\frac{3\pi}{16}\right] \approx 0.29285866$$

$$n = 3: \quad \frac{5(\pi/20)}{24}\left[11\sin\frac{\pi}{20} + \sin\frac{\pi}{10} + \sin\frac{3\pi}{20} + 11\sin\frac{\pi}{5}\right] \approx 0.29286923$$

Table 4.8 summarizes these results and shows the approximation errors. ■

Table 4.8

n	0	1	2	3	4
Closed formulas		0.27768018	0.29293264	0.29291070	0.29289318
Error		0.01521303	0.00003942	0.00001748	0.00000004
Open formulas	0.30055887	0.29798754	0.29285866	0.29286923	
Error	0.00766565	0.00509432	0.00003456	0.00002399	

EXERCISE SET 4.3

1. Approximate the following integrals using the Trapezoidal rule.

   a. $\int_{0.5}^{1} x^4\,dx$ **b.** $\int_{0}^{0.5} \frac{2}{x-4}\,dx$

   c. $\int_{1}^{1.5} x^2 \ln x\,dx$ **d.** $\int_{0}^{1} x^2 e^{-x}\,dx$

   e. $\int_{1}^{1.6} \frac{2x}{x^2-4}\,dx$ **f.** $\int_{0}^{0.35} \frac{2}{x^2-4}\,dx$

   g. $\int_{0}^{\pi/4} x \sin x\,dx$ **h.** $\int_{0}^{\pi/4} e^{3x} \sin 2x\,dx$

2. Approximate the following integrals using the Trapezoidal rule.

   a. $\int_{-0.25}^{0.25} (\cos x)^2\,dx$ **b.** $\int_{-0.5}^{0} x \ln(x+1)\,dx$

   c. $\int_{0.75}^{1.3} \left((\sin x)^2 - 2x \sin x + 1\right)\,dx$ **d.** $\int_{e}^{e+1} \frac{1}{x \ln x}\,dx$

3. Find a bound for the error in Exercise 1 using the error formula, and compare this to the actual error.
4. Find a bound for the error in Exercise 2 using the error formula, and compare this to the actual error.
5. Repeat Exercise 1 using Simpson's rule.
6. Repeat Exercise 2 using Simpson's rule.
7. Repeat Exercise 3 using Simpson's rule and the results of Exercise 5.
8. Repeat Exercise 4 using Simpson's rule and the results of Exercise 6.
9. Repeat Exercise 1 using the Midpoint rule.
10. Repeat Exercise 2 using the Midpoint rule.
11. Repeat Exercise 3 using the Midpoint rule and the results of Exercise 9.
12. Repeat Exercise 4 using the Midpoint rule and the results of Exercise 10.
13. The Trapezoidal rule applied to $\int_0^2 f(x)\,dx$ gives the value 4, and Simpson's rule gives the value 2. What is $f(1)$?
14. The Trapezoidal rule applied to $\int_0^2 f(x)\,dx$ gives the value 5, and the Midpoint rule gives the value 4. What value does Simpson's rule give?
15. Find the degree of precision of the quadrature formula

$$\int_{-1}^{1} f(x)\,dx = f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right).$$

16. Let $h = (b-a)/3$, $x_0 = a$, $x_1 = a+h$, and $x_2 = b$. Find the degree of precision of the quadrature formula

$$\int_a^b f(x)\,dx = \frac{9}{4}hf(x_1) + \frac{3}{4}hf(x_2).$$

17. The quadrature formula $\int_{-1}^{1} f(x)\,dx = c_0 f(-1) + c_1 f(0) + c_2 f(1)$ is exact for all polynomials of degree less than or equal to 2. Determine c_0, c_1, and c_2.
18. The quadrature formula $\int_0^2 f(x)\,dx = c_0 f(0) + c_1 f(1) + c_2 f(2)$ is exact for all polynomials of degree less than or equal to 2. Determine c_0, c_1, and c_2.
19. Find the constants c_0, c_1, and x_1 so that the quadrature formula

$$\int_0^1 f(x)\,dx = c_0 f(0) + c_1 f(x_1)$$

has the highest possible degree of precision.

20. Find the constants x_0, x_1, and c_1 so that the quadrature formula

$$\int_0^1 f(x)\,dx = \frac{1}{2}f(x_0) + c_1 f(x_1)$$

has the highest possible degree of precision.

21. Approximate the following integrals using formulas (4.25) through (4.32). Are the accuracies of the approximations consistent with the error formulas? Which of parts (d) and (e) give the better approximation?

a. $\int_0^{0.1} \sqrt{1+x}\, dx$

b. $\int_0^{\pi/2} (\sin x)^2\, dx$

c. $\int_{1.1}^{1.5} e^x\, dx$

d. $\int_1^{10} \frac{1}{x}\, dx$

e. $\int_1^{5.5} \frac{1}{x}\, dx + \int_{5.5}^{10} \frac{1}{x}\, dx$

f. $\int_0^1 x^{1/3}\, dx$

22. Given the function f at the following values,

x	1.8	2.0	2.2	2.4	2.6
$f(x)$	3.12014	4.42569	6.04241	8.03014	10.46675

approximate $\int_{1.8}^{2.6} f(x)\, dx$ using all the appropriate quadrature formulas of this section.

23. Suppose that the data of Exercise 22 have round-off errors given by the following table.

x	1.8	2.0	2.2	2.4	2.6
Error in $f(x)$	2×10^{-6}	-2×10^{-6}	-0.9×10^{-6}	-0.9×10^{-6}	2×10^{-6}

Calculate the errors due to round-off in Exercise 22.

24. Derive Simpson's rule with error term by using

$$\int_{x_0}^{x_2} f(x)\, dx = a_0 f(x_0) + a_1 f(x_1) + a_2 f(x_2) + k f^{(4)}(\xi).$$

Find a_0, a_1, and a_2 from the fact that Simpson's rule is exact for $f(x) = x^n$ when $n = 1, 2$, and 3. Then find k by applying the integration formula with $f(x) = x^4$.

25. Prove the statement following Definition 4.1; that is, show that a quadrature formula has degree of precision n if and only if the error $E(P(x)) = 0$ for all polynomials $P(x)$ of degree $k = 0, 1, \ldots, n$, but $E(P(x)) \neq 0$ for some polynomial $P(x)$ of degree $n + 1$.

26. Derive Simpson's three-eighths rule (the closed rule with $n = 3$) with error term by using Theorem 4.2.

27. Derive the open rule with $n = 1$ with error term by using Theorem 4.3.

4.4 Composite Numerical Integration

The Newton-Cotes formulas are generally unsuitable for use over large integration intervals. High-degree formulas would be required, and the values of the coefficients in these formulas are difficult to obtain. Also, the Newton-Cotes formulas are based on interpolatory polynomials that use equally-spaced nodes, a procedure that is inaccurate over large intervals because of the oscillatory nature of high-degree polynomials.

Piecewise approximation is often effective. Recall that this was used for spline interpolation.

In this section, we discuss a *piecewise* approach to numerical integration that uses the low-order Newton-Cotes formulas. These are the techniques most often applied.

Example 1 Use Simpson's rule to approximate $\int_0^4 e^x\, dx$ and compare this to the results obtained by adding the Simpson's rule approximations for $\int_0^2 e^x\, dx$ and $\int_2^4 e^x\, dx$. Compare these approximations to the sum of Simpson's rule for $\int_0^1 e^x\, dx$, $\int_1^2 e^x\, dx$, $\int_2^3 e^x\, dx$, and $\int_3^4 e^x\, dx$.

Solution Simpson's rule on $[0, 4]$ uses $h = 2$ and gives

$$\int_0^4 e^x\,dx \approx \frac{2}{3}(e^0 + 4e^2 + e^4) = 56.76958.$$

The exact answer in this case is $e^4 - e^0 = 53.59815$, and the error -3.17143 is far larger than we would normally accept.

Applying Simpson's rule on each of the intervals $[0, 2]$ and $[2, 4]$ uses $h = 1$ and gives

$$\begin{aligned}\int_0^4 e^x\,dx &= \int_0^2 e^x\,dx + \int_2^4 e^x\,dx\\ &\approx \frac{1}{3}\left(e^0 + 4e + e^2\right) + \frac{1}{3}\left(e^2 + 4e^3 + e^4\right)\\ &= \frac{1}{3}\left(e^0 + 4e + 2e^2 + 4e^3 + e^4\right)\\ &= 53.86385.\end{aligned}$$

The error has been reduced to -0.26570.

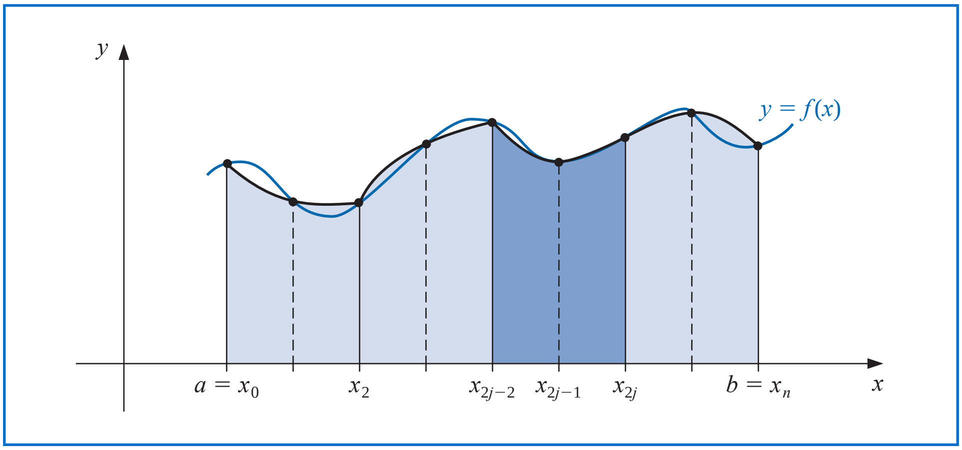
For the integrals on $[0, 1]$,$[1, 2]$,$[3, 4]$, and $[3, 4]$ we use Simpson's rule four times with $h = \frac{1}{2}$ giving

$$\begin{aligned}\int_0^4 e^x\,dx &= \int_0^1 e^x\,dx + \int_1^2 e^x\,dx + \int_2^3 e^x\,dx + \int_3^4 e^x\,dx\\ &\approx \frac{1}{6}\left(e_0 + 4e^{1/2} + e\right) + \frac{1}{6}\left(e + 4e^{3/2} + e^2\right)\\ &\quad + \frac{1}{6}\left(e^2 + 4e^{5/2} + e^3\right) + \frac{1}{6}\left(e^3 + 4e^{7/2} + e^4\right)\\ &= \frac{1}{6}\left(e^0 + 4e^{1/2} + 2e + 4e^{3/2} + 2e^2 + 4e^{5/2} + 2e^3 + 4e^{7/2} + e^4\right)\\ &= 53.61622.\end{aligned}$$

The error for this approximation has been reduced to -0.01807. ■

To generalize this procedure for an arbitrary integral $\int_a^b f(x)\,dx$, choose an even integer n. Subdivide the interval $[a, b]$ into n subintervals, and apply Simpson's rule on each consecutive pair of subintervals. (See Figure 4.7.)

Figure 4.7



With $h = (b-a)/n$ and $x_j = a + jh$, for each $j = 0, 1, \ldots, n$, we have

$$\int_a^b f(x)\,dx = \sum_{j=1}^{n/2} \int_{x_{2j-2}}^{x_{2j}} f(x)\,dx$$

$$= \sum_{j=1}^{n/2} \left\{ \frac{h}{3}[f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})] - \frac{h^5}{90} f^{(4)}(\xi_j) \right\},$$

for some ξ_j with $x_{2j-2} < \xi_j < x_{2j}$, provided that $f \in C^4[a,b]$. Using the fact that for each $j = 1, 2, \ldots, (n/2) - 1$ we have $f(x_{2j})$ appearing in the term corresponding to the interval $[x_{2j-2}, x_{2j}]$ and also in the term corresponding to the interval $[x_{2j}, x_{2j+2}]$, we can reduce this sum to

$$\int_a^b f(x)\,dx = \frac{h}{3}\left[f(x_0) + 2\sum_{j=1}^{(n/2)-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n) \right] - \frac{h^5}{90}\sum_{j=1}^{n/2} f^{(4)}(\xi_j).$$

The error associated with this approximation is

$$E(f) = -\frac{h^5}{90}\sum_{j=1}^{n/2} f^{(4)}(\xi_j),$$

where $x_{2j-2} < \xi_j < x_{2j}$, for each $j = 1, 2, \ldots, n/2$.

If $f \in C^4[a,b]$, the Extreme Value Theorem 1.9 implies that $f^{(4)}$ assumes its maximum and minimum in $[a,b]$. Since

$$\min_{x\in[a,b]} f^{(4)}(x) \le f^{(4)}(\xi_j) \le \max_{x\in[a,b]} f^{(4)}(x),$$

we have

$$\frac{n}{2}\min_{x\in[a,b]} f^{(4)}(x) \le \sum_{j=1}^{n/2} f^{(4)}(\xi_j) \le \frac{n}{2}\max_{x\in[a,b]} f^{(4)}(x)$$

and

$$\min_{x\in[a,b]} f^{(4)}(x) \le \frac{2}{n}\sum_{j=1}^{n/2} f^{(4)}(\xi_j) \le \max_{x\in[a,b]} f^{(4)}(x).$$

By the Intermediate Value Theorem 1.11, there is a $\mu \in (a,b)$ such that

$$f^{(4)}(\mu) = \frac{2}{n}\sum_{j=1}^{n/2} f^{(4)}(\xi_j).$$

Thus

$$E(f) = -\frac{h^5}{90}\sum_{j=1}^{n/2} f^{(4)}(\xi_j) = -\frac{h^5}{180} n f^{(4)}(\mu),$$

or, since $h = (b-a)/n$,

$$E(f) = -\frac{(b-a)}{180} h^4 f^{(4)}(\mu).$$

These observations produce the following result.

Theorem 4.4 Let $f \in C^4[a,b]$, n be even, $h = (b-a)/n$, and $x_j = a + jh$, for each $j = 0, 1, \ldots, n$. There exists a $\mu \in (a,b)$ for which the **Composite Simpson's rule** for n subintervals can be written with its error term as

$$\int_a^b f(x)\,dx = \frac{h}{3}\left[f(a) + 2\sum_{j=1}^{(n/2)-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(b)\right] - \frac{b-a}{180}h^4 f^{(4)}(\mu).$$

■

Notice that the error term for the Composite Simpson's rule is $O(h^4)$, whereas it was $O(h^5)$ for the standard Simpson's rule. However, these rates are not comparable because for standard Simpson's rule we have h fixed at $h = (b-a)/2$, but for Composite Simpson's rule we have $h = (b-a)/n$, for n an even integer. This permits us to considerably reduce the value of h when the Composite Simpson's rule is used.

Algorithm 4.1 uses the Composite Simpson's rule on n subintervals. This is the most frequently used general-purpose quadrature algorithm.

ALGORITHM **4.1**

Composite Simpson's Rule

To approximate the integral $I = \int_a^b f(x)\,dx$:

INPUT endpoints a, b; even positive integer n.

OUTPUT approximation XI to I.

Step 1 Set $h = (b-a)/n$.

Step 2 Set $XI0 = f(a) + f(b)$;
$XI1 = 0$; (*Summation of* $f(x_{2i-1})$.)
$XI2 = 0$. (*Summation of* $f(x_{2i})$.)

Step 3 For $i = 1, \ldots, n-1$ do Steps 4 and 5.

Step 4 Set $X = a + ih$.

Step 5 If i is even then set $XI2 = XI2 + f(X)$
else set $XI1 = XI1 + f(X)$.

Step 6 Set $XI = h(XI0 + 2 \cdot XI2 + 4 \cdot XI1)/3$.

Step 7 OUTPUT (XI);
STOP.

■

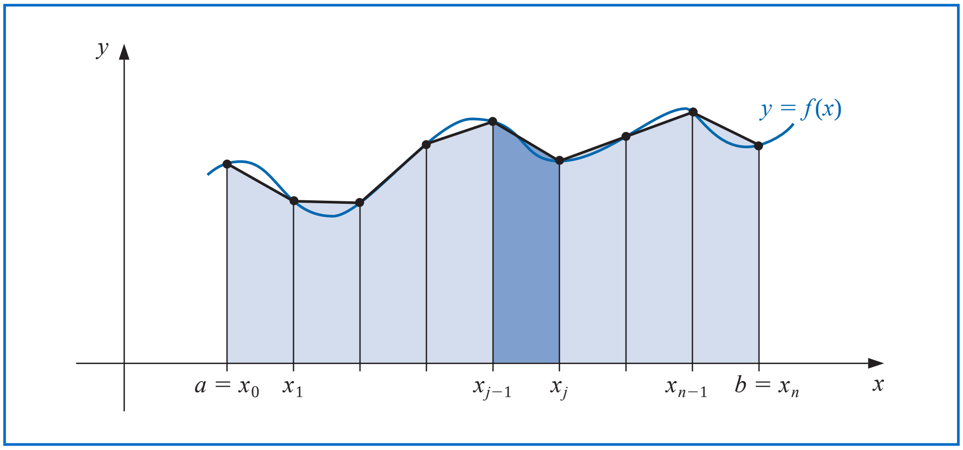
The subdivision approach can be applied to any of the Newton-Cotes formulas. The extensions of the Trapezoidal (see Figure 4.8) and Midpoint rules are given without proof. The Trapezoidal rule requires only one interval for each application, so the integer n can be either odd or even.

Theorem 4.5 Let $f \in C^2[a,b]$, $h = (b-a)/n$, and $x_j = a + jh$, for each $j = 0, 1, \ldots, n$. There exists a $\mu \in (a,b)$ for which the **Composite Trapezoidal rule** for n subintervals can be written with its error term as

$$\int_a^b f(x)\,dx = \frac{h}{2}\left[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right] - \frac{b-a}{12}h^2 f''(\mu).$$

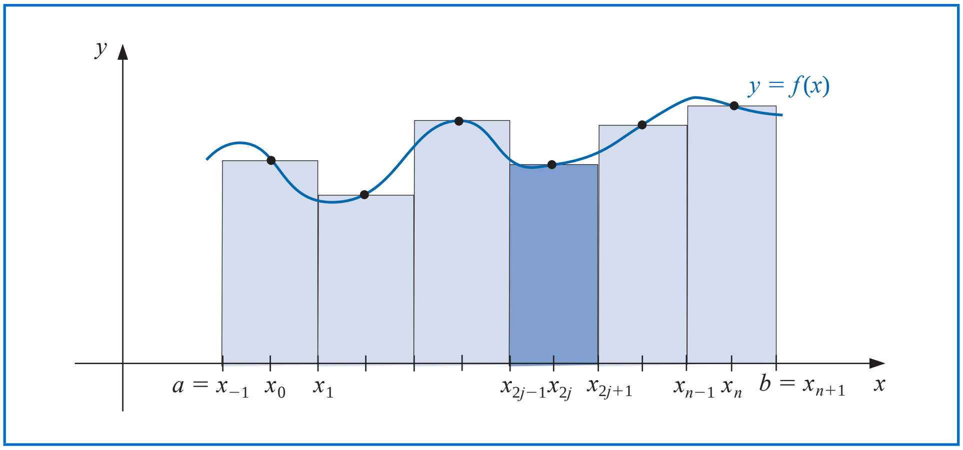
■

Figure 4.8



For the Composite Midpoint rule, n must again be even. (See Figure 4.9.)

Figure 4.9



Theorem 4.6 Let $f \in C^2[a,b]$, n be even, $h = (b-a)/(n+2)$, and $x_j = a + (j+1)h$ for each $j = -1, 0, \ldots, n+1$. There exists a $\mu \in (a,b)$ for which the **Composite Midpoint rule** for $n+2$ subintervals can be written with its error term as

$$\int_a^b f(x)\,dx = 2h\sum_{j=0}^{n/2} f(x_{2j}) + \frac{b-a}{6}h^2 f''(\mu). \qquad \blacksquare$$

Example 2 Determine values of h that will ensure an approximation error of less than 0.00002 when approximating $\int_0^\pi \sin x\,dx$ and employing
(a) Composite Trapezoidal rule and **(b)** Composite Simpson's rule.

Solution **(a)** The error form for the Composite Trapezoidal rule for $f(x) = \sin x$ on $[0,\pi]$ is

$$\left|\frac{\pi h^2}{12} f''(\mu)\right| = \left|\frac{\pi h^2}{12}(-\sin\mu)\right| = \frac{\pi h^2}{12}|\sin\mu|.$$

To ensure sufficient accuracy with this technique we need to have

$$\frac{\pi h^2}{12}|\sin\mu| \le \frac{\pi h^2}{12} < 0.00002.$$

Since $h = \pi/n$ implies that $n = \pi/h$, we need

$$\frac{\pi^3}{12n^2} < 0.00002 \quad \text{which implies that} \quad n > \left(\frac{\pi^3}{12(0.00002)}\right)^{1/2} \approx 359.44.$$

and the Composite Trapezoidal rule requires $n \ge 360$.

