real numbers. If the initial approximation is a real number, all subsequent approximations will also be real numbers. One way to overcome this difficulty is to begin with a complex initial approximation and do all the computations using complex arithmetic. An alternative approach has its basis in the following theorem.

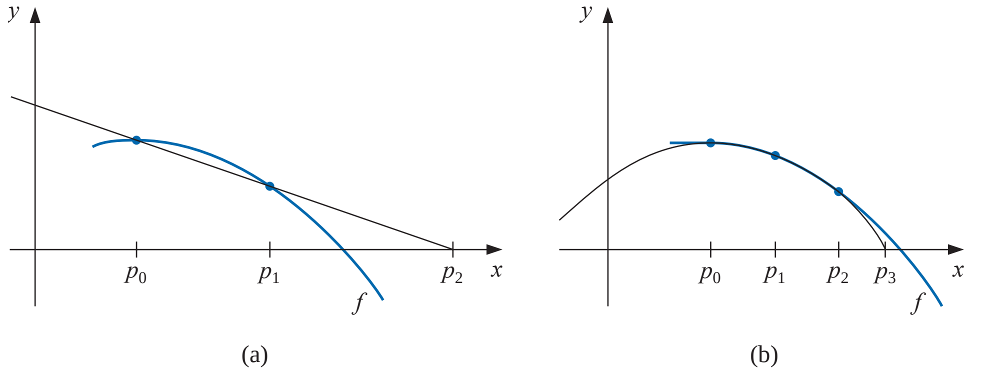
Theorem 2.20 If $z = a+bi$ is a complex zero of multiplicity m of the polynomial $P(x)$ with real coefficients, then $\bar{z} = a - bi$ is also a zero of multiplicity m of the polynomial $P(x)$, and $(x^2 - 2ax + a^2 + b^2)^m$ is a factor of $P(x)$. ■

Müller's method is similar to the Secant method. But whereas the Secant method uses a line through two points on the curve to approximate the root, Müller's method uses a parabola through three points on the curve for the approximation.

A synthetic division involving quadratic polynomials can be devised to approximately factor the polynomial so that one term will be a quadratic polynomial whose complex roots are approximations to the roots of the original polynomial. This technique was described in some detail in our second edition [BFR]. Instead of proceeding along these lines, we will now consider a method first presented by D. E. Müller [Mu]. This technique can be used for any root-finding problem, but it is particularly useful for approximating the roots of polynomials.

The Secant method begins with two initial approximations p_0 and p_1 and determines the next approximation p_2 as the intersection of the x-axis with the line through $(p_0, f(p_0))$ and $(p_1, f(p_1))$. (See Figure 2.13(a).) Müller's method uses three initial approximations, p_0, p_1, and p_2, and determines the next approximation p_3 by considering the intersection of the x-axis with the parabola through $(p_0, f(p_0))$, $(p_1, f(p_1))$, and $(p_2, f(p_2))$. (See Figure 2.13(b).)

Figure 2.13



The derivation of Müller's method begins by considering the quadratic polynomial

$$P(x) = a(x - p_2)^2 + b(x - p_2) + c$$

that passes through $(p_0, f(p_0))$, $(p_1, f(p_1))$, and $(p_2, f(p_2))$. The constants a, b, and c can be determined from the conditions

$$f(p_0) = a(p_0 - p_2)^2 + b(p_0 - p_2) + c, \tag{2.17}$$

$$f(p_1) = a(p_1 - p_2)^2 + b(p_1 - p_2) + c, \tag{2.18}$$

and

$$f(p_2) = a \cdot 0^2 + b \cdot 0 + c = c \tag{2.19}$$

to be

$$c = f(p_2), \tag{2.20}$$

$$b = \frac{(p_0 - p_2)^2[f(p_1) - f(p_2)] - (p_1 - p_2)^2[f(p_0) - f(p_2)]}{(p_0 - p_2)(p_1 - p_2)(p_0 - p_1)}, \tag{2.21}$$

and

$$a = \frac{(p_1 - p_2)[f(p_0) - f(p_2)] - (p_0 - p_2)[f(p_1) - f(p_2)]}{(p_0 - p_2)(p_1 - p_2)(p_0 - p_1)}. \tag{2.22}$$

