CHAPTER

3 Interpolation and Polynomial Approximation

3.1 Introduction

Engineers and scientists commonly assume that relationships between variables in a physical problem can be approximately reproduced from data given by the problem. The ultimate goal might be to determine the values at intermediate points, to approximate the integral or derivative of the underlying function, or to simply give a smooth or continuous representation of the variables in the problem.

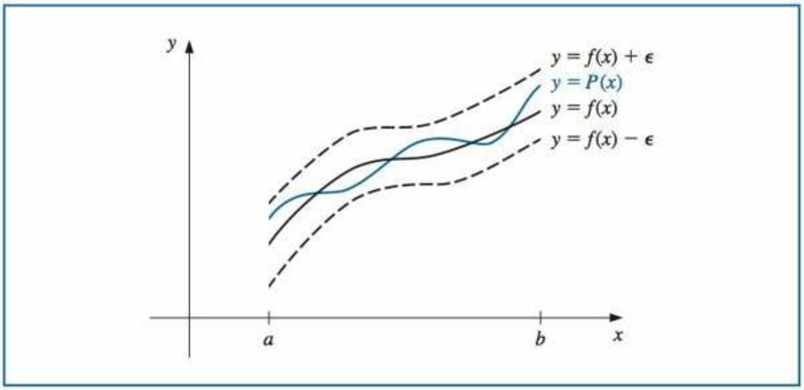
Interpolation refers to determining a function that exactly represents a collection of data. The most elementary type of interpolation consists of fitting a polynomial to a collection of data points. Polynomials have derivatives and integrals that are themselves polynomials, so they are a natural choice for approximating derivatives and integrals. In this chapter we will see that polynomials to approximate continuous functions are easily constructed. The following result implies that there are polynomials that are arbitrarily close to any continuous function.

Weierstrass Approximation Theorem

Suppose that f is defined and continuous on $[a, b]$. For each $\varepsilon > 0$, there exists a polynomial $P(x)$ defined on $[a, b]$, with the property that (see Figure 3.1)

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

Figure 3.1



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.

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.

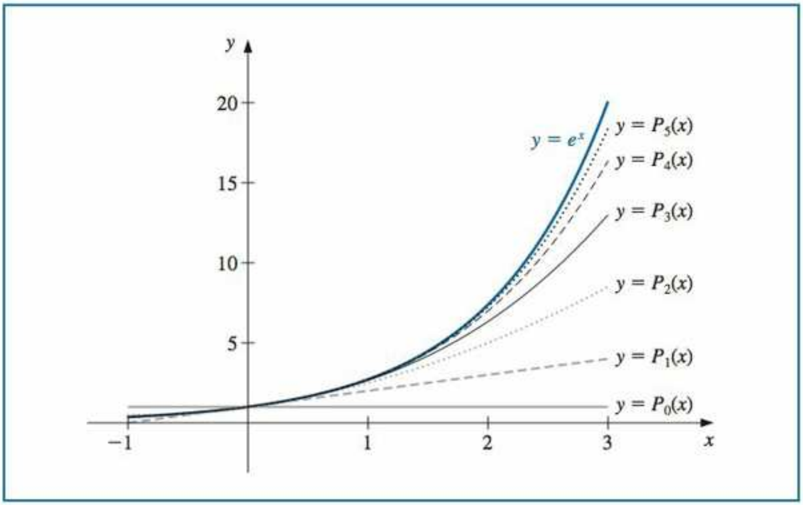
The Taylor polynomials were introduced in Chapter 1, where they were described as one of the fundamental building blocks of numerical analysis. Given this prominence, you might assume that polynomial interpolation makes heavy use of these functions. However, this is not the case. The Taylor polynomials agree as closely as possible with a given function at a specific point, but they concentrate their accuracy only near that point. A good interpolation polynomial needs to provide a relatively accurate approximation over an entire interval, and Taylor polynomials do not do that. 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

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

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

The graphs of these Taylor polynomials are shown in Figure 3.2. Notice that the error becomes progressively worse as we move away from zero.

Figure 3.2



Although better approximations are obtained for this problem if higher-degree Taylor polynomials are used, this situation is not always true. Consider, as an extreme example, using Taylor polynomials of various degrees for $f(x) = 1/x$ expanded about $x_0 = 1$ to approximate $f(3) = \frac{1}{3}$. Since

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

and, in general,

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

the Taylor polynomials for $n \geq 0$ are

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

When we approximate $f(3) = \frac{1}{3}$ by $P_n(3)$ for larger values of n, the approximations become increasingly inaccurate, as shown Table 3.1.

Table 3.1

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

The Taylor polynomials have the property that all the information used in the approximation is concentrated at the single point x_0, so it is not uncommon for these polynomials to 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 points close to x_0. For ordinary computational purposes, it is more efficient to use methods that include information at various points, which we will consider in the remainder of this chapter. The primary use of Taylor polynomials in numerical analysis is *not* for approximation purposes; instead it is for the derivation of numerical techniques.