(b) The error form for the Composite Simpson's rule for $f(x) = \sin x$ on $[0, \pi]$ is

$$\left|\frac{\pi h^4}{180} f^{(4)}(\mu)\right| = \left|\frac{\pi h^4}{180}\sin\mu\right| = \frac{\pi h^4}{180}|\sin\mu|.$$

To ensure sufficient accuracy with this technique we need to have

$$\frac{\pi h^4}{180}|\sin\mu| \le \frac{\pi h^4}{180} < 0.00002.$$

Using again the fact that $n = \pi/h$ gives

$$\frac{\pi^5}{180n^4} < 0.00002 \quad \text{which implies that} \quad n > \left(\frac{\pi^5}{180(0.00002)}\right)^{1/4} \approx 17.07.$$

So Composite Simpson's rule requires only $n \ge 18$.

Composite Simpson's rule with $n = 18$ gives

$$\int_0^\pi \sin x\, dx \approx \frac{\pi}{54}\left[2\sum_{j=1}^{8}\sin\left(\frac{j\pi}{9}\right) + 4\sum_{j=1}^{9}\sin\left(\frac{(2j-1)\pi}{18}\right)\right] = 2.0000104.$$

This is accurate to within about 10^{-5} because the true value is $-\cos(\pi) - (-\cos(0)) = 2$. ▪

Composite Simpson's rule is the clear choice if you wish to minimize computation. For comparison purposes, consider the Composite Trapezoidal rule using $h = \pi/18$ for the integral in Example 2. This approximation uses the same function evaluations as Composite Simpson's rule but the approximation in this case

$$\int_0^\pi \sin x\, dx \approx \frac{\pi}{36}\left[2\sum_{j=1}^{17}\sin\left(\frac{j\pi}{18}\right) + \sin 0 + \sin\pi\right] = \frac{\pi}{36}\left[2\sum_{j=1}^{17}\sin\left(\frac{j\pi}{18}\right)\right] = 1.9949205.$$

is accurate only to about 5×10^{-3}.

Maple contains numerous procedures for numerical integration in the *NumericalAnalysis* subpackage of the *Student* package. First access the library as usual with

with(*Student*[*NumericalAnalysis*])

The command for all methods is *Quadrature* with the options in the call specifying the method to be used. We will use the Trapezoidal method to illustrate the procedure. First define the function and the interval of integration with

$f := x \to \sin(x);\ a := 0.0;\ b := \pi$

After Maple responds with the function and the interval, enter the command

$Quadrature(f(x), x = a..b, method = trapezoid, partition = 20, output = value)$

1.995885973

The value of the step size h in this instance is the width of the interval $b - a$ divided by the number specified by $partition = 20$.

Simpson's method can be called in a similar manner, except that the step size h is determined by $b - a$ divided by twice the value of *partition*. Hence, the Simpson's rule approximation using the same nodes as those in the Trapezoidal rule is called with

$Quadrature(f(x), x = a..b, method = simpson, partition = 10, output = value)$

2.000006785

Any of the Newton-Cotes methods can be called using the option

$$method = newtoncotes[open, n] \quad \text{or} \quad method = newtoncotes[closed, n]$$

Be careful to correctly specify the number in *partition* when an even number of divisions is required, and when an open method is employed.

Round-Off Error Stability

In Example 2 we saw that ensuring an accuracy of 2×10^{-5} for approximating $\int_0^\pi \sin x\, dx$ required 360 subdivisions of $[0, \pi]$ for the Composite Trapezoidal rule and only 18 for Composite Simpson's rule. In addition to the fact that less computation is needed for the Simpson's technique, you might suspect that because of fewer computations this method would also involve less round-off error. However, an important property shared by all the composite integration techniques is a stability with respect to round-off error. That is, the round-off error does not depend on the number of calculations performed.

Numerical integration is expected to be stable, whereas numerical differentiation is unstable.

To demonstrate this rather amazing fact, suppose we apply the Composite Simpson's rule with n subintervals to a function f on $[a, b]$ and determine the maximum bound for the round-off error. Assume that $f(x_i)$ is approximated by $\tilde{f}(x_i)$ and that

$$f(x_i) = \tilde{f}(x_i) + e_i, \quad \text{for each } i = 0, 1, \ldots, n,$$

where e_i denotes the round-off error associated with using $\tilde{f}(x_i)$ to approximate $f(x_i)$. Then the accumulated error, $e(h)$, in the Composite Simpson's rule is

$$\begin{aligned} e(h) &= \left| \frac{h}{3} \left[e_0 + 2 \sum_{j=1}^{(n/2)-1} e_{2j} + 4 \sum_{j=1}^{n/2} e_{2j-1} + e_n \right] \right| \\ &\le \frac{h}{3} \left[|e_0| + 2 \sum_{j=1}^{(n/2)-1} |e_{2j}| + 4 \sum_{j=1}^{n/2} |e_{2j-1}| + |e_n| \right]. \end{aligned}$$

If the round-off errors are uniformly bounded by ε, then

$$e(h) \le \frac{h}{3} \left[\varepsilon + 2 \left(\frac{n}{2} - 1 \right) \varepsilon + 4 \left(\frac{n}{2} \right) \varepsilon + \varepsilon \right] = \frac{h}{3} 3n\varepsilon = nh\varepsilon.$$

But $nh = b - a$, so

$$e(h) \le (b - a)\varepsilon,$$

a bound independent of h (and n). This means that, even though we may need to divide an interval into more parts to ensure accuracy, the increased computation that is required does not increase the round-off error. This result implies that the procedure is stable as h approaches zero. Recall that this was not true of the numerical differentiation procedures considered at the beginning of this chapter.

EXERCISE SET 4.4

1. Use the Composite Trapezoidal rule with the indicated values of n to approximate the following integrals.
   - **a.** $\int_1^2 x \ln x \, dx, \quad n = 4$
   - **b.** $\int_{-2}^2 x^3 e^x \, dx, \quad n = 4$
   - **c.** $\int_0^2 \frac{2}{x^2+4} \, dx, \quad n = 6$
   - **d.** $\int_0^{\pi} x^2 \cos x \, dx, \quad n = 6$
   - **e.** $\int_0^2 e^{2x} \sin 3x \, dx, \quad n = 8$
   - **f.** $\int_1^3 \frac{x}{x^2+4} \, dx, \quad n = 8$
   - **g.** $\int_3^5 \frac{1}{\sqrt{x^2-4}} \, dx, \quad n = 8$
   - **h.** $\int_0^{3\pi/8} \tan x \, dx, \quad n = 8$
2. Use the Composite Trapezoidal rule with the indicated values of n to approximate the following integrals.
   - **a.** $\int_{-0.5}^{0.5} \cos^2 x \, dx, \quad n = 4$
   - **b.** $\int_{-0.5}^{0.5} x \ln(x+1) \, dx, \quad n = 6$
   - **c.** $\int_{.75}^{1.75} (\sin^2 x - 2x \sin x + 1) \, dx, \quad n = 8$
   - **d.** $\int_e^{e+2} \frac{1}{x \ln x} \, dx, \quad n = 8$
3. Use the Composite Simpson's rule to approximate the integrals in Exercise 1.
4. Use the Composite Simpson's rule to approximate the integrals in Exercise 2.
5. Use the Composite Midpoint rule with $n+2$ subintervals to approximate the integrals in Exercise 1.
6. Use the Composite Midpoint rule with $n+2$ subintervals to approximate the integrals in Exercise 2.
7. Approximate $\int_0^2 x^2 \ln(x^2+1) \, dx$ using $h = 0.25$. Use
   - **a.** Composite Trapezoidal rule.
   - **b.** Composite Simpson's rule.
   - **c.** Composite Midpoint rule.
8. Approximate $\int_0^2 x^2 e^{-x^2} \, dx$ using $h = 0.25$. Use
   - **a.** Composite Trapezoidal rule.
   - **b.** Composite Simpson's rule.
   - **c.** Composite Midpoint rule.
9. Suppose that $f(0) = 1$, $f(0.5) = 2.5$, $f(1) = 2$, and $f(0.25) = f(0.75) = \alpha$. Find α if the Composite Trapezoidal rule with $n = 4$ gives the value 1.75 for $\int_0^1 f(x) \, dx$.
10. The Midpoint rule for approximating $\int_{-1}^1 f(x) \, dx$ gives the value 12, the Composite Midpoint rule with $n = 2$ gives 5, and Composite Simpson's rule gives 6. Use the fact that $f(-1) = f(1)$ and $f(-0.5) = f(0.5) - 1$ to determine $f(-1)$, $f(-0.5)$, $f(0)$, $f(0.5)$, and $f(1)$.
11. Determine the values of n and h required to approximate

$$\int_0^2 e^{2x} \sin 3x \, dx$$

to within 10^{-4}. Use

   - **a.** Composite Trapezoidal rule.
   - **b.** Composite Simpson's rule.
   - **c.** Composite Midpoint rule.

12. Repeat Exercise 11 for the integral $\int_0^{\pi} x^2 \cos x \, dx$.

13. Determine the values of n and h required to approximate

$$\int_0^2 \frac{1}{x+4} \, dx$$

to within 10^{-5} and compute the approximation. Use

a. Composite Trapezoidal rule.

b. Composite Simpson's rule.

c. Composite Midpoint rule.

14. Repeat Exercise 13 for the integral $\int_1^2 x \ln x \, dx$.

15. Let f be defined by

$$f(x) = \begin{cases} x^3 + 1, & 0 \le x \le 0.1, \\ 1.001 + 0.03(x - 0.1) + 0.3(x - 0.1)^2 + 2(x - 0.1)^3, & 0.1 \le x \le 0.2, \\ 1.009 + 0.15(x - 0.2) + 0.9(x - 0.2)^2 + 2(x - 0.2)^3, & 0.2 \le x \le 0.3. \end{cases}$$

a. Investigate the continuity of the derivatives of f.

b. Use the Composite Trapezoidal rule with $n = 6$ to approximate $\int_0^{0.3} f(x) \, dx$, and estimate the error using the error bound.

c. Use the Composite Simpson's rule with $n = 6$ to approximate $\int_0^{0.3} f(x) \, dx$. Are the results more accurate than in part (b)?

16. Show that the error $E(f)$ for Composite Simpson's rule can be approximated by

$$-\frac{h^4}{180}[f'''(b) - f'''(a)].$$

[*Hint:* $\sum_{j=1}^{n/2} f^{(4)}(\xi_j)(2h)$ is a Riemann Sum for $\int_a^b f^{(4)}(x) \, dx$.]

17. a. Derive an estimate for $E(f)$ in the Composite Trapezoidal rule using the method in Exercise 16.

b. Repeat part (a) for the Composite Midpoint rule.

18. Use the error estimates of Exercises 16 and 17 to estimate the errors in Exercise 12.

19. Use the error estimates of Exercises 16 and 17 to estimate the errors in Exercise 14.

20. In multivariable calculus and in statistics courses it is shown that

$$\int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-(1/2)(x/\sigma)^2} \, dx = 1,$$

for any positive σ. The function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(1/2)(x/\sigma)^2}$$

is the *normal density function* with *mean* $\mu = 0$ and *standard deviation* σ. The probability that a randomly chosen value described by this distribution lies in $[a, b]$ is given by $\int_a^b f(x) \, dx$. Approximate to within 10^{-5} the probability that a randomly chosen value described by this distribution will lie in

a. $[-\sigma, \sigma]$ **b.** $[-2\sigma, 2\sigma]$ **c.** $[-3\sigma, 3\sigma]$

21. Determine to within 10^{-6} the length of the graph of the ellipse with equation $4x^2 + 9y^2 = 36$.

22. A car laps a race track in 84 seconds. The speed of the car at each 6-second interval is determined by using a radar gun and is given from the beginning of the lap, in feet/second, by the entries in the following table.

Time	0	6	12	18	24	30	36	42	48	54	60	66	72	78	84
Speed	124	134	148	156	147	133	121	109	99	85	78	89	104	116	123

How long is the track?

23. A particle of mass m moving through a fluid is subjected to a viscous resistance R, which is a function of the velocity v. The relationship between the resistance R, velocity v, and time t is given by the equation

$$t = \int_{v(t_0)}^{v(t)} \frac{m}{R(u)}\, du.$$

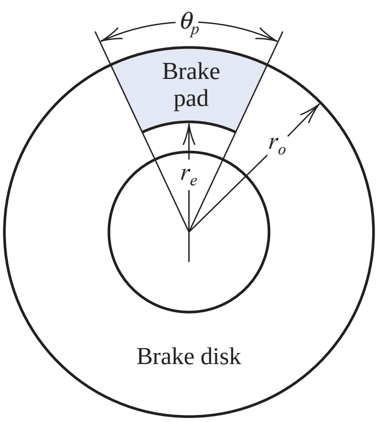
Suppose that $R(v) = -v\sqrt{v}$ for a particular fluid, where R is in newtons and v is in meters/second. If $m = 10$ kg and $v(0) = 10$ m/s, approximate the time required for the particle to slow to $v = 5$ m/s.

24. To simulate the thermal characteristics of disk brakes (see the following figure), D. A. Secrist and R. W. Hornbeck [SH] needed to approximate numerically the "area averaged lining temperature," T, of the brake pad from the equation

$$T = \frac{\displaystyle\int_{r_e}^{r_0} T(r) r\theta_p\, dr}{\displaystyle\int_{r_e}^{r_0} r\theta_p\, dr},$$

where r_e represents the radius at which the pad-disk contact begins, r_0 represents the outside radius of the pad-disk contact, θ_p represents the angle subtended by the sector brake pads, and $T(r)$ is the temperature at each point of the pad, obtained numerically from analyzing the heat equation (see Section 12.2). Suppose $r_e = 0.308$ ft, $r_0 = 0.478$ ft, $\theta_p = 0.7051$ radians, and the temperatures given in the following table have been calculated at the various points on the disk. Approximate T.

r (ft)	$T(r)$ (°F)	r (ft)	$T(r)$ (°F)	r (ft)	$T(r)$ (°F)
0.308	640	0.376	1034	0.444	1204
0.325	794	0.393	1064	0.461	1222
0.342	885	0.410	1114	0.478	1239
0.359	943	0.427	1152		



25. Find an approximation to within 10^{-4} of the value of the integral considered in the application opening this chapter:

$$\int_0^{48} \sqrt{1 + (\cos x)^2}\, dx.$$

26. The equation

$$\int_0^x \frac{1}{\sqrt{2\pi}} e^{-t^2/2}\, dt = 0.45$$

can be solved for x by using Newton's method with

$$f(x) = \int_0^x \frac{1}{\sqrt{2\pi}} e^{-t^2/2}\, dt - 0.45$$

and

$$f'(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

To evaluate f at the approximation p_k, we need a quadrature formula to approximate

$$\int_0^{p_k} \frac{1}{\sqrt{2\pi}} e^{-t^2/2}\, dt.$$

a. Find a solution to $f(x) = 0$ accurate to within 10^{-5} using Newton's method with $p_0 = 0.5$ and the Composite Simpson's rule.

b. Repeat (a) using the Composite Trapezoidal rule in place of the Composite Simpson's rule.

4.5 Romberg Integration

In this section we will illustrate how Richardson extrapolation applied to results from the Composite Trapezoidal rule can be used to obtain high accuracy approximations with little computational cost.

In Section 4.4 we found that the Composite Trapezoidal rule has a truncation error of order $O(h^2)$. Specifically, we showed that for $h = (b-a)/n$ and $x_j = a + jh$ we have

$$\int_a^b f(x)\, dx = \frac{h}{2}\left[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right] - \frac{(b-a)f''(\mu)}{12}h^2.$$

for some number μ in (a, b).

By an alternative method it can be shown (see [RR], pp. 136–140), that if $f \in C^\infty[a, b]$, the Composite Trapezoidal rule can also be written with an error term in the form

$$\int_a^b f(x)\, dx = \frac{h}{2}\left[f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right] + K_1h^2 + K_2h^4 + K_3h^6 + \cdots, \quad (4.33)$$

where each K_i is a constant that depends only on $f^{(2i-1)}(a)$ and $f^{(2i-1)}(b)$.

Recall from Section 4.2 that Richardson extrapolation can be performed on any approximation procedure whose truncation error is of the form

$$\sum_{j=1}^{m-1} K_j h^{\alpha_j} + O(h^{\alpha_m}),$$

for a collection of constants K_j and when $\alpha_1 < \alpha_2 < \alpha_3 < \cdots < \alpha_m$. In that section we gave demonstrations to illustrate how effective this techniques is when the approximation procedure has a truncation error with only even powers of h, that is, when the truncation error has the form.

$$\sum_{j=1}^{m-1} K_j h^{2j} + O(h^{2m}).$$

Werner Romberg (1909–2003) devised this procedure for improving the accuracy of the Trapezoidal rule by eliminating the successive terms in the asymptotic expansion in 1955.

Because the Composite Trapezoidal rule has this form, it is an obvious candidate for extrapolation. This results in a technique known as **Romberg integration**.

To approximate the integral $\int_a^b f(x)\,dx$ we use the results of the Composite Trapezoidal rule with $n = 1, 2, 4, 8, 16, \ldots$, and denote the resulting approximations, respectively, by $R_{1,1}, R_{2,1}, R_{3,1}$, etc. We then apply extrapolation in the manner given in Section 4.2, that is, we obtain $O(h^4)$ approximations $R_{2,2}, R_{3,2}, R_{4,2}$, etc., by

$$R_{k,2} = R_{k,1} + \frac{1}{3}(R_{k,1} - R_{k-1,1}), \quad \text{for } k = 2, 3, \ldots$$

Then $O(h^6)$ approximations $R_{3,3}, R_{4,3}, R_{5,3}$, etc., by

$$R_{k,3} = R_{k,2} + \frac{1}{15}(R_{k,2} - R_{k-1,2}), \quad \text{for } k = 3, 4, \ldots.$$

In general, after the appropriate $R_{k,j-1}$ approximations have been obtained, we determine the $O(h^{2j})$ approximations from

$$R_{k,j} = R_{k,j-1} + \frac{1}{4^{j-1} - 1}(R_{k,j-1} - R_{k-1,j-1}), \quad \text{for } k = j, j+1, \ldots$$

Example 1 Use the Composite Trapezoidal rule to find approximations to $\int_0^\pi \sin x\,dx$ with $n = 1, 2, 4, 8$, and 16. Then perform Romberg extrapolation on the results.

The Composite Trapezoidal rule for the various values of n gives the following approximations to the true value 2.

$$R_{1,1} = \frac{\pi}{2}[\sin 0 + \sin \pi] = 0;$$

$$R_{2,1} = \frac{\pi}{4}\left[\sin 0 + 2\sin\frac{\pi}{2} + \sin \pi\right] = 1.57079633;$$

$$R_{3,1} = \frac{\pi}{8}\left[\sin 0 + 2\left(\sin\frac{\pi}{4} + \sin\frac{\pi}{2} + \sin\frac{3\pi}{4}\right) + \sin \pi\right] = 1.89611890;$$

$$R_{4,1} = \frac{\pi}{16}\left[\sin 0 + 2\left(\sin\frac{\pi}{8} + \sin\frac{\pi}{4} + \cdots + \sin\frac{3\pi}{4} + \sin\frac{7\pi}{8}\right) + \sin \pi\right] = 1.97423160;$$

$$R_{5,1} = \frac{\pi}{32}\left[\sin 0 + 2\left(\sin\frac{\pi}{16} + \sin\frac{\pi}{8} + \cdots + \sin\frac{7\pi}{8} + \sin\frac{15\pi}{16}\right) + \sin \pi\right] = 1.99357034.$$

The $O(h^4)$ approximations are

$$R_{2,2} = R_{2,1} + \frac{1}{3}(R_{2,1} - R_{1,1}) = 2.09439511; \quad R_{3,2} = R_{3,1} + \frac{1}{3}(R_{3,1} - R_{2,1}) = 2.00455976;$$

$$R_{4,2} = R_{4,1} + \frac{1}{3}(R_{4,1} - R_{3,1}) = 2.00026917; \quad R_{5,2} = R_{5,1} + \frac{1}{3}(R_{5,1} - R_{4,1}) = 2.00001659;$$

The $O(h^6)$ approximations are

$$R_{3,3} = R_{3,2} + \frac{1}{15}(R_{3,2} - R_{2,2}) = 1.99857073; \quad R_{4,3} = R_{4,2} + \frac{1}{15}(R_{4,2} - R_{3,2}) = 1.99998313;$$

$$R_{5,3} = R_{5,2} + \frac{1}{15}(R_{5,2} - R_{4,2}) = 1.99999975.$$

The two $O(h^8)$ approximations are

$$R_{4,4} = R_{4,3} + \frac{1}{63}(R_{4,3} - R_{3,3}) = 2.00000555; \quad R_{5,4} = R_{5,3} + \frac{1}{63}(R_{5,3} - R_{4,3}) = 2.00000001,$$

and the final $O(h^{10})$ approximation is

$$R_{5,5} = R_{5,4} + \frac{1}{255}(R_{5,4} - R_{4,4}) = 1.99999999.$$

These results are shown in Table 4.9. ■

Table 4.9

0				
1.57079633	2.09439511			
1.89611890	2.00455976	1.99857073		
1.97423160	2.00026917	1.99998313	2.00000555	
1.99357034	2.00001659	1.99999975	2.00000001	1.99999999

Notice that when generating the approximations for the Composite Trapezoidal rule approximations in Example 1, each consecutive approximation included all the functions evaluations from the previous approximation. That is, $R_{1,1}$ used evaluations at 0 and π, $R_{2,1}$ used these evaluations and added an evaluation at the intermediate point $\pi/2$. Then $R_{3,1}$ used the evaluations of $R_{2,1}$ and added two additional intermediate ones at $\pi/4$ and $3\pi/4$. This pattern continues with $R_{4,1}$ using the same evaluations as $R_{3,1}$ but adding evaluations at the 4 intermediate points $\pi/8$, $3\pi/8$, $5\pi/8$, and $7\pi/8$, and so on.