To determine p_3, a zero of P, we apply the quadratic formula to $P(x) = 0$. However, because of round-off error problems caused by the subtraction of nearly equal numbers, we apply the formula in the manner prescribed in Eq (1.2) and (1.3) of Section 1.2:

$$p_3 - p_2 = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}.$$

This formula gives two possibilities for p_3, depending on the sign preceding the radical term. In Müller's method, the sign is chosen to agree with the sign of b. Chosen in this manner, the denominator will be the largest in magnitude and will result in p_3 being selected as the closest zero of P to p_2. Thus

$$p_3 = p_2 - \frac{2c}{b + \text{sgn}(b)\sqrt{b^2 - 4ac}},$$

where a, b, and c are given in Eqs. (2.20) through (2.22).

Once p_3 is determined, the procedure is reinitialized using p_1, p_2, and p_3 in place of p_0, p_1, and p_2 to determine the next approximation, p_4. The method continues until a satisfactory conclusion is obtained. At each step, the method involves the radical $\sqrt{b^2 - 4ac}$, so the method gives approximate complex roots when $b^2 - 4ac < 0$. Algorithm 2.8 implements this procedure.

ALGORITHM 2.8

Müller's

To find a solution to $f(x) = 0$ given three approximations, p_0, p_1, and p_2:

INPUT p_0, p_1, p_2; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $h_1 = p_1 - p_0$;
$h_2 = p_2 - p_1$;
$\delta_1 = (f(p_1) - f(p_0))/h_1$;
$\delta_2 = (f(p_2) - f(p_1))/h_2$;
$d = (\delta_2 - \delta_1)/(h_2 + h_1)$;
$i = 3$.

Step 2 While $i \le N_0$ do Steps 3–7.

Step 3 $b = \delta_2 + h_2 d$;
$D = (b^2 - 4f(p_2)d)^{1/2}$. (*Note: May require complex arithmetic.*)

Step 4 If $|b - D| < |b + D|$ then set $E = b + D$
else set $E = b - D$.

Step 5 Set $h = -2f(p_2)/E$;
$p = p_2 + h$.

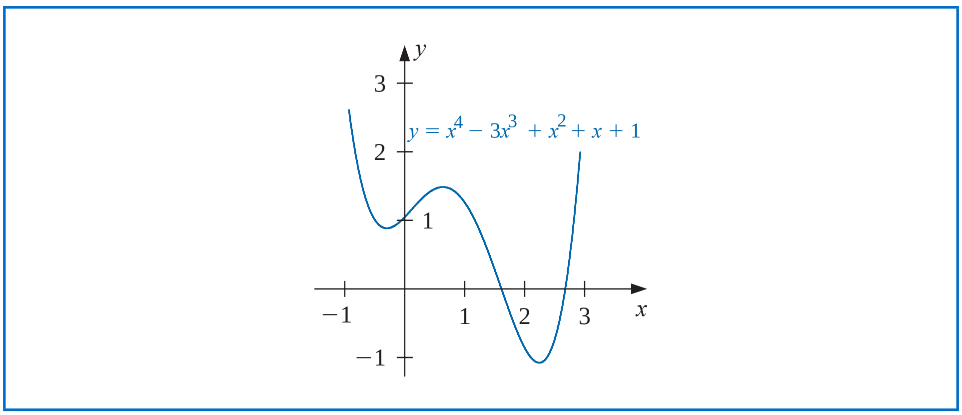
Step 6 If $|h| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 7 Set $p_0 = p_1$; (*Prepare for next iteration.*)
$p_1 = p_2$;
$p_2 = p$;
$h_1 = p_1 - p_0$;
$h_2 = p_2 - p_1$;
$\delta_1 = (f(p_1) - f(p_0))/h_1$;
$\delta_2 = (f(p_2) - f(p_1))/h_2$;
$d = (\delta_2 - \delta_1)/(h_2 + h_1)$;
$i = i + 1$.