3.2 Lagrange Polynomials

The interpolation formula named for Joseph Louis Lagrange (1736–1813) was likely known by Isaac Newton around 1675, but it appears to have been published first 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.

In the previous section we discussed the general unsuitability of Taylor polynomials for approximation. These polynomials are useful only over small intervals for functions whose derivatives exist and are easily evaluated. In this section we find approximating polynomials that can be determined simply by specifying certain points on the plane through which they must pass.

Lagrange Interpolating Polynomials

Determining a polynomial of degree 1 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. We first define the functions

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

and note that these definitions imply that

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

We then define

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

This gives

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

and

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

So, P is the unique linear function passing 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

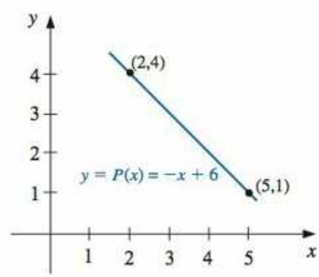
$$L_0(x) = \frac{x-5}{2-5} = -\frac{1}{3}(x-5) \quad \text{and} \quad L_1(x) = \frac{x-2}{5-2} = \frac{1}{3}(x-2),$$

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. ■

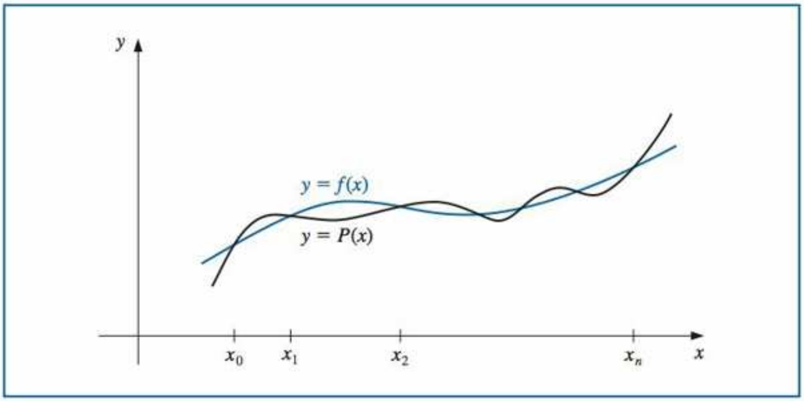
Figure 3.3



To generalize the concept of linear interpolation to higher-degree polynomials, consider the construction of a polynomial of degree at most n, shown in Figure 3.4, that passes through the $n + 1$ points

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

Figure 3.4



In this case, we construct, for each $k = 0, 1, \ldots, n$, a polynomial of degree n, which we will denote by $L_{n,k}(x)$, 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 $i \neq k$, the numerator of $L_{n,k}(x)$ must contain the term

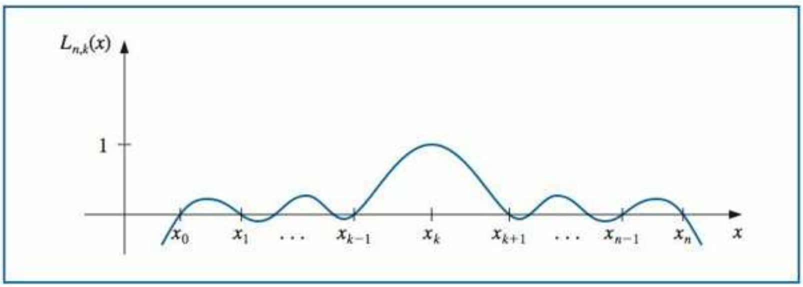
$$(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n).$$

To satisfy $L_{n,k}(x_k) = 1$, the denominator of $L_{n,k}(x)$ must be this term 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 now that the form of $L_{n,k}(x)$ is known. This polynomial is called the nth Lagrange interpolating polynomial.

*n*th Lagrange Interpolating Polynomial

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

where

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

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

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 $P_n(x)$ is the unique polynomial of degree at most n that agrees with $f(x)$ at $x_0, x_1, \ldots, x_n$. The notation for describing the Lagrange interpolating polynomial $P_n(x)$ is rather complicated because $P_n(x)$ is the sum of the $n+1$ polynomials $f(x_k)L_{n,k}(x)$, for $k = 0, 1, \ldots, n$, each of which is of degree n, provided $f(x_k) \neq 0$. To reduce the notational complication, we will write $L_{n,k}(x)$ simply as $L_k(x)$ when there should be no confusion that its degree is n.