This evaluation procedure for Composite Trapezoidal rule approximations holds for an integral on any interval $[a, b]$. In general, the Composite Trapezoidal rule denoted $R_{k+1,1}$ uses the same evaluations as $R_{k,1}$ but adds evaluations at the 2^{k-2} intermediate points. Efficient calculation of these approximations can therefore be done in a recursive manner.

To obtain the Composite Trapezoidal rule approximations for $\int_a^b f(x)\,dx$, let $h_k = (b-a)/m_k = (b-a)/2^{k-1}$. Then

$$R_{1,1} = \frac{h_1}{2}[f(a) + f(b)] = \frac{(b-a)}{2}[f(a) + f(b)];$$

and

$$R_{2,1} = \frac{h_2}{2}[f(a) + f(b) + 2f(a + h_2)].$$

By reexpressing this result for $R_{2,1}$ we can incorporate the previously determined approximation $R_{1,1}$

$$R_{2,1} = \frac{(b-a)}{4}\left[f(a) + f(b) + 2f\left(a + \frac{(b-a)}{2}\right)\right] = \frac{1}{2}[R_{1,1} + h_1 f(a + h_2)].$$

In a similar manner we can write

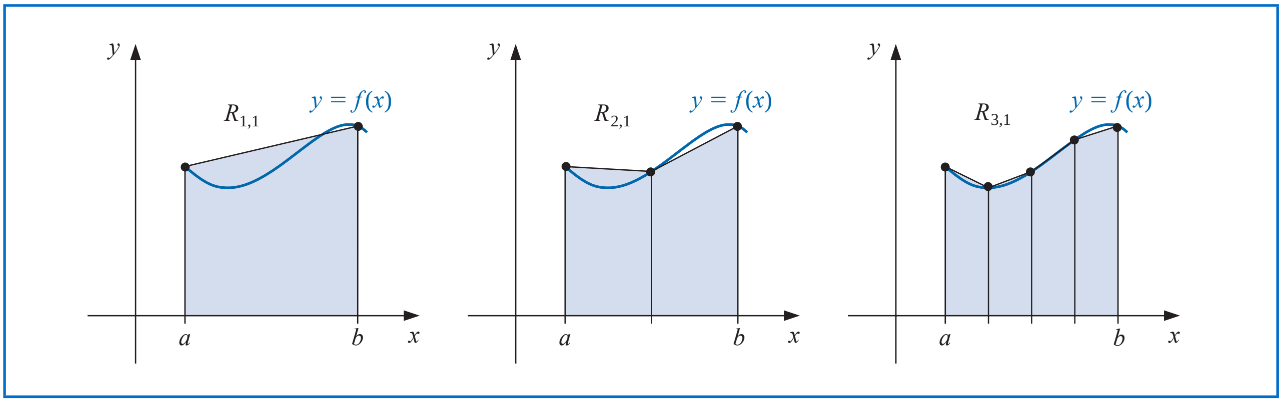
$$R_{3,1} = \frac{1}{2}\{R_{2,1} + h_2[f(a + h_3) + f(a + 3h_3)]\};$$

and, in general (see Figure 4.10 on page 216), we have

$$R_{k,1} = \frac{1}{2}\left[R_{k-1,1} + h_{k-1}\sum_{i=1}^{2^{k-2}} f\left(a + (2i-1)h_k\right)\right], \tag{4.34}$$

for each $k = 2, 3, \ldots, n$. (See Exercises 14 and 15.)

Figure 4.10



Extrapolation then is used to produce $O(h_k^{2j})$ approximations by

$$R_{k,j} = R_{k,j-1} + \frac{1}{4^{j-1}-1}(R_{k,j-1} - R_{k-1,j-1}), \quad \text{for } k = j, j+1, \ldots$$

as shown in Table 4.10.

Table 4.10

k	$O(h_k^2)$	$O(h_k^4)$	$O(h_k^6)$	$O(h_k^8)$		$O(h_k^{2n})$
1	$R_{1,1}$					
2	$R_{2,1}$	$R_{2,2}$				
3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$			
4	$R_{4,1}$	$R_{4,2}$	$R_{4,3}$	$R_{4,4}$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
n	$R_{n,1}$	$R_{n,2}$	$R_{n,3}$	$R_{n,4}$	$\cdots$	$R_{n,n}$

The effective method to construct the Romberg table makes use of the highest order of approximation at each step. That is, it calculates the entries row by row, in the order $R_{1,1}$, $R_{2,1}$, $R_{2,2}$, $R_{3,1}$, $R_{3,2}$, $R_{3,3}$, etc. This also permits an entire new row in the table to be calculated by doing only one additional application of the Composite Trapezoidal rule. It then uses a simple averaging on the previously calculated values to obtain the remaining entries in the row. Remember

- Calculate the Romberg table one complete row at a time.

Example 2 Add an additional extrapolation row to Table 4.10 to approximate $\int_0^{\pi} \sin x\, dx$.

Solution To obtain the additional row we need the trapezoidal approximation

$$R_{6,1} = \frac{1}{2}\left[R_{5,1} + \frac{\pi}{16}\sum_{k=1}^{2^4} \sin\frac{(2k-1)\pi}{32}\right] = 1.99839336.$$

The values in Table 4.10 give

$$R_{6,2} = R_{6,1} + \frac{1}{3}(R_{6,1} - R_{5,1}) = 1.99839336 + \frac{1}{3}(1.99839336 - 1.99357035)$$
$$= 2.00000103;$$

$$R_{6,3} = R_{6,2} + \frac{1}{15}(R_{6,2} - R_{5,2}) = 2.00000103 + \frac{1}{15}(2.00000103 - 2.00001659)$$
$$= 2.00000000;$$

$$R_{6,4} = R_{6,3} + \frac{1}{63}(R_{6,3} - R_{5,3}) = 2.00000000;$$

$$R_{6,5} = R_{6,4} + \frac{1}{255}(R_{6,4} - R_{5,4}) = 2.00000000;$$

and $R_{6,6} = R_{6,5} + \frac{1}{1023}(R_{6,5} - R_{5,5}) = 2.00000000$. The new extrapolation table is shown in Table 4.11. ■

Table 4.11

0					
1.57079633	2.09439511				
1.89611890	2.00455976	1.99857073			
1.97423160	2.00026917	1.99998313	2.00000555		
1.99357034	2.00001659	1.99999975	2.00000001	1.99999999	
1.99839336	2.00000103	2.00000000	2.00000000	2.00000000	2.00000000

Notice that all the extrapolated values except for the first (in the first row of the second column) are more accurate than the best composite trapezoidal approximation (in the last row of the first column). Although there are 21 entries in Table 4.11, only the six in the left column require function evaluations since these are the only entries generated by the Composite Trapezoidal rule; the other entries are obtained by an averaging process. In fact, because of the recurrence relationship of the terms in the left column, the only function evaluations needed are those to compute the final Composite Trapezoidal rule approximation. In general, $R_{k,1}$ requires $1 + 2^{k-1}$ function evaluations, so in this case $1 + 2^5 = 33$ are needed.

Algorithm 4.2 uses the recursive procedure to find the initial Composite Trapezoidal Rule approximations and computes the results in the table row by row.

ALGORITHM **4.2**

Romberg

To approximate the integral $I = \int_a^b f(x)\,dx$, select an integer $n > 0$.

INPUT endpoints a, b; integer n.

OUTPUT an array R. (*Compute R by rows; only the last 2 rows are saved in storage.*)

Step 1 Set $h = b - a$;
$R_{1,1} = \frac{h}{2}(f(a) + f(b))$.

Step 2 OUTPUT $(R_{1,1})$.

Step 3 For $i = 2, \ldots, n$ do Steps 4–8.

Step 4 Set $R_{2,1} = \frac{1}{2}\left[R_{1,1} + h\sum_{k=1}^{2^{i-2}} f(a + (k - 0.5)h)\right].$

(*Approximation from Trapezoidal method.*)

Step 5 For $j = 2, \ldots, i$

set $R_{2,j} = R_{2,j-1} + \frac{R_{2,j-1} - R_{1,j-1}}{4^{j-1} - 1}.$ (*Extrapolation.*)

Step 6 OUTPUT ($R_{2,j}$ for $j = 1, 2, \ldots, i$).

Step 7 Set $h = h/2$.

Step 8 For $j = 1, 2, \ldots, i$ set $R_{1,j} = R_{2,j}$. (*Update row 1 of R.*)

Step 9 STOP. ■

Algorithm 4.2 requires a preset integer n to determine the number of rows to be generated. We could also set an error tolerance for the approximation and generate n, within some upper bound, until consecutive diagonal entries $R_{n-1,n-1}$ and $R_{n,n}$ agree to within the tolerance. To guard against the possibility that two consecutive row elements agree with each other but not with the value of the integral being approximated, it is common to generate approximations until not only $|R_{n-1,n-1} - R_{n,n}|$ is within the tolerance, but also $|R_{n-2,n-2} - R_{n-1,n-1}|$. Although not a universal safeguard, this will ensure that two differently generated sets of approximations agree within the specified tolerance before $R_{n,n}$, is accepted as sufficiently accurate.

Romberg integration can be performed with the *Quadrature* command in the *NumericalAnalysis* subpackage of Maple's *Student* package. For example, after loading the package and defining the function and interval, the command

$$Quadrature(f(x), x = a..b, method = romberg_6, output = information)$$

produces the values shown in Table 4.11 together with the information that 6 applications of the Trapezoidal rule were used and 33 function evaluations were required.

Romberg integration applied to a function f on the interval $[a, b]$ relies on the assumption that the Composite Trapezoidal rule has an error term that can be expressed in the form of Eq. (4.33); that is, we must have $f \in C^{2k+2}[a, b]$ for the kth row to be generated. General-purpose algorithms using Romberg integration include a check at each stage to ensure that this assumption is fulfilled. These methods are known as *cautious Romberg algorithms* and are described in [Joh]. This reference also describes methods for using the Romberg technique as an adaptive procedure, similar to the adaptive Simpson's rule that will be discussed in Section 4.6.

The adjective *cautious* used in the description of a numerical method indicates that a check is incorporated to determine if the continuity hypotheses are likely to be true.

EXERCISE SET 4.5

1. Use Romberg integration to compute $R_{3,3}$ for the following integrals.

a. $\int_1^{1.5} x^2 \ln x \, dx$ **b.** $\int_0^1 x^2 e^{-x} \, dx$

c. $\int_0^{0.35} \frac{2}{x^2 - 4} \, dx$ **d.** $\int_0^{\pi/4} x^2 \sin x \, dx$

e. $\int_0^{\pi/4} e^{3x} \sin 2x\, dx$ **f.** $\int_1^{1.6} \frac{2x}{x^2-4}\, dx$

g. $\int_3^{3.5} \frac{x}{\sqrt{x^2-4}}\, dx$ **h.** $\int_0^{\pi/4} (\cos x)^2\, dx$

2. Use Romberg integration to compute $R_{3,3}$ for the following integrals.

a. $\int_{-1}^{1} (\cos x)^2\, dx$ **b.** $\int_{-0.75}^{0.75} x \ln(x+1)\, dx$

c. $\int_1^4 \left((\sin x)^2 - 2x \sin x + 1\right)\, dx$ **d.** $\int_e^{2e} \frac{1}{x \ln x}\, dx$

3. Calculate $R_{4,4}$ for the integrals in Exercise 1.

4. Calculate $R_{4,4}$ for the integrals in Exercise 2.

5. Use Romberg integration to approximate the integrals in Exercise 1 to within 10^{-6}. Compute the Romberg table until either $|R_{n-1,n-1} - R_{n,n}| < 10^{-6}$, or $n = 10$. Compare your results to the exact values of the integrals.

6. Use Romberg integration to approximate the integrals in Exercise 2 to within 10^{-6}. Compute the Romberg table until either $|R_{n-1,n-1} - R_{n,n}| < 10^{-6}$, or $n = 10$. Compare your results to the exact values of the integrals.

7. Use the following data to approximate $\int_1^5 f(x)\, dx$ as accurately as possible.

x	1	2	3	4	5
$f(x)$	2.4142	2.6734	2.8974	3.0976	3.2804

8. Romberg integration is used to approximate

$$\int_0^1 \frac{x^2}{1+x^3}\, dx.$$

If $R_{11} = 0.250$ and $R_{22} = 0.2315$, what is R_{21}?

9. Romberg integration is used to approximate

$$\int_2^3 f(x)\, dx.$$

If $f(2) = 0.51342$, $f(3) = 0.36788$, $R_{31} = 0.43687$, and $R_{33} = 0.43662$, find $f(2.5)$.

10. Romberg integration for approximating $\int_0^1 f(x)\, dx$ gives $R_{11} = 4$ and $R_{22} = 5$. Find $f(1/2)$.

11. Romberg integration for approximating $\int_a^b f(x)\, dx$ gives $R_{11} = 8$, $R_{22} = 16/3$, and $R_{33} = 208/45$. Find R_{31}.

12. Use Romberg integration to compute the following approximations to

$$\int_0^{48} \sqrt{1 + (\cos x)^2}\, dx.$$

[*Note:* The results in this exercise are most interesting if you are using a device with between seven- and nine-digit arithmetic.]

a. Determine $R_{1,1}$, $R_{2,1}$, $R_{3,1}$, $R_{4,1}$, and $R_{5,1}$, and use these approximations to predict the value of the integral.

b. Determine $R_{2,2}$, $R_{3,3}$, $R_{4,4}$, and $R_{5,5}$, and modify your prediction.

c. Determine $R_{6,1}$, $R_{6,2}$, $R_{6,3}$, $R_{6,4}$, $R_{6,5}$, and $R_{6,6}$, and modify your prediction.

d. Determine $R_{7,7}$, $R_{8,8}$, $R_{9,9}$, and $R_{10,10}$, and make a final prediction.

e. Explain why this integral causes difficulty with Romberg integration and how it can be reformulated to more easily determine an accurate approximation.

13. Show that the approximation obtained from $R_{k,2}$ is the same as that given by the Composite Simpson's rule described in Theorem 4.4 with $h = h_k$.

14. Show that, for any k,

$$\sum_{i=1}^{2^{k-1}-1} f\left(a+\frac{i}{2}h_{k-1}\right) = \sum_{i=1}^{2^{k-2}} f\left(a+\left(i-\frac{1}{2}\right)h_{k-1}\right) + \sum_{i=1}^{2^{k-2}-1} f(a+ih_{k-1}).$$

15. Use the result of Exercise 14 to verify Eq. (4.34); that is, show that for all k,

$$R_{k,1} = \frac{1}{2}\left[R_{k-1,1} + h_{k-1}\sum_{i=1}^{2^{k-2}} f\left(a+\left(i-\frac{1}{2}\right)h_{k-1}\right)\right].$$

16. In Exercise 26 of Section 1.1, a Maclaurin series was integrated to approximate erf(1), where erf(x) is the normal distribution error function defined by

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

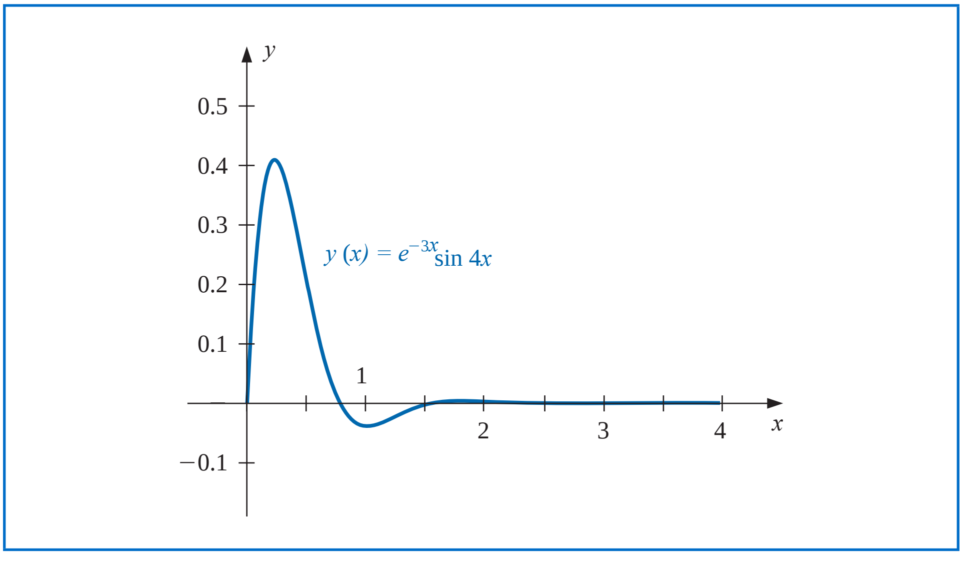
Approximate erf(1) to within 10^{-7}.

4.6 Adaptive Quadrature Methods

The composite formulas are very effective in most situations, but they suffer occasionally because they require the use of equally-spaced nodes. This is inappropriate when integrating a function on an interval that contains both regions with large functional variation and regions with small functional variation.

Illustration The unique solution to the differential equation $y'' + 6y' + 25 = 0$ that additionally satisfies $y(0) = 0$ and $y'(0) = 4$ is $y(x) = e^{-3x}\sin 4x$. Functions of this type are common in mechanical engineering because they describe certain features of spring and shock absorber systems, and in electrical engineering because they are common solutions to elementary circuit problems. The graph of $y(x)$ for x in the interval $[0, 4]$ is shown in Figure 4.11.

Figure 4.11



Suppose that we need the integral of $y(x)$ on $[0, 4]$. The graph indicates that the integral on $[3, 4]$ must be very close to 0, and on $[2, 3]$ would also not be expected to be large. However, on $[0, 2]$ there is significant variation of the function and it is not at all clear what the integral is on this interval. This is an example of a situation where composite integration would be inappropriate. A very low order method could be used on $[2, 4]$, but a higher-order method would be necessary on $[0, 2]$. □

The question we will consider in this section is:

- How can we determine what technique should be applied on various portions of the interval of integration, and how accurate can we expect the final approximation to be?

We will see that under quite reasonable conditions we can answer this question and also determine approximations that satisfy given accuracy requirements.

If the approximation error for an integral on a given interval is to be evenly distributed, a smaller step size is needed for the large-variation regions than for those with less variation. An efficient technique for this type of problem should predict the amount of functional variation and adapt the step size as necessary. These methods are called **Adaptive quadrature methods**. Adaptive methods are particularly popular for inclusion in professional software packages because, in addition to being efficient, they generally provide approximations that are within a given specified tolerance.

In this section we consider an Adaptive quadrature method and see how it can be used to reduce approximation error and also to predict an error estimate for the approximation that does not rely on knowledge of higher derivatives of the function. The method we discuss is based on the Composite Simpson's rule, but the technique is easily modified to use other composite procedures.

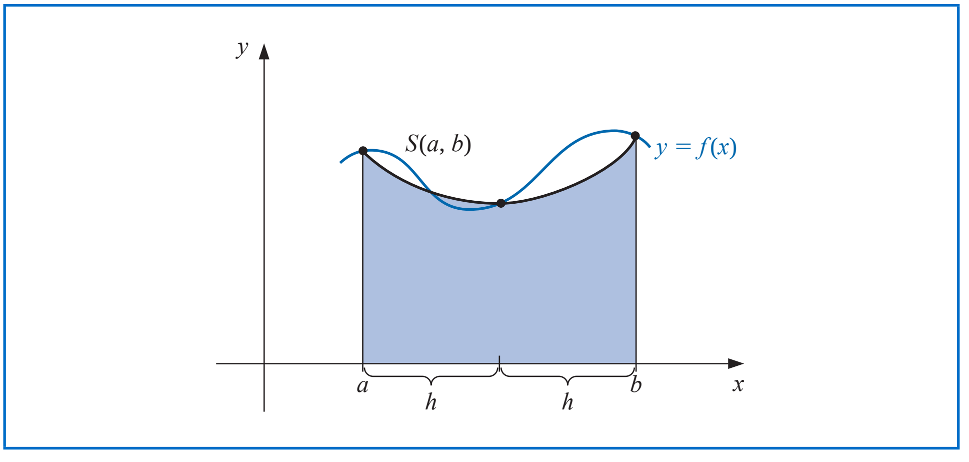
Suppose that we want to approximate $\int_a^b f(x)\,dx$ to within a specified tolerance $\varepsilon > 0$. The first step is to apply Simpson's rule with step size $h = (b - a)/2$. This produces (see Figure 4.12)

$$\int_a^b f(x)\,dx = S(a, b) - \frac{h^5}{90} f^{(4)}(\xi), \quad \text{for some } \xi \text{ in } (a, b), \tag{4.35}$$

where we denote the Simpson's rule approximation on $[a, b]$ by

$$S(a, b) = \frac{h}{3}[f(a) + 4f(a + h) + f(b)].$$

Figure 4.12



The next step is to determine an accuracy approximation that does not require $f^{(4)}(\xi)$. To do this, we apply the Composite Simpson's rule with $n = 4$ and step size $(b-a)/4 = h/2$, giving

$$\int_a^b f(x)\,dx = \frac{h}{6}\left[f(a) + 4f\left(a+\frac{h}{2}\right) + 2f(a+h) + 4f\left(a+\frac{3h}{2}\right) + f(b)\right] - \left(\frac{h}{2}\right)^4 \frac{(b-a)}{180} f^{(4)}(\tilde{\xi}), \tag{4.36}$$

for some $\tilde{\xi}$ in (a, b). To simplify notation, let

$$S\left(a, \frac{a+b}{2}\right) = \frac{h}{6}\left[f(a) + 4f\left(a+\frac{h}{2}\right) + f(a+h)\right]$$

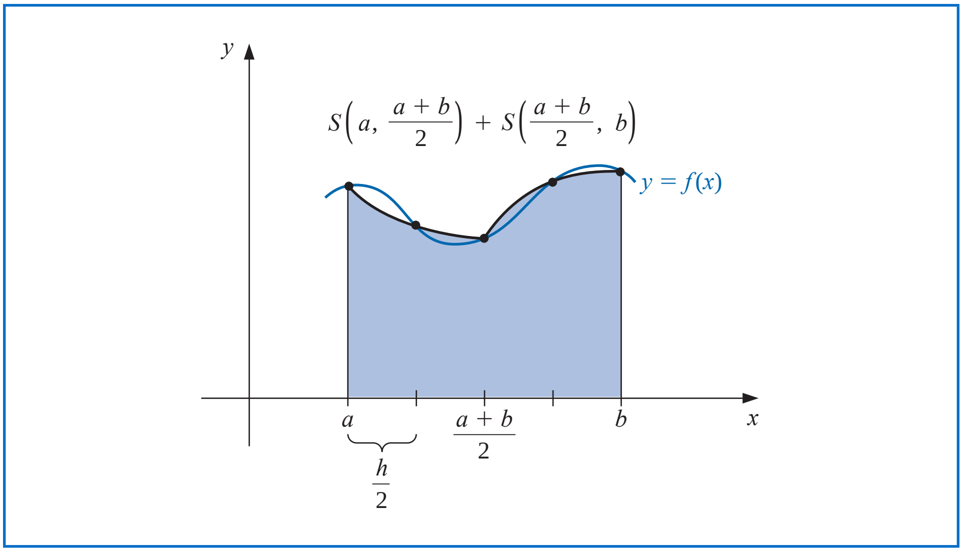
and

$$S\left(\frac{a+b}{2}, b\right) = \frac{h}{6}\left[f(a+h) + 4f\left(a+\frac{3h}{2}\right) + f(b)\right].$$

Then Eq. (4.36) can be rewritten (see Figure 4.13) as

$$\int_a^b f(x)\,dx = S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right) - \frac{1}{16}\left(\frac{h^5}{90}\right) f^{(4)}(\tilde{\xi}). \tag{4.37}$$

Figure 4.13



The error estimation is derived by assuming that $\xi \approx \tilde{\xi}$ or, more precisely, that $f^{(4)}(\xi) \approx f^{(4)}(\tilde{\xi})$, and the success of the technique depends on the accuracy of this assumption. If it is accurate, then equating the integrals in Eqs. (4.35) and (4.37) gives

$$S\left(a, \frac{a+b}{2}\right) + S\left(\frac{a+b}{2}, b\right) - \frac{1}{16}\left(\frac{h^5}{90}\right) f^{(4)}(\xi) \approx S(a,b) - \frac{h^5}{90} f^{(4)}(\xi),$$

so

$$\frac{h^5}{90} f^{(4)}(\xi) \approx \frac{16}{15}\left[S(a,b) - S\left(a, \frac{a+b}{2}\right) - S\left(\frac{a+b}{2}, b\right)\right].$$

Using this estimate in Eq. (4.37) produces the error estimation

$$\left|\int_a^b f(x)\,dx - S\left(a,\frac{a+b}{2}\right) - S\left(\frac{a+b}{2},b\right)\right| \approx \frac{1}{16}\left(\frac{h^5}{90}\right) f^{(4)}(\xi)$$
$$\approx \frac{1}{15}\left|S(a,b) - S\left(a,\frac{a+b}{2}\right) - S\left(\frac{a+b}{2},b\right)\right|.$$

This implies that $S(a,(a+b)/2)+S((a+b)/2,b)$ approximates $\int_a^b f(x)\,dx$ about 15 times better than it agrees with the computed value $S(a,b)$. Thus, if

$$\left|S(a,b) - S\left(a,\frac{a+b}{2}\right) - S\left(\frac{a+b}{2},b\right)\right| < 15\varepsilon, \tag{4.38}$$

we expect to have

$$\left|\int_a^b f(x)\,dx - S\left(a,\frac{a+b}{2}\right) - S\left(\frac{a+b}{2},b\right)\right| < \varepsilon, \tag{4.39}$$

and

$$S\left(a,\frac{a+b}{2}\right) + S\left(\frac{a+b}{2},b\right)$$

is assumed to be a sufficiently accurate approximation to $\int_a^b f(x)\,dx$.

Example 1 Check the accuracy of the error estimate given in (4.38) and (4.39) when applied to the integral

$$\int_0^{\pi/2} \sin x\,dx = 1.$$

by comparing

$$\frac{1}{15}\left|S\left(0,\frac{\pi}{2}\right) - S\left(0,\frac{\pi}{4}\right) - S\left(\frac{\pi}{4},\frac{\pi}{2}\right)\right| \quad \text{to} \quad \left|\int_0^{\pi/2} \sin x\,dx - S\left(0,\frac{\pi}{4}\right) - S\left(\frac{\pi}{4},\frac{\pi}{2}\right)\right|.$$

Solution We have

$$S\left(0,\frac{\pi}{2}\right) = \frac{\pi/4}{3}\left[\sin 0 + 4\sin\frac{\pi}{4} + \sin\frac{\pi}{2}\right] = \frac{\pi}{12}(2\sqrt{2}+1) = 1.002279878$$

and

$$S\left(0,\frac{\pi}{4}\right) + S\left(\frac{\pi}{4},\frac{\pi}{2}\right) = \frac{\pi/8}{3}\left[\sin 0 + 4\sin\frac{\pi}{8} + 2\sin\frac{\pi}{4} + 4\sin\frac{3\pi}{8} + \sin\frac{\pi}{2}\right]$$
$$= 1.000134585.$$

So

$$\left|S\left(0,\frac{\pi}{2}\right) - S\left(0,\frac{\pi}{4}\right) - S\left(\frac{\pi}{4},\frac{\pi}{2}\right)\right| = |1.002279878 - 1.000134585| = 0.002145293.$$

The estimate for the error obtained when using $S(a,(a+b)) + S((a+b),b)$ to approximate $\int_a^b f(x)\,dx$ is consequently

$$\frac{1}{15}\left|S\left(0,\frac{\pi}{2}\right) - S\left(0,\frac{\pi}{4}\right) - S\left(\frac{\pi}{4},\frac{\pi}{2}\right)\right| = 0.000143020,$$

which closely approximates the actual error

$$\left| \int_0^{\pi/2} \sin x \, dx - 1.000134585 \right| = 0.000134585,$$

even though $D_x^4 \sin x = \sin x$ varies significantly in the interval $(0, \pi/2)$. ▪

When the approximations in (4.38) differ by more than 15ε, we can apply the Simpson's rule technique individually to the subintervals $[a, (a+b)/2]$ and $[(a+b)/2, b]$. Then we use the error estimation procedure to determine if the approximation to the integral on each subinterval is within a tolerance of $\varepsilon/2$. If so, we sum the approximations to produce an approximation to $\int_a^b f(x)\, dx$ within the tolerance ε.

If the approximation on one of the subintervals fails to be within the tolerance $\varepsilon/2$, then that subinterval is itself subdivided, and the procedure is reapplied to the two subintervals to determine if the approximation on each subinterval is accurate to within $\varepsilon/4$. This halving procedure is continued until each portion is within the required tolerance.

Problems can be constructed for which this tolerance will never be met, but the technique is usually successful, because each subdivision typically increases the accuracy of the approximation by a factor of 16 while requiring an increased accuracy factor of only 2.

Algorithm 4.3 details this Adaptive quadrature procedure for Simpson's rule, although some technical difficulties arise that require the implementation to differ slightly from the preceding discussion. For example, in Step 1 the tolerance has been set at 10ε rather than the 15ε figure in Inequality (4.38). This bound is chosen conservatively to compensate for error in the assumption $f^{(4)}(\xi) \approx f^{(4)}(\tilde{\xi})$. In problems where $f^{(4)}$ is known to be widely varying, this bound should be decreased even further.

It is a good idea to include a margin of safety when it is impossible to verify accuracy assumptions.

The procedure listed in the algorithm first approximates the integral on the leftmost subinterval in a subdivision. This requires the efficient storing and recalling of previously computed functional evaluations for the nodes in the right half subintervals. Steps 3, 4, and 5 contain a stacking procedure with an indicator to keep track of the data that will be required for calculating the approximation on the subinterval immediately adjacent and to the right of the subinterval on which the approximation is being generated. The method is easier to implement using a recursive programming language.

ALGORITHM 4.3

Adaptive Quadrature

To approximate the integral $I = \int_a^b f(x)\, dx$ to within a given tolerance:

INPUT endpoints a, b; tolerance TOL; limit N to number of levels.

OUTPUT approximation APP or message that N is exceeded.

Step 1 Set $APP = 0$;
$i = 1$;
$TOL_i = 10\, TOL$;
$a_i = a$;
$h_i = (b-a)/2$;
$FA_i = f(a)$;
$FC_i = f(a+h_i)$;
$FB_i = f(b)$;
$S_i = h_i(FA_i + 4FC_i + FB_i)/3$; (*Approximation from Simpson's method for entire interval.*)
$L_i = 1$.

Step 2 While $i > 0$ do Steps 3–5.

Step 3 Set $FD = f(a_i + h_i/2)$;
$FE = f(a_i + 3h_i/2)$;
$S1 = h_i(FA_i + 4FD + FC_i)/6$; (*Approximations from Simpson's method for halves of subintervals.*)
$S2 = h_i(FC_i + 4FE + FB_i)/6$;
$v_1 = a_i$; (*Save data at this level.*)
$v_2 = FA_i$;
$v_3 = FC_i$;
$v_4 = FB_i$;
$v_5 = h_i$;
$v_6 = TOL_i$;
$v_7 = S_i$;
$v_8 = L_i$.

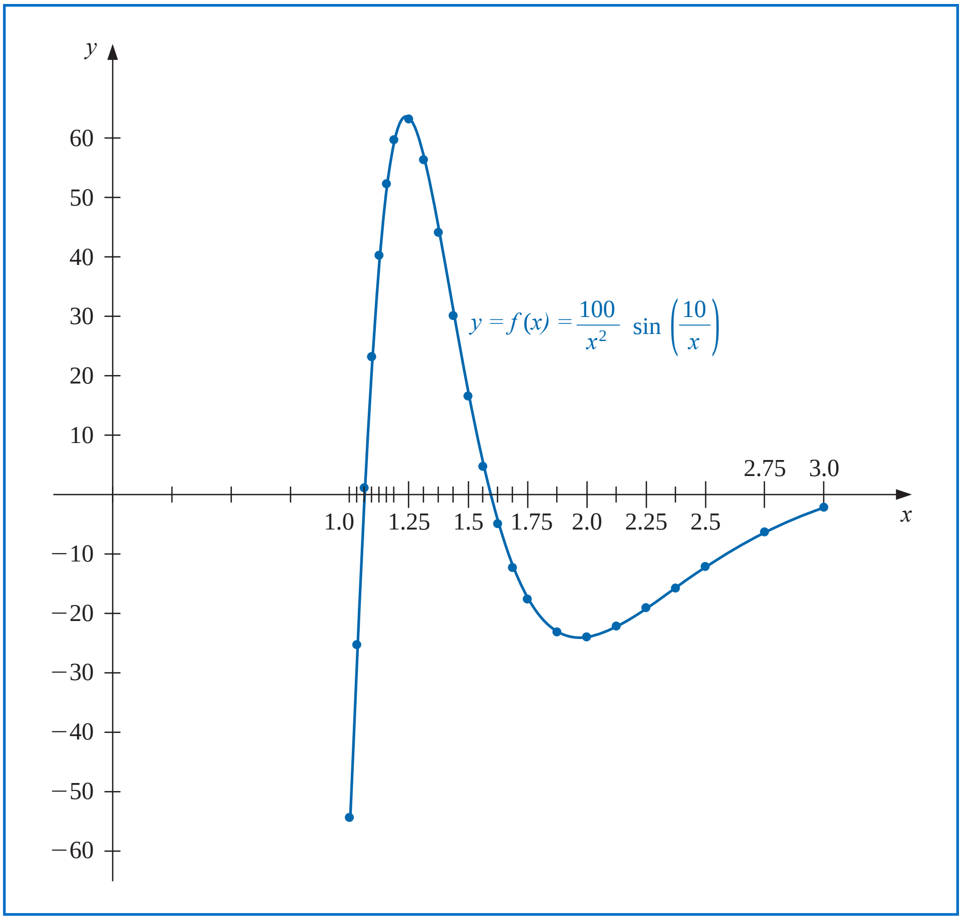
Step 4 Set $i = i - 1$. (*Delete the level.*)

Step 5 If $|S1 + S2 - v_7| < v_6$
then set $APP = APP + (S1 + S2)$
else
if ($v_8 \geq N$)
then
OUTPUT ('LEVEL EXCEEDED'); (*Procedure fails.*)
STOP.
else (*Add one level.*)
set $i = i + 1$; (*Data for right half subinterval.*)
$a_i = v_1 + v_5$;
$FA_i = v_3$;
$FC_i = FE$;
$FB_i = v_4$;
$h_i = v_5/2$;
$TOL_i = v_6/2$;
$S_i = S2$;
$L_i = v_8 + 1$;
set $i = i + 1$; (*Data for left half subinterval.*)
$a_i = v_1$;
$FA_i = v_2$;
$FC_i = FD$;
$FB_i = v_3$;
$h_i = h_{i-1}$;
$TOL_i = TOL_{i-1}$;
$S_i = S1$;
$L_i = L_{i-1}$.

Step 6 OUTPUT (APP); (*APP approximates I to within TOL.*)
STOP. ■

Illustration The graph of the function $f(x) = (100/x^2)\sin(10/x)$ for x in $[1, 3]$ is shown in Figure 4.14. Using the Adaptive Quadrature Algorithm 4.3 with tolerance 10^{-4} to approximate $\int_1^3 f(x)\,dx$ produces -1.426014, a result that is accurate to within 1.1×10^{-5}. The approximation required that Simpson's rule with $n = 4$ be performed on the 23 subintervals whose endpoints are shown on the horizontal axis in Figure 4.14. The total number of functional evaluations required for this approximation is 93.

Figure 4.14



The largest value of h for which the standard Composite Simpson's rule gives 10^{-4} accuracy is $h = 1/88$. This application requires 177 function evaluations, nearly twice as many as Adaptive quadrature. □

Adaptive quadrature can be performed with the *Quadrature* command in the *Numerical-Analysis* subpackage of Maple's *Student* package. In this situation the option *adaptive* = *true* is used. For example, to produce the values in the Illustration we first load the package and define the function and interval with

$$f := x \rightarrow \frac{100}{x^2} \cdot \sin\left(\frac{10}{x}\right);\ a := 1.0;\ b := 3.0$$

Then give the *NumericalAnalysis* command

$Quadrature(f(x), x = a..b, adaptive = true, method = [simpson, 10^{-4}], output = information)$

This produces the approximation -1.42601481 and a table that lists all the intervals on which Simpson's rule was employed and whether the appropriate tolerance was satisfied (indicated by the word `PASS`) or was not satisfied (indicated by the word `fail`). It also gives what Maple thinks is the correct value of the integral to the decimal places listed, in this case -1.42602476. Then it gives the absolute and relative errors, 9.946×10^{-6} and 6.975×10^{-4}, respectively, assuming that its correct value is accurate.

EXERCISE SET 4.6

1. Compute the Simpson's rule approximations $S(a, b)$, $S(a, (a+b)/2)$, and $S((a+b)/2, b)$ for the following integrals, and verify the estimate given in the approximation formula.

a. $\displaystyle\int_1^{1.5} x^2 \ln x \, dx$ **b.** $\displaystyle\int_0^1 x^2 e^{-x} \, dx$

c. $\displaystyle\int_0^{0.35} \frac{2}{x^2-4} \, dx$ **d.** $\displaystyle\int_0^{\pi/4} x^2 \sin x \, dx$

e. $\displaystyle\int_0^{\pi/4} e^{3x} \sin 2x \, dx$ **f.** $\displaystyle\int_1^{1.6} \frac{2x}{x^2-4} \, dx$

g. $\displaystyle\int_3^{3.5} \frac{x}{\sqrt{x^2-4}} \, dx$ **h.** $\displaystyle\int_0^{\pi/4} (\cos x)^2 \, dx$

2. Use Adaptive quadrature to find approximations to within 10^{-3} for the integrals in Exercise 1. Do not use a computer program to generate these results.

3. Use Adaptive quadrature to approximate the following integrals to within 10^{-5}.

a. $\displaystyle\int_1^3 e^{2x} \sin 3x \, dx$ **b.** $\displaystyle\int_1^3 e^{3x} \sin 2x \, dx$

c. $\displaystyle\int_0^5 \left(2x\cos(2x) - (x-2)^2\right) dx$ **d.** $\displaystyle\int_0^5 \left(4x\cos(2x) - (x-2)^2\right) dx$

4. Use Adaptive quadrature to approximate the following integrals to within 10^{-5}.

a. $\displaystyle\int_0^{\pi} (\sin x + \cos x) \, dx$ **b.** $\displaystyle\int_1^2 (x + \sin 4x) \, dx$

c. $\displaystyle\int_{-1}^1 x \sin 4x \, dx$ **d.** $\displaystyle\int_0^{\pi/2} (6\cos 4x + 4\sin 6x)e^x \, dx$

5. Use Simpson's Composite rule with $n = 4, 6, 8, \ldots$, until successive approximations to the following integrals agree to within 10^{-6}. Determine the number of nodes required. Use the Adaptive Quadrature Algorithm to approximate the integral to within 10^{-6}, and count the number of nodes. Did Adaptive quadrature produce any improvement?

a. $\displaystyle\int_0^{\pi} x \cos x^2 \, dx$ **b.** $\displaystyle\int_0^{\pi} x \sin x^2 \, dx$

c. $\displaystyle\int_0^{\pi} x^2 \cos x \, dx$ **d.** $\displaystyle\int_0^{\pi} x^2 \sin x \, dx$

6. Sketch the graphs of $\sin(1/x)$ and $\cos(1/x)$ on $[0.1, 2]$. Use Adaptive quadrature to approximate the following integrals to within 10^{-3}.

a. $\displaystyle\int_{0.1}^2 \sin\frac{1}{x} \, dx$ **b.** $\displaystyle\int_{0.1}^2 \cos\frac{1}{x} \, dx$

7. The differential equation

$$mu''(t) + ku(t) = F_0 \cos \omega t$$

describes a spring-mass system with mass m, spring constant k, and no applied damping. The term $F_0 \cos \omega t$ describes a periodic external force applied to the system. The solution to the equation when the system is initially at rest ($u'(0) = u(0) = 0$) is

$$u(t) = \frac{F_0}{m(\omega_0^2 - \omega^2)} (\cos \omega t - \cos \omega_0 t), \quad \text{where} \quad \omega_0 = \sqrt{\frac{k}{m}} \neq \omega.$$

Sketch the graph of u when $m = 1$, $k = 9$, $F_0 = 1$, $\omega = 2$, and $t \in [0, 2\pi]$. Approximate $\int_0^{2\pi} u(t)\, dt$ to within 10^{-4}.

8. If the term $cu'(t)$ is added to the left side of the motion equation in Exercise 7, the resulting differential equation describes a spring-mass system that is damped with damping constant $c \neq 0$. The solution to this equation when the system is initially at rest is

$$u(t) = c_1 e^{r_1 t} + c_2 e^{r_2 t} + \frac{F_0}{c^2\omega^2 + m^2(\omega_0^2 - \omega^2)^2} \left(c\omega \sin \omega t + m\left(\omega_0^2 - \omega^2\right) \cos \omega t\right),$$

where

$$r_1 = \frac{-c + \sqrt{c^2 - 4\omega_0^2 m^2}}{2m} \quad \text{and} \quad r_2 = \frac{-c - \sqrt{c^2 - 4\omega_0^2 m^2}}{2m}.$$

a. Let $m = 1$, $k = 9$, $F_0 = 1$, $c = 10$, and $\omega = 2$. Find the values of c_1 and c_2 so that $u(0) = u'(0) = 0$.

b. Sketch the graph of $u(t)$ for $t \in [0, 2\pi]$ and approximate $\int_0^{2\pi} u(t)\, dt$ to within 10^{-4}.