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 $L_2(x)$. They are

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

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

and

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

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

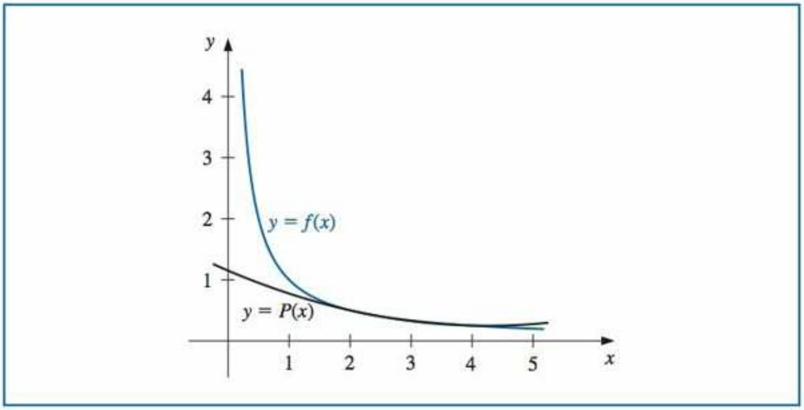
$$\begin{aligned} P(x) &= \sum_{k=0}^{2} f(x_k)L_k(x) \\ &= \frac{1}{3}(x-2.75)(x-4) - \frac{64}{165}(x-2)(x-4) + \frac{1}{10}(x-2)(x-2.75) \\ &= \frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44}. \end{aligned}$$

(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 Section 3.1 (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 Lagrange polynomials have remainder terms that are reminiscent of those for the Taylor polynomials. The nth Taylor polynomial about x_0 concentrates all the known information at x_0 and has an error term of the form

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

where $\xi(x)$ is between x and x_0. The nth Lagrange polynomial uses information at the distinct numbers $x_0, x_1, \ldots, x_n$. In place of $(x-x_0)^{n+1}$, its error formula uses a product of the $n+1$ terms $(x-x_0)$, $(x-x_1)$, $\ldots$, $(x-x_n)$, and the number $\xi(x)$ can lie anywhere in the interval that contains the points $x_0, x_1, \ldots, x_n$, and x. Otherwise it has the same form as the error formula for the Taylor polynomials.

Lagrange Polynomial Error Formula

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

for some (unknown) number $\xi(x)$ that lies in the smallest interval that contains $x_0, x_1, \ldots, x_n$ and x.

This error formula 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 polynomial error formula. The specific use of this error formula, however, is restricted to those functions whose derivatives have known bounds. The next Illustration shows interpolation techniques for a situation in which the Lagrange error formula cannot be used. This shows that we should look for a more efficient way to obtain approximations via interpolation.

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 to 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

$$P_1(1.5) = \frac{(1.5-1.6)}{(1.3-1.6)}f(1.3) + \frac{(1.5-1.3)}{(1.6-1.3)}f(1.6)$$

$$= \frac{(1.5-1.6)}{(1.3-1.6)}(0.6200860) + \frac{(1.5-1.3)}{(1.6-1.3)}(0.4554022) = 0.5102968.$$

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 because 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 because it includes the most data about the function. The Lagrange error term 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 because the error term is difficult to apply, the degree of the polynomial needed for the desired accuracy is generally not known until the computations are determined. The usual practice is to compute the results given from various polynomials until appropriate agreement is obtained, as was done in the previous example. 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. To derive these approximating polynomials in a manner that uses the previous calculations to advantage, we need to introduce some new notation.

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 $0 \leq m_i \leq n$ 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 3 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

$$\begin{aligned} f(5) \approx P_{1,2,4}(5) &= \frac{(5-3)(5-6)}{(2-3)(2-6)}e^2 + \frac{(5-2)(5-6)}{(3-2)(3-6)}e^3 + \frac{(5-2)(5-3)}{(6-2)(6-3)}e^6 \\ &= -\frac{1}{2}e^2 + e^3 + \frac{1}{2}e^6 \approx 218.105. \end{aligned}$$

■

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

Recursively Generated Lagrange Polynomials

Let f be defined at $x_0, x_1, \ldots, x_k$ and x_j, x_i be two numbers in this set. If

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

then $P(x)$ is the kth Lagrange polynomial that interpolates, or agrees with, $f(x)$ at the $k+1$ points $x_0, x_1, \ldots, x_k$.

To see why this recursive formula is true, first 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 at most $k-1$,

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

must be of degree at most k. If $0 \leq r \leq k$ with $r \neq i$ and $r \neq j$, then $Q(x_r) = \hat{Q}(x_r) = f(x_r)$, so

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

Moreover,

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

and similarly, $P(x_j) = f(x_j)$. But there is only one polynomial of degree at most k that agrees with $f(x)$ at $x_0, x_1, \ldots, x_k$, and this polynomial by definition is $P_{0,1,\ldots,k}(x)$. Hence,

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

This result implies that the approximations from the interpolating polynomials can be generated recursively in the manner shown in Table 3.3. The row-by-row generation is performed to move across the rows as rapidly as possible, because these entries are given by successively higher-degree interpolating polynomials. This procedure is called **Neville's method**.