9. Let $T(a, b)$ and $T(a, \frac{a+b}{2}) + T(\frac{a+b}{2}, b)$ be the single and double applications of the Trapezoidal rule to $\int_a^b f(x)\, dx$. Derive the relationship between

$$\left| T(a, b) - T\left(a, \frac{a+b}{2}\right) - T\left(\frac{a+b}{2}, b\right) \right|$$

and

$$\left| \int_a^b f(x)\, dx - T\left(a, \frac{a+b}{2}\right) - T\left(\frac{a+b}{2}, b\right) \right|.$$

10. The study of light diffraction at a rectangular aperture involves the Fresnel integrals

$$c(t) = \int_0^t \cos \frac{\pi}{2}\omega^2\, d\omega \quad \text{and} \quad s(t) = \int_0^t \sin \frac{\pi}{2}\omega^2\, d\omega.$$

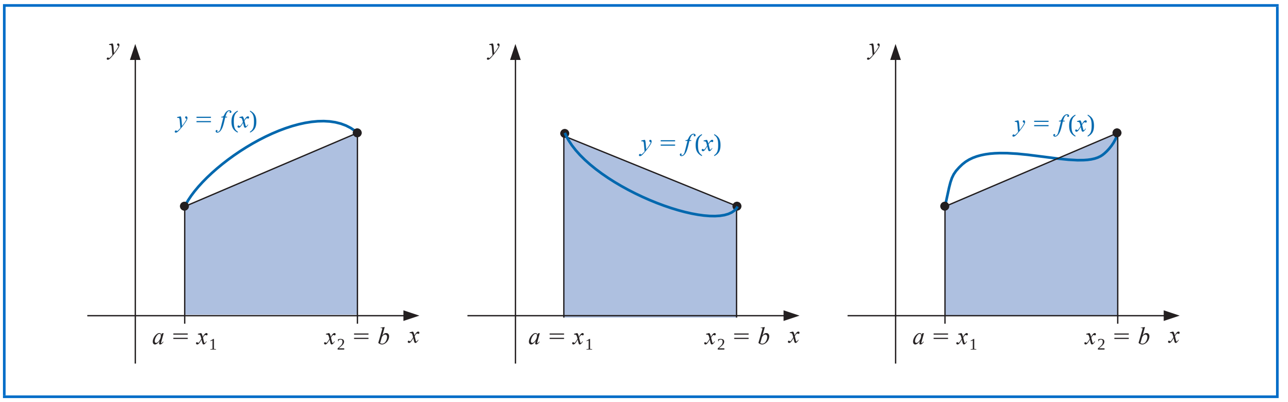
Construct a table of values for $c(t)$ and $s(t)$ that is accurate to within 10^{-4} for values of $t = 0.1, 0.2, \ldots, 1.0$.

4.7 Gaussian Quadrature

The Newton-Cotes formulas in Section 4.3 were derived by integrating interpolating polynomials. The error term in the interpolating polynomial of degree n involves the $(n + 1)$st derivative of the function being approximated, so a Newton-Cotes formula is exact when approximating the integral of any polynomial of degree less than or equal to n.

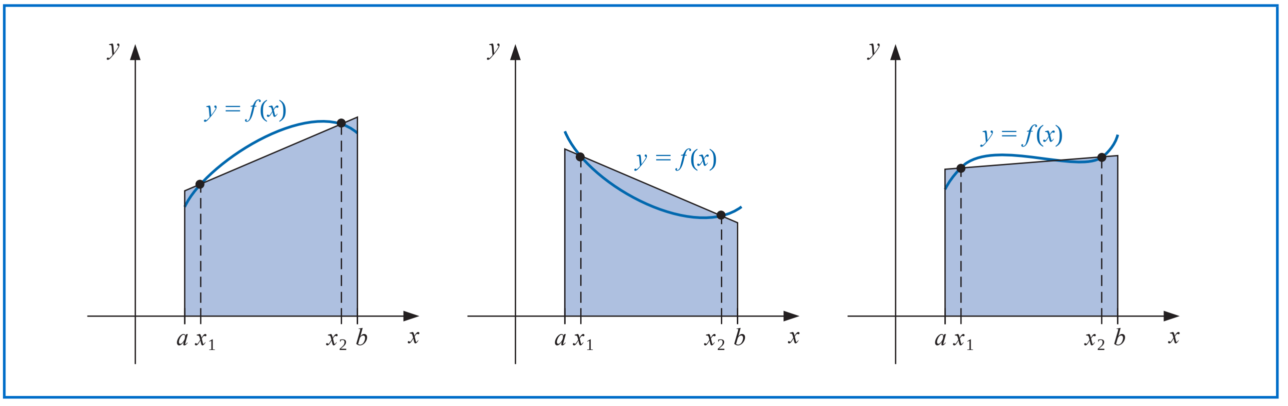
All the Newton-Cotes formulas use values of the function at equally-spaced points. This restriction is convenient when the formulas are combined to form the composite rules we considered in Section 4.4, but it can significantly decrease the accuracy of the approximation. Consider, for example, the Trapezoidal rule applied to determine the integrals of the functions whose graphs are shown in Figure 4.15.

Figure 4.15



The Trapezoidal rule approximates the integral of the function by integrating the linear function that joins the endpoints of the graph of the function. But this is not likely the best line for approximating the integral. Lines such as those shown in Figure 4.16 would likely give much better approximations in most cases.

Figure 4.16



Gauss demonstrated his method of efficient numerical integration in a paper that was presented to the Göttingen Society in 1814. He let the nodes as well as the coefficients of the function evaluations be parameters in the summation formula and found the optimal placement of the nodes. Goldstine [Golds], pp 224–232, has an interesting description of his development.

Gaussian quadrature chooses the points for evaluation in an optimal, rather than equally-spaced, way. The nodes $x_1, x_2, \ldots, x_n$ in the interval $[a, b]$ and coefficients $c_1, c_2, \ldots, c_n$, are chosen to minimize the expected error obtained in the approximation

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} c_i f(x_i).$$

To measure this accuracy, we assume that the best choice of these values produces the exact result for the largest class of polynomials, that is, the choice that gives the greatest degree of precision.

The coefficients $c_1, c_2, \ldots, c_n$ in the approximation formula are arbitrary, and the nodes $x_1, x_2, \ldots, x_n$ are restricted only by the fact that they must lie in $[a, b]$, the interval of integration. This gives us $2n$ parameters to choose. If the coefficients of a polynomial are considered parameters, the class of polynomials of degree at most $2n-1$ also contains $2n$ parameters. This, then, is the largest class of polynomials for which it is reasonable to expect a formula to be exact. With the proper choice of the values and constants, exactness on this set can be obtained.

To illustrate the procedure for choosing the appropriate parameters, we will show how to select the coefficients and nodes when $n = 2$ and the interval of integration is $[-1, 1]$. We will then discuss the more general situation for an arbitrary choice of nodes and coefficients and show how the technique is modified when integrating over an arbitrary interval.

Suppose we want to determine c_1, c_2, x_1, and x_2 so that the integration formula

$$\int_{-1}^{1} f(x)\,dx \approx c_1 f(x_1) + c_2 f(x_2)$$

gives the exact result whenever $f(x)$ is a polynomial of degree $2(2)-1 = 3$ or less, that is, when

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3,$$

for some collection of constants, a_0, a_1, a_2, and a_3. Because

$$\int (a_0 + a_1 x + a_2 x^2 + a_3 x^3)\,dx = a_0 \int 1\,dx + a_1 \int x\,dx + a_2 \int x^2\,dx + a_3 \int x^3\,dx,$$

this is equivalent to showing that the formula gives exact results when $f(x)$ is 1, x, x^2, and x^3. Hence, we need c_1, c_2, x_1, and x_2, so that

$$c_1 \cdot 1 + c_2 \cdot 1 = \int_{-1}^{1} 1\,dx = 2, \qquad c_1 \cdot x_1 + c_2 \cdot x_2 = \int_{-1}^{1} x\,dx = 0,$$

$$c_1 \cdot x_1^2 + c_2 \cdot x_2^2 = \int_{-1}^{1} x^2\,dx = \frac{2}{3}, \quad \text{and} \quad c_1 \cdot x_1^3 + c_2 \cdot x_2^3 = \int_{-1}^{1} x^3\,dx = 0.$$

A little algebra shows that this system of equations has the unique solution

$$c_1 = 1, \quad c_2 = 1, \quad x_1 = -\frac{\sqrt{3}}{3}, \quad \text{and} \quad x_2 = \frac{\sqrt{3}}{3},$$

which gives the approximation formula

$$\int_{-1}^{1} f(x)\,dx \approx f\left(\frac{-\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right). \tag{4.40}$$

This formula has degree of precision 3, that is, it produces the exact result for every polynomial of degree 3 or less.

Legendre Polynomials

The technique we have described could be used to determine the nodes and coefficients for formulas that give exact results for higher-degree polynomials, but an alternative method obtains them more easily. In Sections 8.2 and 8.3 we will consider various collections of orthogonal polynomials, functions that have the property that a particular definite integral of the product of any two of them is 0. The set that is relevant to our problem is the Legendre polynomials, a collection $\{P_0(x), P_1(x), \ldots, P_n(x), \ldots,\}$ with properties:

(1) For each n, $P_n(x)$ is a monic polynomial of degree n.

(2) $\int_{-1}^{1} P(x)P_n(x)\,dx = 0$ whenever $P(x)$ is a polynomial of degree less than n.

Recall that *monic* polynomials have leading coefficient 1.

The first few Legendre polynomials are

$$P_0(x) = 1, \quad P_1(x) = x, \quad P_2(x) = x^2 - \frac{1}{3},$$

$$P_3(x) = x^3 - \frac{3}{5}x, \quad \text{and} \quad P_4(x) = x^4 - \frac{6}{7}x^2 + \frac{3}{35}.$$

Adrien-Marie Legendre (1752–1833) introduced this set of polynomials in 1785. He had numerous priority disputes with Gauss, primarily due to Gauss' failure to publish many of his original results until long after he had discovered them.

The roots of these polynomials are distinct, lie in the interval $(-1, 1)$, have a symmetry with respect to the origin, and, most importantly, are the correct choice for determining the parameters that give us the nodes and coefficients for our quadrature method.

The nodes $x_1, x_2, \ldots, x_n$ needed to produce an integral approximation formula that gives exact results for any polynomial of degree less than $2n$ are the roots of the nth-degree Legendre polynomial. This is established by the following result.

Theorem 4.7 Suppose that $x_1, x_2, \ldots, x_n$ are the roots of the nth Legendre polynomial $P_n(x)$ and that for each $i = 1, 2, \ldots, n$, the numbers c_i are defined by

$$c_i = \int_{-1}^{1} \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}\,dx.$$

If $P(x)$ is any polynomial of degree less than $2n$, then

$$\int_{-1}^{1} P(x)\,dx = \sum_{i=1}^{n} c_i P(x_i).$$ ■

Proof Let us first consider the situation for a polynomial $P(x)$ of degree less than n. Rewrite $P(x)$ in terms of $(n-1)$st Lagrange coefficient polynomials with nodes at the roots of the nth Legendre polynomial $P_n(x)$. The error term for this representation involves the nth derivative of $P(x)$. Since $P(x)$ is of degree less than n, the nth derivative of $P(x)$ is 0, and this representation of is exact. So

$$P(x) = \sum_{i=1}^{n} P(x_i)L_i(x) = \sum_{i=1}^{n} \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} P(x_i)$$

and

$$\int_{-1}^{1} P(x)\,dx = \int_{-1}^{1} \left[\sum_{i=1}^{n} \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} P(x_i) \right] dx$$

$$= \sum_{i=1}^{n} \left[\int_{-1}^{1} \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}\,dx \right] P(x_i) = \sum_{i=1}^{n} c_i P(x_i).$$

Hence the result is true for polynomials of degree less than n.

Now consider a polynomial $P(x)$ of degree at least n but less than $2n$. Divide $P(x)$ by the nth Legendre polynomial $P_n(x)$. This gives two polynomials $Q(x)$ and $R(x)$, each of degree less than n, with

$$P(x) = Q(x)P_n(x) + R(x).$$

Note that x_i is a root of $P_n(x)$ for each $i = 1, 2, \ldots, n$, so we have

$$P(x_i) = Q(x_i)P_n(x_i) + R(x_i) = R(x_i).$$

We now invoke the unique power of the Legendre polynomials. First, the degree of the polynomial $Q(x)$ is less than n, so (by Legendre property **(2)**),

$$\int_{-1}^{1} Q(x)P_n(x)\,dx = 0.$$

Then, since $R(x)$ is a polynomial of degree less than n, the opening argument implies that

$$\int_{-1}^{1} R(x)\,dx = \sum_{i=1}^{n} c_i R(x_i).$$

Putting these facts together verifies that the formula is exact for the polynomial $P(x)$:

$$\int_{-1}^{1} P(x)\,dx = \int_{-1}^{1} [Q(x)P_n(x) + R(x)]\,dx = \int_{-1}^{1} R(x)\,dx = \sum_{i=1}^{n} c_i R(x_i) = \sum_{i=1}^{n} c_i P(x_i).$$

■ ■ ■

The constants c_i needed for the quadrature rule can be generated from the equation in Theorem 4.7, but both these constants and the roots of the Legendre polynomials are extensively tabulated. Table 4.12 lists these values for $n = 2, 3, 4$, and 5.

Table 4.12

n	Roots $r_{n,i}$	Coefficients $c_{n,i}$
2	0.5773502692	1.0000000000
	−0.5773502692	1.0000000000
3	0.7745966692	0.5555555556
	0.0000000000	0.8888888889
	−0.7745966692	0.5555555556
4	0.8611363116	0.3478548451
	0.3399810436	0.6521451549
	−0.3399810436	0.6521451549
	−0.8611363116	0.3478548451
5	0.9061798459	0.2369268850
	0.5384693101	0.4786286705
	0.0000000000	0.5688888889
	−0.5384693101	0.4786286705
	−0.9061798459	0.2369268850

Example 1 Approximate $\int_{-1}^{1} e^x \cos x\,dx$ using Gaussian quadrature with $n = 3$.

Solution The entries in Table 4.12 give us

$$\int_{-1}^{1} e^x \cos x\,dx \approx 0.\overline{5}e^{0.774596692} \cos 0.774596692$$
$$+ 0.\overline{8} \cos 0 + 0.\overline{5}e^{-0.774596692} \cos(-0.774596692)$$
$$= 1.9333904.$$

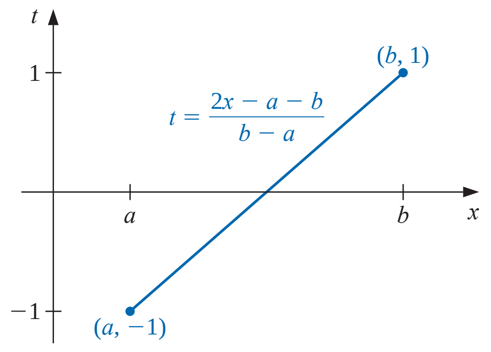
Integration by parts can be used to show that the true value of the integral is 1.9334214, so the absolute error is less than 3.2×10^{-5}. ■

Gaussian Quadrature on Arbitrary Intervals

An integral $\int_a^b f(x)\,dx$ over an arbitrary $[a, b]$ can be transformed into an integral over $[-1, 1]$ by using the change of variables (see Figure 4.17):

$$t = \frac{2x - a - b}{b - a} \iff x = \frac{1}{2}[(b - a)t + a + b].$$

Figure 4.17



This permits Gaussian quadrature to be applied to any interval $[a, b]$, because

$$\int_a^b f(x)\,dx = \int_{-1}^{1} f\left(\frac{(b-a)t + (b+a)}{2}\right)\frac{(b-a)}{2}\,dt. \tag{4.41}$$

Example 2 Consider the integral $\displaystyle\int_1^3 x^6 - x^2 \sin(2x)\,dx = 317.3442466.$

(a) Compare the results for the closed Newton-Cotes formula with $n = 1$, the open Newton-Cotes formula with $n = 1$, and Gaussian Quadrature when $n = 2$.

(b) Compare the results for the closed Newton-Cotes formula with $n = 2$, the open Newton-Cotes formula with $n = 2$, and Gaussian Quadrature when $n = 3$.

Solution **(a)** Each of the formulas in this part requires 2 evaluations of the function $f(x) = x^6 - x^2\sin(2x)$. The Newton-Cotes approximations are

$$\textbf{Closed } \boldsymbol{n = 1}: \quad \frac{2}{2}[f(1) + f(3)] = 731.6054420;$$

$$\textbf{Open } \boldsymbol{n = 1}: \quad \frac{3(2/3)}{2}[f(5/3) + f(7/3)] = 188.7856682.$$

Gaussian quadrature applied to this problem requires that the integral first be transformed into a problem whose interval of integration is $[-1, 1]$. Using Eq. (4.41) gives

$$\int_1^3 x^6 - x^2\sin(2x)\,dx = \int_{-1}^{1} (t+2)^6 - (t+2)^2\sin(2(t+2))\,dt.$$

Gaussian quadrature with $n = 2$ then gives

$$\int_1^3 x^6 - x^2\sin(2x)\,dx \approx f(-0.5773502692 + 2) + f(0.5773502692 + 2) = 306.8199344;$$

(b) Each of the formulas in this part requires 3 function evaluations. The Newton-Cotes approximations are

$$\textbf{Closed } n = 2: \quad \frac{(1)}{3}[f(1) + 4f(2) + f(3)] = 333.2380940;$$

$$\textbf{Open } n = 2: \quad \frac{4(1/2)}{3}[2f(1.5) - f(2) + 2f(2.5)] = 303.5912023.$$

Gaussian quadrature with $n = 3$, once the transformation has been done, gives

$$\int_1^3 x^6 - x^2 \sin(2x)\, dx \approx 0.\overline{5}f(-0.7745966692 + 2) + 0.\overline{8}f(2)$$

$$+ 0.\overline{5}f(0.7745966692 + 2) = 317.2641516.$$

The Gaussian quadrature results are clearly superior in each instance. ■

Maple has Composite Gaussian Quadrature in the *NumericalAnalysis* subpackage of Maple's *Student* package. The default for the number of partitions in the command is 10, so the results in Example 2 would be found for $n = 2$ with

$f := x^6 - x^2 \sin(2x); a := 1; b := 3:$
$Quadrature(f(x), x = a..b, method = gaussian[2], partition = 1, output = information)$

which returns the approximation, what Maple assumes is the exact value of the integral, the absolute, and relative errors in the approximations, and the number of function evaluations.

The result when $n = 3$ is, of course, obtained by replacing the statement *method* = *gaussian*[2] with *method* = *gaussian*[3].

EXERCISE SET 4.7

1. Approximate the following integrals using Gaussian quadrature with $n = 2$, and compare your results to the exact values of the integrals.

   a. $\int_1^{1.5} x^2 \ln x\, dx$ **b.** $\int_0^1 x^2 e^{-x}\, dx$

   c. $\int_0^{0.35} \frac{2}{x^2 - 4}\, dx$ **d.** $\int_0^{\pi/4} x^2 \sin x\, dx$

   e. $\int_0^{\pi/4} e^{3x} \sin 2x\, dx$ **f.** $\int_1^{1.6} \frac{2x}{x^2 - 4}\, dx$

   g. $\int_3^{3.5} \frac{x}{\sqrt{x^2 - 4}}\, dx$ **h.** $\int_0^{\pi/4} (\cos x)^2\, dx$

2. Repeat Exercise 1 with $n = 3$.
3. Repeat Exercise 1 with $n = 4$.
4. Repeat Exercise 1 with $n = 5$.
5. Determine constants a, b, c, and d that will produce a quadrature formula

   $$\int_{-1}^1 f(x)\, dx = af(-1) + bf(1) + cf'(-1) + df'(1)$$

   that has degree of precision 3.
6. Determine constants a, b, c, and d that will produce a quadrature formula

   $$\int_{-1}^1 f(x)\, dx = af(-1) + bf(0) + cf(1) + df'(-1) + ef'(1)$$

   that has degree of precision 4.

7. Verify the entries for the values of $n = 2$ and 3 in Table 4.12 on page 232 by finding the roots of the respective Legendre polynomials, and use the equations preceding this table to find the coefficients associated with the values.

8. Show that the formula $Q(P) = \sum_{i=1}^{n} c_i P(x_i)$ cannot have degree of precision greater than $2n - 1$, regardless of the choice of $c_1, \ldots, c_n$ and $x_1, \ldots, x_n$. [*Hint:* Construct a polynomial that has a double root at each of the x_i's.]

9. Apply Maple's Composite Gaussian Quadrature routine to approximate $\int_{-1}^{1} x^2 e^x\, dx$ in the following manner.

a. Use Gaussian Quadrature with $n = 8$ on the single interval $[-1, 1]$.

b. Use Gaussian Quadrature with $n = 4$ on the intervals $[-1, 0]$ and $[0, 1]$.

c. Use Gaussian Quadrature with $n = 2$ on the intervals $[-1, -0.5]$, $[-0.5, 0]$, $[0, 0.5]$ and $[0.5, 1]$.

d. Give an explanation for the accuracy of the results.

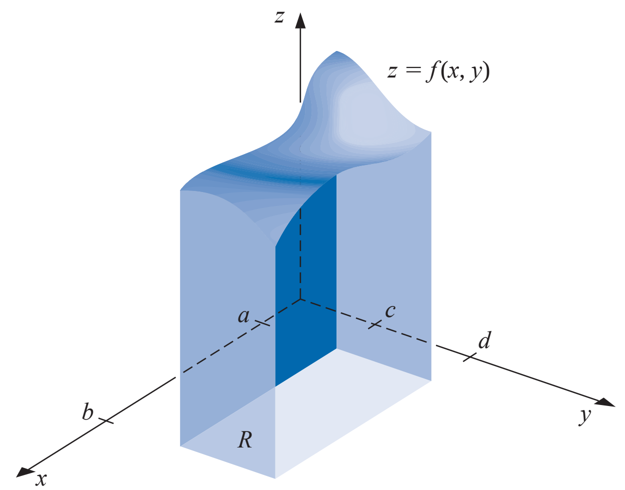
4.8 Multiple Integrals

The techniques discussed in the previous sections can be modified for use in the approximation of multiple integrals. Consider the double integral

$$\iint_R f(x, y)\, dA,$$

where $R = \{ (x, y) \mid a \le x \le b, c \le y \le d \}$, for some constants a, b, c, and d, is a rectangular region in the plane. (See Figure 4.18.)

Figure 4.18



The following illustration shows how the Composite Trapezoidal rule using two subintervals in each coordinate direction would be applied to this integral.

Illustration Writing the double integral as an iterated integral gives

$$\iint_R f(x, y)\, dA = \int_a^b \left(\int_c^d f(x, y)\, dy \right) dx.$$

To simplify notation, let $k = (d-c)/2$ and $h = (b-a)/2$. Apply the Composite Trapezoidal rule to the interior integral to obtain

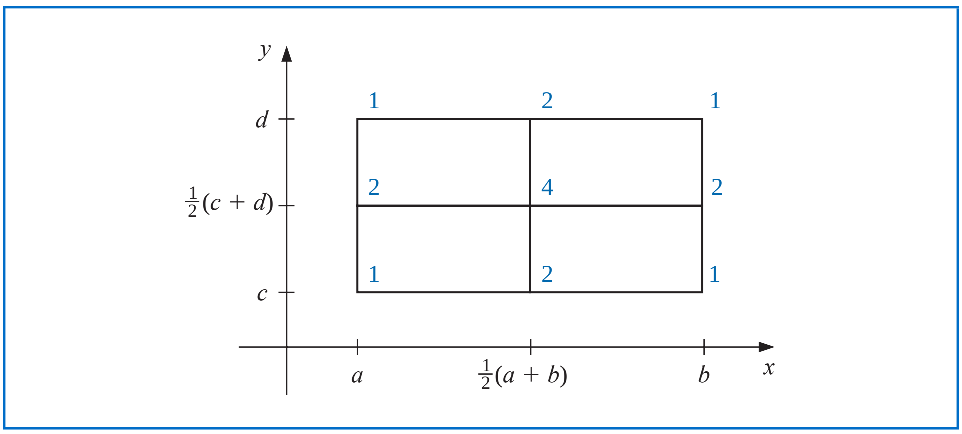
$$\int_c^d f(x,y)\,dy \approx \frac{k}{2}\left[f(x,c) + f(x,d) + 2f\left(x, \frac{c+d}{2}\right)\right].$$

This approximation is of order $O\left((d-c)^3\right)$. Then apply the Composite Trapezoidal rule again to approximate the integral of this function of x:

$$\begin{aligned}
\int_a^b \left(\int_c^d f(x,y)\,dy\right) dx &\approx \int_a^b \left(\frac{d-c}{4}\right)\left[f(x,c) + 2f\left(x, \frac{c+d}{2}\right) + f(d)\right] dx \\
&= \frac{b-a}{4}\left(\frac{d-c}{4}\right)\left[f(a,c) + 2f\left(a, \frac{c+d}{2}\right) + f(a,d)\right] \\
&\quad + \frac{b-a}{4}\left(2\left(\frac{d-c}{4}\right)\left[f\left(\frac{a+b}{2}, c\right)\right.\right. \\
&\quad \left.\left. + 2f\left(\frac{a+b}{2}, \frac{c+d}{2}\right) + \left(\frac{a+b}{2}, d\right)\right]\right) \\
&\quad + \frac{b-a}{4}\left(\frac{d-c}{4}\right)\left[f(b,c) + 2f\left(b, \frac{c+d}{2}\right) + f(b,d)\right] \\
&= \frac{(b-a)(d-c)}{16}\left[f(a,c) + f(a,d) + f(b,c) + f(b,d)\right. \\
&\quad + 2\left(f\left(\frac{a+b}{2}, c\right) + f\left(\frac{a+b}{2}, d\right) + f\left(a, \frac{c+d}{2}\right)\right. \\
&\quad \left.\left. + f\left(b, \frac{c+d}{2}\right)\right) + 4f\left(\frac{a+b}{2}, \frac{c+d}{2}\right)\right]
\end{aligned}$$

This approximation is of order $O\left((b-a)(d-c)\left[(b-a)^2 + (d-c)^2\right]\right)$. Figure 4.19 shows a grid with the number of functional evaluations at each of the nodes used in the approximation. □

Figure 4.19



As the illustration shows, the procedure is quite straightforward. But the number of function evaluations grows with the square of the number required for a single integral. In a practical situation we would not expect to use a method as elementary as the Composite Trapezoidal rule. Instead we will employ the Composite Simpson's rule to illustrate the general approximation technique, although any other composite formula could be used in its place.

To apply the Composite Simpson's rule, we divide the region R by partitioning both $[a, b]$ and $[c, d]$ into an even number of subintervals. To simplify the notation, we choose even integers n and m and partition $[a, b]$ and $[c, d]$ with the evenly spaced mesh points $x_0, x_1, \ldots, x_n$ and $y_0, y_1, \ldots, y_m$, respectively. These subdivisions determine step sizes $h = (b-a)/n$ and $k = (d-c)/m$. Writing the double integral as the iterated integral

$$\iint_R f(x, y)\, dA = \int_a^b \left(\int_c^d f(x, y)\, dy \right) dx,$$

we first use the Composite Simpson's rule to approximate

$$\int_c^d f(x, y)\, dy,$$

treating x as a constant.

Let $y_j = c + jk$, for each $j = 0, 1, \ldots, m$. Then

$$\int_c^d f(x, y)\, dy = \frac{k}{3}\left[f(x, y_0) + 2\sum_{j=1}^{(m/2)-1} f(x, y_{2j}) + 4\sum_{j=1}^{m/2} f(x, y_{2j-1}) + f(x, y_m) \right]$$
$$- \frac{(d-c)k^4}{180}\frac{\partial^4 f}{\partial y^4}(x, \mu),$$

for some μ in (c, d). Thus

$$\int_a^b \int_c^d f(x, y)\, dy\, dx = \frac{k}{3}\Bigg[\int_a^b f(x, y_0)\, dx + 2\sum_{j=1}^{(m/2)-1} \int_a^b f(x, y_{2j})\, dx$$
$$+ 4\sum_{j=1}^{m/2} \int_a^b f(x, y_{2j-1})\, dx + \int_a^b f(x, y_m)\, dx \Bigg]$$
$$- \frac{(d-c)k^4}{180} \int_a^b \frac{\partial^4 f}{\partial y^4}(x, \mu)\, dx.$$

Composite Simpson's rule is now employed on the integrals in this equation. Let $x_i = a + ih$, for each $i = 0, 1, \ldots, n$. Then for each $j = 0, 1, \ldots, m$, we have

$$\int_a^b f(x, y_j)\, dx = \frac{h}{3}\left[f(x_0, y_j) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i}, y_j) + 4\sum_{i=1}^{n/2} f(x_{2i-1}, y_j) + f(x_n, y_j) \right]$$
$$- \frac{(b-a)h^4}{180}\frac{\partial^4 f}{\partial x^4}(\xi_j, y_j),$$

for some ξ_j in (a, b). The resulting approximation has the form

$$\int_a^b \int_c^d f(x,y)\,dy\,dx \approx \frac{hk}{9}\Bigg\{\Bigg[f(x_0,y_0) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i},y_0) + 4\sum_{i=1}^{n/2} f(x_{2i-1},y_0) + f(x_n,y_0)\Bigg]$$

$$+ 2\Bigg[\sum_{j=1}^{(m/2)-1} f(x_0,y_{2j}) + 2\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{(n/2)-1} f(x_{2i},y_{2j}) + 4\sum_{j=1}^{(m/2)-1}\sum_{i=1}^{n/2} f(x_{2i-1},y_{2j}) + \sum_{j=1}^{(m/2)-1} f(x_n,y_{2j})\Bigg]$$

$$+ 4\Bigg[\sum_{j=1}^{m/2} f(x_0,y_{2j-1}) + 2\sum_{j=1}^{m/2}\sum_{i=1}^{(n/2)-1} f(x_{2i},y_{2j-1}) + 4\sum_{j=1}^{m/2}\sum_{i=1}^{n/2} f(x_{2i-1},y_{2j-1}) + \sum_{j=1}^{m/2} f(x_n,y_{2j-1})\Bigg]$$

$$+ \Bigg[f(x_0,y_m) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i},y_m) + 4\sum_{i=1}^{n/2} f(x_{2i-1},y_m) + f(x_n,y_m)\Bigg]\Bigg\}.$$

The error term E is given by

$$E = \frac{-k(b-a)h^4}{540}\Bigg[\frac{\partial^4 f}{\partial x^4}(\xi_0,y_0) + 2\sum_{j=1}^{(m/2)-1}\frac{\partial^4 f}{\partial x^4}(\xi_{2j},y_{2j}) + 4\sum_{j=1}^{m/2}\frac{\partial^4 f}{\partial x^4}(\xi_{2j-1},y_{2j-1}) + \frac{\partial^4 f}{\partial x^4}(\xi_m,y_m)\Bigg] - \frac{(d-c)k^4}{180}\int_a^b \frac{\partial^4 f}{\partial y^4}(x,\mu)\,dx.$$

If $\partial^4 f/\partial x^4$ is continuous, the Intermediate Value Theorem 1.11 can be repeatedly applied to show that the evaluation of the partial derivatives with respect to x can be replaced by a common value and that

$$E = \frac{-k(b-a)h^4}{540}\left[3m\frac{\partial^4 f}{\partial x^4}(\overline{\eta},\overline{\mu})\right] - \frac{(d-c)k^4}{180}\int_a^b \frac{\partial^4 f}{\partial y^4}(x,\mu)\,dx,$$

for some $(\overline{\eta},\overline{\mu})$ in R. If $\partial^4 f/\partial y^4$ is also continuous, the Weighted Mean Value Theorem for Integrals 1.13 implies that

$$\int_a^b \frac{\partial^4 f}{\partial y^4}(x,\mu)\,dx = (b-a)\frac{\partial^4 f}{\partial y^4}(\hat{\eta},\hat{\mu}),$$

for some $(\hat{\eta},\hat{\mu})$ in R. Because $m = (d-c)/k$, the error term has the form

$$E = \frac{-k(b-a)h^4}{540}\left[3m\frac{\partial^4 f}{\partial x^4}(\overline{\eta},\overline{\mu})\right] - \frac{(d-c)(b-a)}{180}k^4\frac{\partial^4 f}{\partial y^4}(\hat{\eta},\hat{\mu})$$

which simplifies to

$$E = -\frac{(d-c)(b-a)}{180}\left[h^4\frac{\partial^4 f}{\partial x^4}(\overline{\eta},\overline{\mu}) + k^4\frac{\partial^4 f}{\partial y^4}(\hat{\eta},\hat{\mu})\right],$$

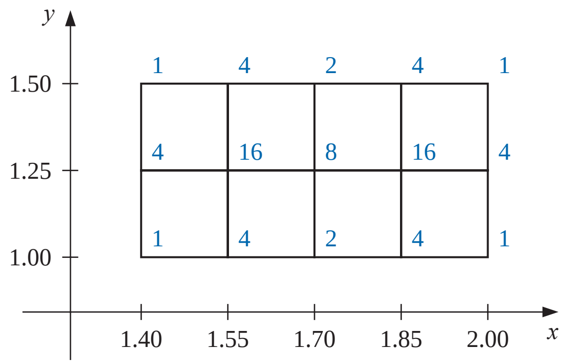
for some $(\overline{\eta},\overline{\mu})$ and $(\hat{\eta},\hat{\mu})$ in R.

Example 1 Use Composite Simpson's rule with $n = 4$ and $m = 2$ to approximate

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\, dy\, dx,$$

Solution The step sizes for this application are $h = (2.0 - 1.4)/4 = 0.15$ and $k = (1.5 - 1.0)/2 = 0.25$. The region of integration R is shown in Figure 4.20, together with the nodes (x_i, y_j), where $i = 0, 1, 2, 3, 4$ and $j = 0, 1, 2$. It also shows the coefficients $w_{i,j}$ of $f(x_i, y_i) = \ln(x_i + 2y_i)$ in the sum that gives the Composite Simpson's rule approximation to the integral.

Figure 4.20



The approximation is

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\, dy\, dx \approx \frac{(0.15)(0.25)}{9}\sum_{i=0}^{4}\sum_{j=0}^{2} w_{i,j}\ln(x_i + 2y_j)$$
$$= 0.4295524387.$$

We have

$$\frac{\partial^4 f}{\partial x^4}(x,y) = \frac{-6}{(x+2y)^4} \quad \text{and} \quad \frac{\partial^4 f}{\partial y^4}(x,y) = \frac{-96}{(x+2y)^4},$$

and the maximum values of the absolute values of these partial derivatives occur on R when $x = 1.4$ and $y = 1.0$. So the error is bounded by

$$|E| \le \frac{(0.5)(0.6)}{180}\left[(0.15)^4 \max_{(x,y)\text{in}R}\frac{6}{(x+2y)^4} + (0.25)^4 \max_{(x,y)\text{in}R}\frac{96}{(x+2y)^4}\right] \le 4.72 \times 10^{-6}.$$

The actual value of the integral to ten decimal places is

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\, dy\, dx = 0.4295545265,$$

so the approximation is accurate to within 2.1×10^{-6}. ■

The same techniques can be applied for the approximation of triple integrals as well as higher integrals for functions of more than three variables. The number of functional evaluations required for the approximation is the product of the number of functional evaluations required when the method is applied to each variable.

Gaussian Quadrature for Double Integral Approximation

To reduce the number of functional evaluations, more efficient methods such as Gaussian quadrature, Romberg integration, or Adaptive quadrature can be incorporated in place of the Newton-Cotes formulas. The following example illustrates the use of Gaussian quadrature for the integral considered in Example 1.

Example 2 Use Gaussian quadrature with $n = 3$ in both dimensions to approximate the integral

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\,dy\,dx.$$

Solution Before employing Gaussian quadrature to approximate this integral, we need to transform the region of integration

$$R = \{\,(x,y) \mid 1.4 \le x \le 2.0, 1.0 \le y \le 1.5\,\}$$

into

$$\hat{R} = \{\,(u,v) \mid -1 \le u \le 1, -1 \le v \le 1\,\}.$$

The linear transformations that accomplish this are

$$u = \frac{1}{2.0-1.4}(2x - 1.4 - 2.0) \quad \text{and} \quad v = \frac{1}{1.5-1.0}(2y - 1.0 - 1.5),$$

or, equivalently, $x = 0.3u + 1.7$ and $y = 0.25v + 1.25$. Employing this change of variables gives an integral on which Gaussian quadrature can be applied:

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\,dy\,dx = 0.075\int_{-1}^{1}\int_{-1}^{1} \ln(0.3u + 0.5v + 4.2)\,dv\,du.$$

The Gaussian quadrature formula for $n = 3$ in both u and v requires that we use the nodes

$$u_1 = v_1 = r_{3,2} = 0, \quad u_0 = v_0 = r_{3,1} = -0.7745966692,$$

and

$$u_2 = v_2 = r_{3,3} = 0.7745966692.$$

The associated weights are $c_{3,2} = 0.\overline{8}$ and $c_{3,1} = c_{3,3} = 0.\overline{5}$. (These are given in Table 4.12 on page 232.) The resulting approximation is

$$\int_{1.4}^{2.0}\int_{1.0}^{1.5} \ln(x+2y)\,dy\,dx \approx 0.075\sum_{i=1}^{3}\sum_{j=1}^{3} c_{3,i}c_{3,j}\ln(0.3r_{3,i} + 0.5r_{3,j} + 4.2)$$
$$= 0.4295545313.$$

Although this result requires only 9 functional evaluations compared to 15 for the Composite Simpson's rule considered in Example 1, it is accurate to within 4.8×10^{-9}, compared to 2.1×10^{-6} accuracy in Example 1. ■

Non-Rectangular Regions

The use of approximation methods for double integrals is not limited to integrals with rectangular regions of integration. The techniques previously discussed can be modified to approximate double integrals of the form

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx \tag{4.42}$$

or

$$\int_c^d \int_{a(y)}^{b(y)} f(x, y)\, dx\, dy. \tag{4.43}$$

In fact, integrals on regions not of this type can also be approximated by performing appropriate partitions of the region. (See Exercise 10.)

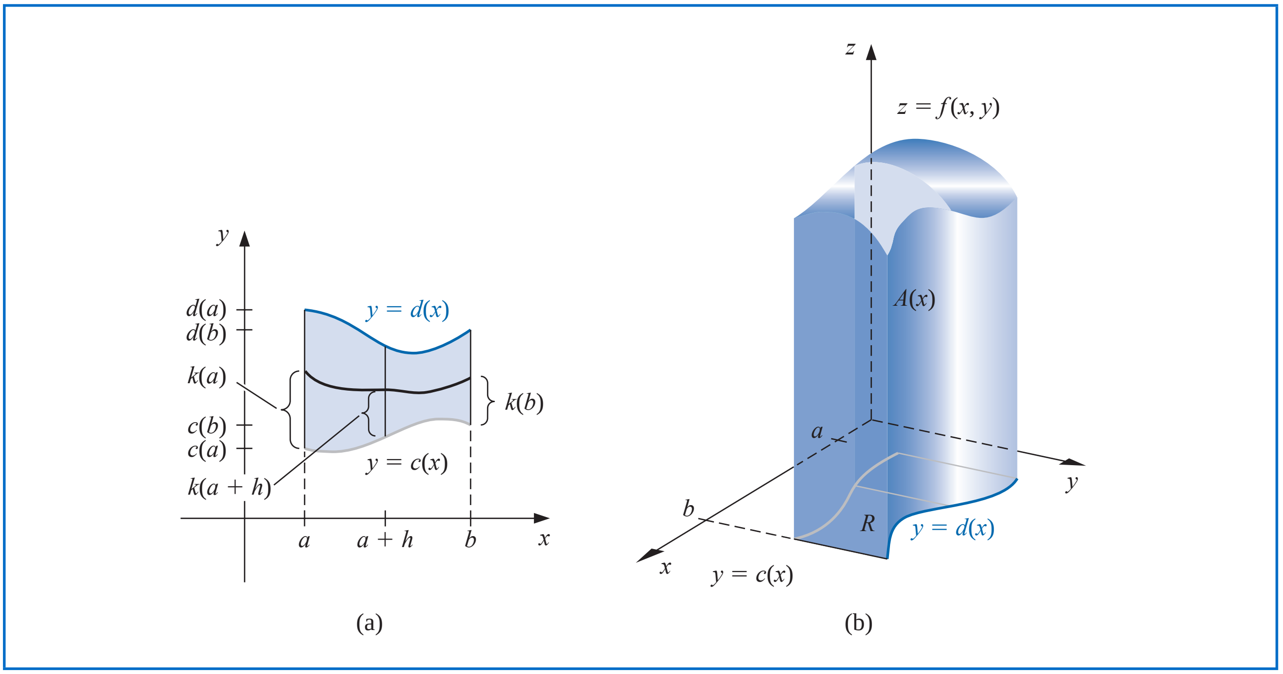
To describe the technique involved with approximating an integral in the form

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx,$$

we will use the basic Simpson's rule to integrate with respect to both variables. The step size for the variable x is $h = (b - a)/2$, but the step size for y varies with x (see Figure 4.21) and is written

$$k(x) = \frac{d(x) - c(x)}{2}.$$

Figure 4.21



This gives

$$\int_a^b \int_{c(x)}^{d(x)} f(x,y)\,dy\,dx \approx \int_a^b \frac{k(x)}{3}[f(x,c(x)) + 4f(x,c(x)+k(x)) + f(x,d(x))]\,dx$$

$$\approx \frac{h}{3}\Bigg\{\frac{k(a)}{3}[f(a,c(a)) + 4f(a,c(a)+k(a)) + f(a,d(a))]$$

$$+ \frac{4k(a+h)}{3}[f(a+h,c(a+h)) + 4f(a+h,c(a+h)$$

$$+ k(a+h)) + f(a+h,d(a+h))]$$

$$+ \frac{k(b)}{3}[f(b,c(b)) + 4f(b,c(b)+k(b)) + f(b,d(b))]\Bigg\}.$$

Algorithm 4.4 applies the Composite Simpson's rule to an integral in the form (4.42). Integrals in the form (4.43) can, of course, be handled similarly.