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	$P_{3,4}$	$P_{2,3,4}$	$P_{1,2,3,4}$	$P_{0,1,2,3,4}$

Program NEVLLE31 implements the Neville's method.

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

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. To avoid the cumbersome subscripts we let $Q_{i,j}(x)$, for $0 \le j \le i$, denote the jth interpolating polynomial 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 for Neville's method provides the Q notation 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 4 Table 3.5 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 3.5

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

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

The final approximation we can obtain from this data is

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

These values are shown in Table 3.6. ■

Table 3.6

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

If the latest approximation, $Q_{2,2}$, is not as accurate as desired, another node, x_3, can be selected and another row can be added to the table:

$$x_3 \quad Q_{3,0} \quad Q_{3,1} \quad Q_{3,2} \quad Q_{3,3}.$$

Then $Q_{2,2}$, $Q_{3,2}$, and $Q_{3,3}$ can be compared to determine further accuracy.

Using $x_3 = 2.4$ in Example 4 gives no improvement of accuracy because the additional row is

$$2.4 \quad 0.8755 \quad 0.7480 \quad 0.7420 \quad 0.7420.$$

Had the data been given with more digits of accuracy, there might have been an improvement.

EXERCISE SET 3.2

1. For the given functions $f(x)$, let $x_0 = 0$, $x_1 = 0.6$, and $x_2 = 0.9$. Construct the Lagrange interpolating polynomials of degree (i) at most 1 and (ii) at most 2 to approximate $f(0.45)$, and find the actual error.
   a. $f(x) = \cos x$
   b. $f(x) = \sqrt{1+x}$
   c. $f(x) = \ln(x+1)$
   d. $f(x) = \tan x$
2. Use the Lagrange polynomial error formula to find an error bound for the approximations in Exercise 1.
3. Use appropriate Lagrange interpolating polynomials of degrees 1, 2, and 3 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 $f(0.6) = -0.17694460$, $f(0.7) = 0.01375227$, $f(0.8) = 0.22363362$, $f(1.0) = 0.65809197$
4. Use Neville's method to obtain the approximations for Exercise 3.
5. Use Neville's method to approximate $\sqrt{3}$ with the function $f(x) = 3^x$ and the values $x_0 = -2$, $x_1 = -1$, $x_2 = 0$, $x_3 = 1$, and $x_4 = 2$.

6. Use Neville's method to approximate $\sqrt{3}$ with the function $f(x) = \sqrt{x}$ and the values $x_0 = 0, x_1 = 1, x_2 = 2, x_3 = 4$, and $x_4 = 5$. Compare the accuracy with that of Exercise 5.

7. The data for Exercise 3 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. Use the Lagrange interpolating polynomial of degree 3 or less and four-digit chopping arithmetic to approximate cos 0.750 using the following values. Find an error bound for the approximation.

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

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.

9. 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.

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

10. Repeat Exercise 9 using MATLAB in `long format` mode.

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

12. 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 $P_2 = f(0.7)$.

13. 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?

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

15. Neville's method 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 $f(0)$.

16. The following table lists the population of the United States from 1960 to 2010.

Year	1960	1970	1980	1990	2000	2010
Population (thousands)	179,323	203,302	226,542	249,633	281,442	307,746

   - **a.** Find the Lagrange polynomial of degree 5 fitting this data, and use this polynomial to estimate the population in the years 1950, 1975, and 2020.
   - **b.** The population in 1950 was approximately 151,326,000. How accurate do you think your 1975 and 2020 figures are?

17. In Exercise 15 of Section 1.2, 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 more feasible?

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.

Divided Differences

We first need to introduce the divided-difference notation, which should remind you of the Aitken's Δ^2 notation defined on page 53. Suppose we are given the $n+1$ points $(x_0, f(x_0))$, $(x_1, f(x_1)), \ldots (x_n, f(x_n))$. There are $n+1$ **zeroth divided differences** of the function f. For each $i = 0, 1, \ldots, n$ we define $f[x_i]$ simply as the value of f at x_i:

$$f[x_i] = f(x_i).$$

The remaining divided differences are defined inductively. There are n **first divided differences** of f, one for each $i = 0, 1, \ldots, n-1$. The first divided difference relative to x_i and x_{i+1} is denoted $f[x_i, x_{i+1}]$ and is defined by

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

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 ***k*th divided difference** relative to $x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+k}$ is defined by

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

The process ends with the single nth divided difference,

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

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.

With this notation, it can be shown that the nth Lagrange interpolation polynomial for f with respect to $x_0, x_1, \ldots, x_n$ can be expressed as

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

which is called *Newton's divided-difference formula.* In compressed form we have the following.