ALGORITHM
4.4

Simpson's Double Integral

To approximate the integral

$$I = \int_a^b \int_{c(x)}^{d(x)} f(x,y)\,dy\,dx :$$

INPUT endpoints a, b: even positive integers m, n.

OUTPUT approximation J to I.

Step 1 Set $h = (b-a)/n$;
$J_1 = 0$; (*End terms.*)
$J_2 = 0$; (*Even terms.*)
$J_3 = 0$. (*Odd terms.*)

Step 2 For $i = 0, 1, \ldots, n$ do Steps 3–8.

Step 3 Set $x = a + ih$; (*Composite Simpson's method for* x.)
$HX = (d(x) - c(x))/m$;
$K_1 = f(x, c(x)) + f(x, d(x))$; (*End terms.*)
$K_2 = 0$; (*Even terms.*)
$K_3 = 0$. (*Odd terms.*)

Step 4 For $j = 1, 2, \ldots, m-1$ do Step 5 and 6.

Step 5 Set $y = c(x) + jHX$;
$Q = f(x, y)$.

Step 6 If j is even then set $K_2 = K_2 + Q$
else set $K_3 = K_3 + Q$.

Step 7 Set $L = (K_1 + 2K_2 + 4K_3)HX/3$.

$$\left(L \approx \int_{c(x_i)}^{d(x_i)} f(x_i, y)\,dy \quad \textit{by the Composite Simpson's method.}\right)$$

Step 8 If $i = 0$ or $i = n$ then set $J_1 = J_1 + L$
else if i is even then set $J_2 = J_2 + L$
else set $J_3 = J_3 + L$.

Step 9 Set $J = h(J_1 + 2J_2 + 4J_3)/3$.

Step 10 OUTPUT (J);
STOP.

To apply Gaussian quadrature to the double integral

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx,$$

first requires transforming, for each x in $[a, b]$, the variable y in the interval $[c(x), d(x)]$ into the variable t in the interval $[-1, 1]$. This linear transformation gives

$$f(x, y) = f\left(x, \frac{(d(x) - c(x))t + d(x) + c(x)}{2}\right) \quad \text{and} \quad dy = \frac{d(x) - c(x)}{2}\, dt.$$

The reduced calculation makes it generally worthwhile to apply Gaussian quadrature rather than a Simpson's technique when approximating double integrals.

Then, for each x in $[a, b]$, we apply Gaussian quadrature to the resulting integral

$$\int_{c(x)}^{d(x)} f(x, y)\, dy = \int_{-1}^{1} f\left(x, \frac{(d(x) - c(x))t + d(x) + c(x)}{2}\right) dt$$

to produce

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx \approx \int_a^b \frac{d(x) - c(x)}{2} \sum_{j=1}^{n} c_{n,j} f\left(x, \frac{(d(x) - c(x))r_{n,j} + d(x) + c(x)}{2}\right) dx,$$

where, as before, the roots $r_{n,j}$ and coefficients $c_{n,j}$ come from Table 4.12 on page 232. Now the interval $[a, b]$ is transformed to $[-1, 1]$, and Gaussian quadrature is applied to approximate the integral on the right side of this equation. The details are given in Algorithm 4.5.

ALGORITHM 4.5

Gaussian Double Integral

To approximate the integral

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx :$$

INPUT endpoints a, b; positive integers m, n.
(*The roots $r_{i,j}$ and coefficients $c_{i,j}$ need to be available for $i = \max\{m, n\}$ and for $1 \le j \le i$.*)

OUTPUT approximation J to I.

Step 1 Set $h_1 = (b - a)/2$;
$h_2 = (b + a)/2$;
$J = 0$.

Step 2 For $i = 1, 2, \ldots, m$ do Steps 3–5.

Step 3 Set $JX = 0$;
$x = h_1 r_{m,i} + h_2$;
$d_1 = d(x)$;
$c_1 = c(x)$;
$k_1 = (d_1 - c_1)/2$;
$k_2 = (d_1 + c_1)/2$.

Step 4 For $j = 1, 2, \ldots, n$ do
set $y = k_1 r_{n,j} + k_2$;
$Q = f(x, y)$;
$JX = JX + c_{n,j} Q$.

Step 5 Set $J = J + c_{m,i} k_1 JX$.

Step 6 Set $J = h_1 J$.

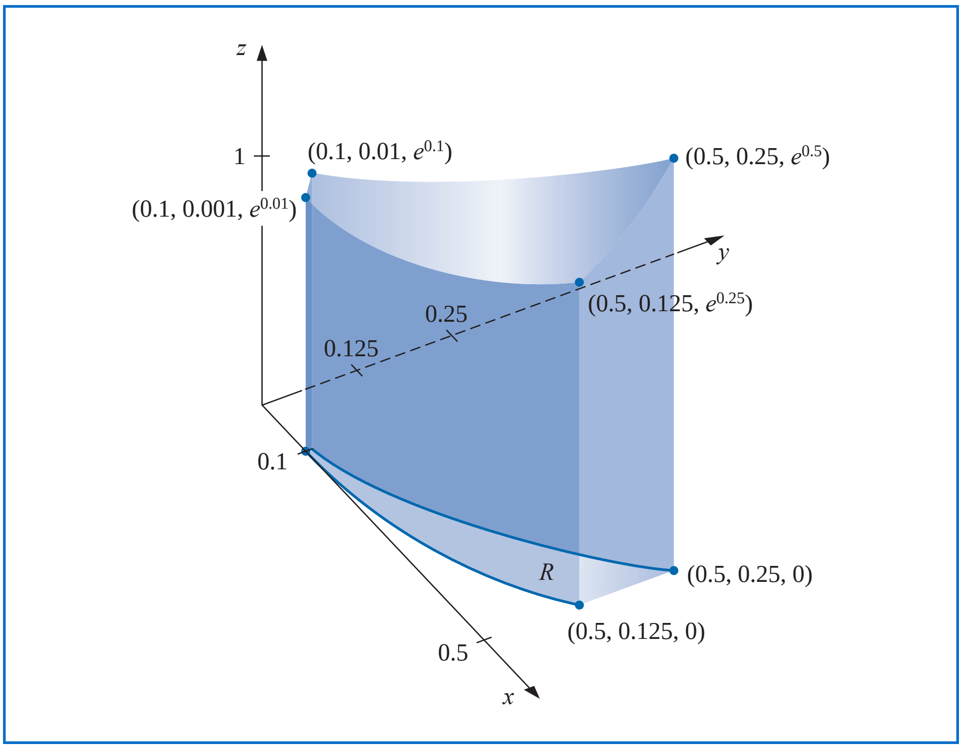
Step 7 OUTPUT (J);
STOP. ■

Illustration The volume of the solid in Figure 4.22 is approximated by applying Simpson's Double Integral Algorithm with $n = m = 10$ to

$$\int_{0.1}^{0.5} \int_{x^3}^{x^2} e^{y/x}\, dy\, dx.$$

This requires 121 evaluations of the function $f(x, y) = e^{y/x}$ and produces the value 0.0333054, which approximates the volume of the solid shown in Figure 4.22 to nearly seven decimal places. Applying the Gaussian Quadrature Algorithm with $n = m = 5$ requires only 25 function evaluations and gives the approximation 0.03330556611, which is accurate to 11 decimal places. □

Figure 4.22



Triple Integral Approximation

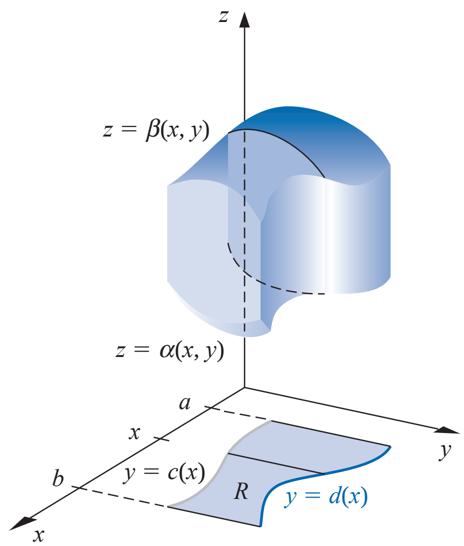
Triple integrals of the form

$$\int_a^b \int_{c(x)}^{d(x)} \int_{\alpha(x,y)}^{\beta(x,y)} f(x, y, z)\, dz\, dy\, dx$$

The reduced calculation makes it almost always worthwhile to apply Gaussian quadrature rather than a Simpson's technique when approximating triple or higher integrals.

(see Figure 4.23) are approximated in a similar manner. Because of the number of calculations involved, Gaussian quadrature is the method of choice. Algorithm 4.6 implements this procedure.

Figure 4.23



ALGORITHM 4.6

Gaussian Triple Integral

To approximate the integral

$$\int_a^b \int_{c(x)}^{d(x)} \int_{\alpha(x,y)}^{\beta(x,y)} f(x, y, z)\, dz\, dy\, dx :$$

INPUT endpoints a, b; positive integers m, n, p.
(*The roots* $r_{i,j}$ *and coefficients* $c_{i,j}$ *need to be available for* $i = \max\{n, m, p\}$ *and for* $1 \le j \le i$.)

OUTPUT approximation J to I.

Step 1 Set $h_1 = (b - a)/2$;
$h_2 = (b + a)/2$;
$J = 0$.

Step 2 For $i = 1, 2, \ldots, m$ do Steps 3–8.

Step 3 Set $JX = 0$;
$x = h_1 r_{m,i} + h_2$;
$d_1 = d(x)$;
$c_1 = c(x)$;
$k_1 = (d_1 - c_1)/2$;
$k_2 = (d_1 + c_1)/2$.

Step 4 For $j = 1, 2, \ldots, n$ do Steps 5–7.

Step 5 Set $JY = 0$;
$y = k_1 r_{n,j} + k_2$;
$\beta_1 = \beta(x, y)$;
$\alpha_1 = \alpha(x, y)$;
$l_1 = (\beta_1 - \alpha_1)/2$;
$l_2 = (\beta_1 + \alpha_1)/2$.

Step 6 For $k = 1, 2, \ldots, p$ do
set $z = l_1 r_{p,k} + l_2$;
$Q = f(x, y, z)$;
$JY = JY + c_{p,k} Q$.

Step 7 Set $JX = JX + c_{n,j} l_1 JY$.

Step 8 Set $J = J + c_{m,i} k_1 JX$.

Step 9 Set $J = h_1 J$.

Step 10 OUTPUT (J);
STOP. ■

The following example requires the evaluation of four triple integrals.

Illustration The center of a mass of a solid region D with density function σ occurs at

$$(\bar{x}, \bar{y}, \bar{z}) = \left(\frac{M_{yz}}{M}, \frac{M_{xz}}{M}, \frac{M_{xy}}{M} \right),$$

where

$$M_{yz} = \iiint_D x\sigma(x, y, z)\, dV, \quad M_{xz} = \iiint_D y\sigma(x, y, z)\, dV$$

and

$$M_{xy} = \iiint_D z\sigma(x, y, z)\, dV$$

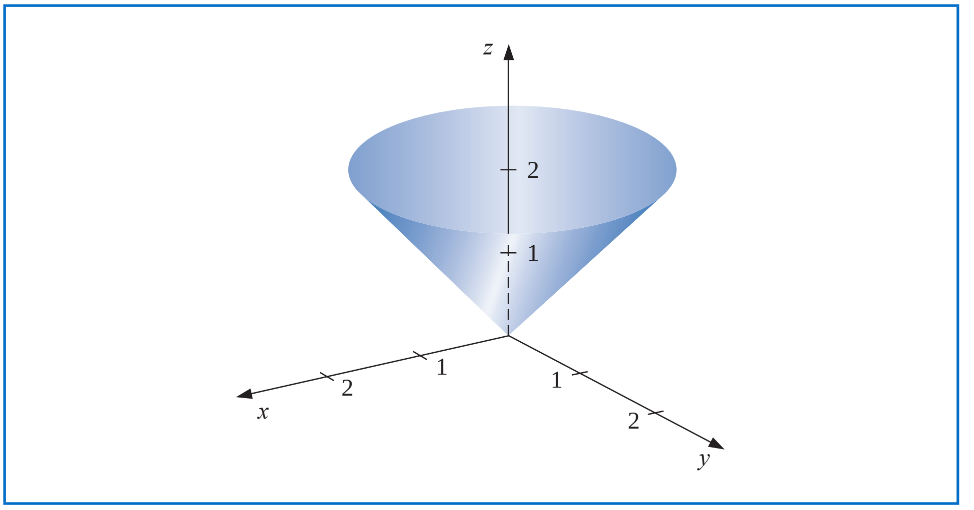
are the moments about the coordinate planes and the mass of D is

$$M = \iiint_D \sigma(x, y, z)\, dV.$$

The solid shown in Figure 4.24 is bounded by the upper nappe of the cone $z^2 = x^2 + y^2$ and the plane $z = 2$. Suppose that this solid has density function given by

$$\sigma(x, y, z) = \sqrt{x^2 + y^2}.$$

Figure 4.24



Applying the Gaussian Triple Integral Algorithm 4.6 with $n = m = p = 5$ requires 125 function evaluations per integral and gives the following approximations:

$$M = \int_{-2}^{2}\int_{-\sqrt{4-x^2}}^{\sqrt{4-x^2}}\int_{\sqrt{x^2+y^2}}^{2} \sqrt{x^2+y^2}\,dz\,dy\,dx$$

$$= 4\int_{0}^{2}\int_{0}^{\sqrt{4-x^2}}\int_{\sqrt{x^2+y^2}}^{2} \sqrt{x^2+y^2}\,dz\,dy\,dx \approx 8.37504476,$$

$$M_{yz} = \int_{-2}^{2}\int_{-\sqrt{4-x^2}}^{\sqrt{4-x^2}}\int_{\sqrt{x^2+y^2}}^{2} x\sqrt{x^2+y^2}\,dz\,dy\,dx \approx -5.55111512 \times 10^{-17},$$

$$M_{xz} = \int_{-2}^{2}\int_{-\sqrt{4-x^2}}^{\sqrt{4-x^2}}\int_{\sqrt{x^2+y^2}}^{2} y\sqrt{x^2+y^2}\,dz\,dy\,dx \approx -8.01513675 \times 10^{-17},$$

$$M_{xy} = \int_{-2}^{2}\int_{-\sqrt{4-x^2}}^{\sqrt{4-x^2}}\int_{\sqrt{x^2+y^2}}^{2} z\sqrt{x^2+y^2}\,dz\,dy\,dx \approx 13.40038156.$$

This implies that the approximate location of the center of mass is

$$(\overline{x}, \overline{y}, \overline{z}) = (0, 0, 1.60003701).$$

These integrals are quite easy to evaluate directly. If you do this, you will find that the exact center of mass occurs at $(0, 0, 1.6)$. □

Multiple integrals can be evaluated in Maple using the *MultInt* command in the *MultivariateCalculus* subpackage of the *Student* package. For example, to evaluate the multiple integral

$$\int_{2}^{4}\int_{x-1}^{x+6}\int_{-2}^{4+y^2} x^2 + y^2 + z\,dz\,dy\,dx$$

we first load the package and define the function with

$with(Student[MultivariateCalculus])$: $f := (x, y, z) \to x^2 + y^2 + z$

Then issue the command

$MultiInt(f(x, y, z), z = -2..4 + y^2,\ y = x - 1..x + 6,\ x = 2..4)$

which produces the result

$$1.995885970$$

EXERCISE SET 4.8

1. Use Algorithm 4.4 with $n = m = 4$ to approximate the following double integrals, and compare the results to the exact answers.

   a. $\int_{2.1}^{2.5} \int_{1.2}^{1.4} xy^2 \, dy \, dx$ **b.** $\int_{0}^{0.5} \int_{0}^{0.5} e^{y-x} \, dy \, dx$

   c. $\int_{2}^{2.2} \int_{x}^{2x} (x^2 + y^3) \, dy \, dx$ **d.** $\int_{1}^{1.5} \int_{0}^{x} (x^2 + \sqrt{y}) \, dy \, dx$

2. Find the smallest values for $n = m$ so that Algorithm 4.4 can be used to approximate the integrals in Exercise 1 to within 10^{-6} of the actual value.

3. Use Algorithm 4.4 with (i) $n = 4$, $m = 8$, (ii) $n = 8$, $m = 4$, and (iii) $n = m = 6$ to approximate the following double integrals, and compare the results to the exact answers.

   a. $\int_{0}^{\pi/4} \int_{\sin x}^{\cos x} (2y \sin x + \cos^2 x) \, dy \, dx$ **b.** $\int_{1}^{e} \int_{1}^{x} \ln xy \, dy \, dx$

   c. $\int_{0}^{1} \int_{x}^{2x} (x^2 + y^3) \, dy \, dx$ **d.** $\int_{0}^{1} \int_{x}^{2x} (y^2 + x^3) \, dy \, dx$

   e. $\int_{0}^{\pi} \int_{0}^{x} \cos x \, dy \, dx$ **f.** $\int_{0}^{\pi} \int_{0}^{x} \cos y \, dy \, dx$

   g. $\int_{0}^{\pi/4} \int_{0}^{\sin x} \frac{1}{\sqrt{1 - y^2}} \, dy \, dx$ **h.** $\int_{-\pi}^{3\pi/2} \int_{0}^{2\pi} (y \sin x + x \cos y) \, dy \, dx$

4. Find the smallest values for $n = m$ so that Algorithm 4.4 can be used to approximate the integrals in Exercise 3 to within 10^{-6} of the actual value.

5. Use Algorithm 4.5 with $n = m = 2$ to approximate the integrals in Exercise 1, and compare the results to those obtained in Exercise 1.

6. Find the smallest values of $n = m$ so that Algorithm 4.5 can be used to approximate the integrals in Exercise 1 to within 10^{-6}. Do not continue beyond $n = m = 5$. Compare the number of functional evaluations required to the number required in Exercise 2.

7. Use Algorithm 4.5 with (i) $n = m = 3$, (ii) $n = 3$, $m = 4$, (iii) $n = 4$, $m = 3$, and (iv) $n = m = 4$ to approximate the integrals in Exercise 3.

8. Use Algorithm 4.5 with $n = m = 5$ to approximate the integrals in Exercise 3. Compare the number of functional evaluations required to the number required in Exercise 4.

9. Use Algorithm 4.4 with $n = m = 14$ and Algorithm 4.5 with $n = m = 4$ to approximate

$$\iint_R e^{-(x+y)} \, dA,$$

for the region R in the plane bounded by the curves $y = x^2$ and $y = \sqrt{x}$.

10. Use Algorithm 4.4 to approximate

$$\iint_R \sqrt{xy + y^2}\, dA,$$

where R is the region in the plane bounded by the lines $x + y = 6$, $3y - x = 2$, and $3x - y = 2$. First partition R into two regions R_1 and R_2 on which Algorithm 4.4 can be applied. Use $n = m = 6$ on both R_1 and R_2.

11. A plane lamina is a thin sheet of continuously distributed mass. If σ is a function describing the density of a lamina having the shape of a region R in the xy-plane, then the center of the mass of the lamina $(\bar{x}, \bar{y})$ is

$$\bar{x} = \frac{\iint_R x\sigma(x, y)\, dA}{\iint_R \sigma(x, y)\, dA}, \quad \bar{y} = \frac{\iint_R y\sigma(x, y)\, dA}{\iint_R \sigma(x, y)\, dA}.$$

Use Algorithm 4.4 with $n = m = 14$ to find the center of mass of the lamina described by $R = \{(x, y) \mid 0 \le x \le 1, 0 \le y \le \sqrt{1 - x^2}\}$ with the density function $\sigma(x, y) = e^{-(x^2+y^2)}$. Compare the approximation to the exact result.

12. Repeat Exercise 11 using Algorithm 4.5 with $n = m = 5$.

13. The area of the surface described by $z = f(x, y)$ for (x, y) in R is given by

$$\iint_R \sqrt{[f_x(x, y)]^2 + [f_y(x, y)]^2 + 1}\, dA.$$

Use Algorithm 4.4 with $n = m = 8$ to find an approximation to the area of the surface on the hemisphere $x^2 + y^2 + z^2 = 9$, $z \ge 0$ that lies above the region in the plane described by $R = \{ (x, y) \mid 0 \le x \le 1, 0 \le y \le 1 \}$.

14. Repeat Exercise 13 using Algorithm 4.5 with $n = m = 4$.

15. Use Algorithm 4.6 with $n = m = p = 2$ to approximate the following triple integrals, and compare the results to the exact answers.

a. $\displaystyle\int_0^1 \int_1^2 \int_0^{0.5} e^{x+y+z}\, dz\, dy\, dx$

b. $\displaystyle\int_0^1 \int_x^1 \int_0^y y^2 z\, dz\, dy\, dx$

c. $\displaystyle\int_0^1 \int_{x^2}^x \int_{x-y}^{x+y} y\, dz\, dy\, dx$

d. $\displaystyle\int_0^1 \int_{x^2}^x \int_{x-y}^{x+y} z\, dz\, dy\, dx$

e. $\displaystyle\int_0^{\pi} \int_0^x \int_0^{xy} \frac{1}{y} \sin\frac{z}{y}\, dz\, dy\, dx$

f. $\displaystyle\int_0^1 \int_0^1 \int_{-xy}^{xy} e^{x^2+y^2}\, dz\, dy\, dx$

16. Repeat Exercise 15 using $n = m = p = 3$.

17. Repeat Exercise 15 using $n = m = p = 4$ and $n = m = p = 5$.

18. Use Algorithm 4.6 with $n = m = p = 4$ to approximate

$$\iiint_S xy \sin(yz)\, dV,$$

where S is the solid bounded by the coordinate planes and the planes $x = \pi$, $y = \pi/2$, $z = \pi/3$. Compare this approximation to the exact result.

19. Use Algorithm 4.6 with $n = m = p = 5$ to approximate

$$\iiint_S \sqrt{xyz}\, dV,$$

where S is the region in the first octant bounded by the cylinder $x^2 + y^2 = 4$, the sphere $x^2 + y^2 + z^2 = 4$, and the plane $x + y + z = 8$. How many functional evaluations are required for the approximation?

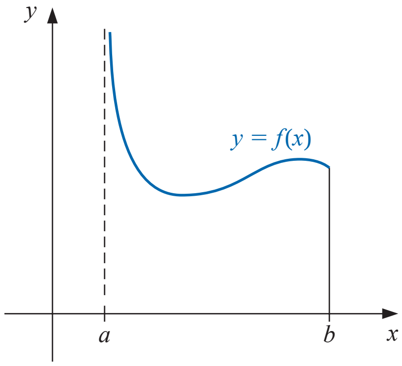
4.9 Improper Integrals

Improper integrals result when the notion of integration is extended either to an interval of integration on which the function is unbounded or to an interval with one or more infinite endpoints. In either circumstance, the normal rules of integral approximation must be modified.

Left Endpoint Singularity

We will first consider the situation when the integrand is unbounded at the left endpoint of the interval of integration, as shown in Figure 4.25. In this case we say that f has a **singularity** at the endpoint a. We will then show how other improper integrals can be reduced to problems of this form.

Figure 4.25



It is shown in calculus that the improper integral with a singularity at the left endpoint,

$$\int_a^b \frac{dx}{(x-a)^p},$$

converges if and only if $0 < p < 1$, and in this case, we define

$$\int_a^b \frac{1}{(x-a)^p}\,dx = \lim_{M\to a^+} \left.\frac{(x-a)^{1-p}}{1-p}\right|_{x=M}^{x=b} = \frac{(b-a)^{1-p}}{1-p}.$$

Example 1 Show that the improper integral $\int_0^1 \frac{1}{\sqrt{x}}\,dx$ converges but $\int_0^1 \frac{1}{x^2}\,dx$ diverges.

Solution For the first integral we have

$$\int_0^1 \frac{1}{\sqrt{x}}\,dx = \lim_{M\to 0^+} \int_M^1 x^{-1/2}\,dx = \lim_{M\to 0^+} 2x^{1/2}\Big|_{x=M}^{x=1} = 2 - 0 = 2,$$

but the second integral

$$\int_0^1 \frac{1}{x^2}\,dx = \lim_{M\to 0^+} \int_M^1 x^{-2}\,dx = \lim_{M\to 0^+} -x^{-1}\Big|_{x=M}^{x=1}$$

is unbounded.

If f is a function that can be written in the form

$$f(x) = \frac{g(x)}{(x-a)^p},$$

where $0 < p < 1$ and g is continuous on $[a, b]$, then the improper integral

$$\int_a^b f(x)\,dx$$

also exists. We will approximate this integral using the Composite Simpson's rule, provided that $g \in C^5[a, b]$. In that case, we can construct the fourth Taylor polynomial, $P_4(x)$, for g about a,

$$P_4(x) = g(a) + g'(a)(x-a) + \frac{g''(a)}{2!}(x-a)^2 + \frac{g'''(a)}{3!}(x-a)^3 + \frac{g^{(4)}(a)}{4!}(x-a)^4,$$

and write

$$\int_a^b f(x)\,dx = \int_a^b \frac{g(x) - P_4(x)}{(x-a)^p}\,dx + \int_a^b \frac{P_4(x)}{(x-a)^p}\,dx. \tag{4.44}$$

Because $P(x)$ is a polynomial, we can exactly determine the value of

$$\int_a^b \frac{P_4(x)}{(x-a)^p}\,dx = \sum_{k=0}^{4} \int_a^b \frac{g^{(k)}(a)}{k!}(x-a)^{k-p}\,dx = \sum_{k=0}^{4} \frac{g^{(k)}(a)}{k!(k+1-p)}(b-a)^{k+1-p}. \tag{4.45}$$

This is generally the dominant portion of the approximation, especially when the Taylor polynomial $P_4(x)$ agrees closely with $g(x)$ throughout the interval $[a, b]$.

To approximate the integral of f, we must add to this value the approximation of

$$\int_a^b \frac{g(x) - P_4(x)}{(x-a)^p}\,dx.$$

To determine this, we first define

$$G(x) = \begin{cases} \frac{g(x)-P_4(x)}{(x-a)^p}, & \text{if } a < x \le b, \\ 0, & \text{if } x = a. \end{cases}$$

■

This gives us a continuous function on $[a, b]$. In fact, $0 < p < 1$ and $P_4^{(k)}(a)$ agrees with $g^{(k)}(a)$ for each $k = 0, 1, 2, 3, 4$, so we have $G \in C^4[a, b]$. This implies that the Composite Simpson's rule can be applied to approximate the integral of G on $[a, b]$. Adding this approximation to the value in Eq. (4.45) gives an approximation to the improper integral of f on $[a, b]$, within the accuracy of the Composite Simpson's rule approximation.

Example 2 Use Composite Simpson's rule with $h = 0.25$ to approximate the value of the improper integral

$$\int_0^1 \frac{e^x}{\sqrt{x}}\,dx.$$

Solution The fourth Taylor polynomial for e^x about $x = 0$ is

$$P_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24},$$

so the dominant portion of the approximation to $\int_0^1 \frac{e^x}{\sqrt{x}}\,dx$ is

$$\int_0^1 \frac{P_4(x)}{\sqrt{x}}\,dx = \int_0^1 \left(x^{-1/2} + x^{1/2} + \frac{1}{2}x^{3/2} + \frac{1}{6}x^{5/2} + \frac{1}{24}x^{7/2}\right) dx$$
$$= \lim_{M\to 0^+} \left[2x^{1/2} + \frac{2}{3}x^{3/2} + \frac{1}{5}x^{5/2} + \frac{1}{21}x^{7/2} + \frac{1}{108}x^{9/2}\right]_M^1$$
$$= 2 + \frac{2}{3} + \frac{1}{5} + \frac{1}{21} + \frac{1}{108} \approx 2.9235450.$$

For the second portion of the approximation to $\int_0^1 \frac{e^x}{\sqrt{x}}\,dx$ we need to approximate $\int_0^1 G(x)\,dx$, where

$$G(x) = \begin{cases} \frac{1}{\sqrt{x}}\,(e^x - P_4(x)), & \text{if } 0 < x \le 1, \\ 0, & \text{if } x = 0. \end{cases}$$

Table 4.13

x	$G(x)$
0.00	0
0.25	0.0000170
0.50	0.0004013
0.75	0.0026026
1.00	0.0099485

Table 4.13 lists the values needed for the Composite Simpson's rule for this approximation.

Using these data and the Composite Simpson's rule gives

$$\int_0^1 G(x)\,dx \approx \frac{0.25}{3}[0 + 4(0.0000170) + 2(0.0004013) + 4(0.0026026) + 0.0099485]$$
$$= 0.0017691.$$

Hence

$$\int_0^1 \frac{e^x}{\sqrt{x}}\,dx \approx 2.9235450 + 0.0017691 = 2.9253141.$$

This result is accurate to within the accuracy of the Composite Simpson's rule approximation for the function G. Because $|G^{(4)}(x)| < 1$ on $[0, 1]$, the error is bounded by

$$\frac{1-0}{180}(0.25)^4 = 0.0000217. \qquad \blacksquare$$

Right Endpoint Singularity

To approximate the improper integral with a singularity at the right endpoint, we could develop a similar technique but expand in terms of the right endpoint b instead of the left endpoint a. Alternatively, we can make the substitution

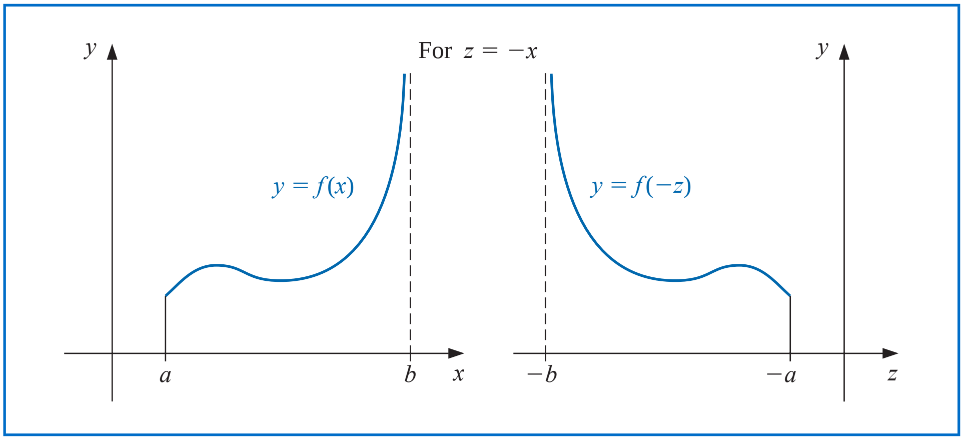
$$z = -x, \qquad dz = -\,dx$$

to change the improper integral into one of the form

$$\int_a^b f(x)\,dx = \int_{-b}^{-a} f(-z)\,dz, \tag{4.46}$$

which has its singularity at the left endpoint. Then we can apply the left endpoint singularity technique we have already developed. (See Figure 4.26.)

Figure 4.26



An improper integral with a singularity at c, where $a < c < b$, is treated as the sum of improper integrals with endpoint singularities since

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

Infinite Singularity

The other type of improper integral involves infinite limits of integration. The basic integral of this type has the form

$$\int_a^\infty \frac{1}{x^p}\,dx,$$

for $p > 1$. This is converted to an integral with left endpoint singularity at 0 by making the integration substitution

$$t = x^{-1}, \quad dt = -x^{-2}\,dx, \quad \text{so} \quad dx = -x^2\,dt = -t^{-2}\,dt.$$

Then

$$\int_a^\infty \frac{1}{x^p}\,dx = \int_{1/a}^0 -\frac{t^p}{t^2}\,dt = \int_0^{1/a} \frac{1}{t^{2-p}}\,dt.$$

In a similar manner, the variable change $t = x^{-1}$ converts the improper integral $\int_a^\infty f(x)\,dx$ into one that has a left endpoint singularity at zero:

$$\int_a^\infty f(x)\,dx = \int_0^{1/a} t^{-2} f\left(\frac{1}{t}\right)\,dt. \tag{4.47}$$

It can now be approximated using a quadrature formula of the type described earlier.

Example 3 Approximate the value of the improper integral

$$I = \int_1^\infty x^{-3/2} \sin\frac{1}{x}\,dx.$$

Solution We first make the variable change $t = x^{-1}$, which converts the infinite singularity into one with a left endpoint singularity. Then

$$dt = -x^{-2}\,dx, \quad \text{so} \quad dx = -x^2\,dt = -\frac{1}{t^2}\,dt,$$

and

$$I = \int_{x=1}^{x=\infty} x^{-3/2} \sin\frac{1}{x}\,dx = \int_{t=1}^{t=0} \left(\frac{1}{t}\right)^{-3/2} \sin t \left(-\frac{1}{t^2}\,dt\right) = \int_0^1 t^{-1/2} \sin t\,dt.$$

The fourth Taylor polynomial, $P_4(t)$, for $\sin t$ about 0 is

$$P_4(t) = t - \frac{1}{6}t^3,$$

so

$$G(t) = \begin{cases} \dfrac{\sin t - t + \frac{1}{6}t^3}{t^{1/2}}, & \text{if } 0 < t \le 1 \\ 0, & \text{if } t = 0 \end{cases}$$

is in $C^4[0, 1]$, and we have

$$\begin{aligned} I &= \int_0^1 t^{-1/2}\left(t - \frac{1}{6}t^3\right) dt + \int_0^1 \frac{\sin t - t + \frac{1}{6}t^3}{t^{1/2}}\,dt \\ &= \left[\frac{2}{3}t^{3/2} - \frac{1}{21}t^{7/2}\right]_0^1 + \int_0^1 \frac{\sin t - t + \frac{1}{6}t^3}{t^{1/2}}\,dt \\ &= 0.61904761 + \int_0^1 \frac{\sin t - t + \frac{1}{6}t^3}{t^{1/2}}\,dt. \end{aligned}$$

The result from the Composite Simpson's rule with $n = 16$ for the remaining integral is 0.0014890097. This gives a final approximation of

$$I = 0.0014890097 + 0.61904761 = 0.62053661,$$

which is accurate to within 4.0×10^{-8}. ■

EXERCISE SET 4.9

1. Use Simpson's Composite rule and the given values of n to approximate the following improper integrals.

   a. $\displaystyle\int_0^1 x^{-1/4} \sin x\,dx, \quad n = 4$

   b. $\displaystyle\int_0^1 \frac{e^{2x}}{\sqrt[5]{x^2}}\,dx, \quad n = 6$

   c. $\displaystyle\int_1^2 \frac{\ln x}{(x-1)^{1/5}}\,dx, \quad n = 8$

   d. $\displaystyle\int_0^1 \frac{\cos 2x}{x^{1/3}}\,dx, \quad n = 6$

2. Use the Composite Simpson's rule and the given values of n to approximate the following improper integrals.

a. $\displaystyle\int_0^1 \frac{e^{-x}}{\sqrt{1-x}}\,dx, \quad n=6$ **b.** $\displaystyle\int_0^2 \frac{xe^x}{\sqrt[3]{(x-1)^2}}\,dx, \quad n=8$

3. Use the transformation $t = x^{-1}$ and then the Composite Simpson's rule and the given values of n to approximate the following improper integrals.

a. $\displaystyle\int_1^\infty \frac{1}{x^2+9}\,dx, \quad n=4$ **b.** $\displaystyle\int_1^\infty \frac{1}{1+x^4}\,dx, \quad n=4$

c. $\displaystyle\int_1^\infty \frac{\cos x}{x^3}\,dx, \quad n=6$ **d.** $\displaystyle\int_1^\infty x^{-4}\sin x\,dx, \quad n=6$

4. The improper integral $\int_0^\infty f(x)\,dx$ cannot be converted into an integral with finite limits using the substitution $t = 1/x$ because the limit at zero becomes infinite. The problem is resolved by first writing $\int_0^\infty f(x)\,dx = \int_0^1 f(x)\,dx + \int_1^\infty f(x)\,dx$. Apply this technique to approximate the following improper integrals to within 10^{-6}.

a. $\displaystyle\int_0^\infty \frac{1}{1+x^4}\,dx$ **b.** $\displaystyle\int_0^\infty \frac{1}{(1+x^2)^3}\,dx$

5. Suppose a body of mass m is traveling vertically upward starting at the surface of the earth. If all resistance except gravity is neglected, the escape velocity v is given by

$$v^2 = 2gR\int_1^\infty z^{-2}\,dz, \quad \text{where } z = \frac{x}{R},$$

$R = 3960$ miles is the radius of the earth, and $g = 0.00609$ mi/s^2 is the force of gravity at the earth's surface. Approximate the escape velocity v.

6. The Laguerre polynomials $\{L_0(x), L_1(x)\ldots\}$ form an orthogonal set on $[0,\infty)$ and satisfy $\int_0^\infty e^{-x}L_i(x)L_j(x)\,dx = 0$, for $i \neq j$. (See Section 8.2.) The polynomial $L_n(x)$ has n distinct zeros $x_1, x_2, \ldots, x_n$ in $[0,\infty)$. Let

$$c_{n,i} = \int_0^\infty e^{-x}\prod_{\substack{j=1\\ j\neq i}}^{n}\frac{x-x_j}{x_i-x_j}\,dx.$$

Show that the quadrature formula

$$\int_0^\infty f(x)e^{-x}\,dx = \sum_{i=1}^{n} c_{n,i}f(x_i)$$

has degree of precision $2n-1$. (Hint: Follow the steps in the proof of Theorem 4.7.)

7. The Laguerre polynomials $L_0(x) = 1$, $L_1(x) = 1-x$, $L_2(x) = x^2-4x+2$, and $L_3(x) = -x^3 + 9x^2 - 18x + 6$ are derived in Exercise 11 of Section 8.2. As shown in Exercise 6, these polynomials are useful in approximating integrals of the form

$$\int_0^\infty e^{-x}f(x)\,dx = 0.$$

a. Derive the quadrature formula using $n = 2$ and the zeros of $L_2(x)$.

b. Derive the quadrature formula using $n = 3$ and the zeros of $L_3(x)$.

8. Use the quadrature formulas derived in Exercise 7 to approximate the integral

$$\int_0^\infty \sqrt{x}e^{-x}\,dx.$$

9. Use the quadrature formulas derived in Exercise 7 to approximate the integral

$$\int_{-\infty}^\infty \frac{1}{1+x^2}\,dx.$$

4.10 Survey of Methods and Software

In this chapter we considered approximating integrals of functions of one, two, or three variables, and approximating the derivatives of a function of a single real variable.

The Midpoint rule, Trapezoidal rule, and Simpson's rule were studied to introduce the techniques and error analysis of quadrature methods. Composite Simpson's rule is easy to use and produces accurate approximations unless the function oscillates in a subinterval of the interval of integration. Adaptive quadrature can be used if the function is suspected of oscillatory behavior. To minimize the number of nodes while maintaining accuracy, we used Gaussian quadrature. Romberg integration was introduced to take advantage of the easily applied Composite Trapezoidal rule and extrapolation.

Most software for integrating a function of a single real variable is based either on the adaptive approach or extremely accurate Gaussian formulas. Cautious Romberg integration is an adaptive technique that includes a check to make sure that the integrand is smoothly behaved over subintervals of the integral of integration. This method has been successfully used in software libraries. Multiple integrals are generally approximated by extending good adaptive methods to higher dimensions. Gaussian-type quadrature is also recommended to decrease the number of function evaluations.

The main routines in both the IMSL and NAG Libraries are based on *QUADPACK: A Subroutine Package for Automatic Integration* by R. Piessens, E. de Doncker-Kapenga, C. W. Uberhuber, and D. K. Kahaner published by Springer-Verlag in 1983 [PDUK].

The IMSL Library contains an adaptive integration scheme based on the 21-point Gaussian-Kronrod rule using the 10-point Gaussian rule for error estimation. The Gaussian rule uses the ten points $x_1, \ldots, x_{10}$ and weights $w_1, \ldots, w_{10}$ to give the quadrature formula $\sum_{i=1}^{10} w_i f(x_i)$ to approximate $\int_a^b f(x)\, dx$. The additional points $x_{11}, \ldots, x_{21}$, and the new weights $v_1, \ldots, v_{21}$, are then used in the Kronrod formula $\sum_{i=1}^{21} v_i f(x_i)$. The results of the two formulas are compared to eliminate error. The advantage in using $x_1, \ldots, x_{10}$ in each formula is that f needs to be evaluated only at 21 points. If independent 10- and 21-point Gaussian rules were used, 31 function evaluations would be needed. This procedure permits endpoint singularities in the integrand.

Other IMSL subroutines allow for endpoint singularities, user-specified singularities, and infinite intervals of integration. In addition, there are routines for applying Gauss-Kronrod rules to integrate a function of two variables, and a routine to use Gaussian quadrature to integrate a function of n variables over n intervals of the form $[a_i, b_i]$.

The NAG Library includes a routine to compute the integral of f over the interval $[a, b]$ using an adaptive method based on Gaussian Quadrature using Gauss 10-point and Kronrod 21-point rules. It also has a routine to approximate an integral using a family of Gaussian-type formulas based on 1, 3, 5, 7, 15, 31, 63, 127, and 255 nodes. These interlacing high-precision rules are due to Patterson [Pat] and are used in an adaptive manner. NAG includes many other subroutines for approximating integrals.

MATLAB has a routine to approximate a definite integral using an adaptive Simpson's rule, and another to approximate the definite integral using an adaptive eight-panel Newton-Cotes rule.

Although numerical differentiation is unstable, derivative approximation formulas are needed for solving differential equations. The NAG Library includes a subroutine for the numerical differentiation of a function of one real variable with differentiation to the fourteenth derivative being possible. IMSL has a function that uses an adaptive change in step size for finite differences to approximate the first, second, or third, derivative of f at x to within a given tolerance. IMSL also includes a subroutine to compute the derivatives of a function defined on a set of points using quadratic interpolation. Both packages allow the differentiation and integration of interpolatory cubic splines constructed by the subroutines mentioned in Section 3.5.

For further reading on numerical integration we recommend the books by Engels [E] and by Davis and Rabinowitz [DR]. For more information on Gaussian quadrature see Stroud and Secrest [StS]. Books on multiple integrals include those by Stroud [Stro] and by Sloan and Joe [SJ].