Step 8 OUTPUT ('Method failed after N_0 iterations, $N_0 =$', N_0);
(*The procedure was unsuccessful.*)
STOP. ■

Illustration Consider the polynomial $f(x) = x^4 - 3x^3 + x^2 + x + 1$, part of whose graph is shown in Figure 2.14.

Figure 2.14



Three sets of three initial points will be used with Algorithm 2.8 and $TOL = 10^{-5}$ to approximate the zeros of f. The first set will use $p_0 = 0.5$, $p_1 = -0.5$, and $p_2 = 0$. The parabola passing through these points has complex roots because it does not intersect the x-axis. Table 2.12 gives approximations to the corresponding complex zeros of f.

Table 2.12

	$p_0 = 0.5,\ p_1 = -0.5,\ p_2 = 0$	
i	p_i	$f(p_i)$
3	$-0.100000 + 0.888819i$	$-0.01120000 + 3.014875548i$
4	$-0.492146 + 0.447031i$	$-0.1691201 - 0.7367331502i$
5	$-0.352226 + 0.484132i$	$-0.1786004 + 0.0181872213i$
6	$-0.340229 + 0.443036i$	$0.01197670 - 0.0105562185i$
7	$-0.339095 + 0.446656i$	$-0.0010550 + 0.000387261i$
8	$-0.339093 + 0.446630i$	$0.000000 + 0.000000i$
9	$-0.339093 + 0.446630i$	$0.000000 + 0.000000i$

Table 2.13 gives the approximations to the two real zeros of f. The smallest of these uses $p_0 = 0.5$, $p_1 = 1.0$, and $p_2 = 1.5$, and the largest root is approximated when $p_0 = 1.5$, $p_1 = 2.0$, and $p_2 = 2.5$.

Table 2.13

$p_0 = 0.5$,	$p_1 = 1.0$,	$p_2 = 1.5$	$p_0 = 1.5$,	$p_1 = 2.0$,	$p_2 = 2.5$
i	p_i	$f(p_i)$	i	p_i	$f(p_i)$
3	1.40637	−0.04851	3	2.24733	−0.24507
4	1.38878	0.00174	4	2.28652	−0.01446
5	1.38939	0.00000	5	2.28878	−0.00012
6	1.38939	0.00000	6	2.28880	0.00000
			7	2.28879	0.00000

The values in the tables are accurate approximations to the places listed. □

We used Maple to generate the results in Table 2.12. To find the first result in the table, define $f(x)$ with

$$f := x \to x^4 - 3x^3 + x^2 + x + 1$$

Then enter the initial approximations with

$$p0 := 0.5; p1 := -0.5; p2 := 0.0$$

and evaluate the function at these points with

$$f0 := f(p0); f1 := f(p1); f2 := f(p2)$$

To determine the coefficients a, b, c, and the approximate solution, enter

$$c := f2;$$

$$b := \frac{\left((p0-p2)^2 \cdot (f1-f2) - (p1-p2)^2 \cdot (f0-f2)\right)}{(p0-p2)\cdot(p1-p2)\cdot(p0-p1)}$$

$$a := \frac{((p1-p2)\cdot(f0-f2) - (p0-p2)\cdot(f1-f2))}{(p0-p2)\cdot(p1-p2)\cdot(p0-p1)}$$

$$p3 := p2 - \frac{2c}{b + \left(\frac{b}{abs(b)}\right)\sqrt{b^2 - 4a\cdot c}}$$

This produces the final Maple output

$$-0.1000000000 + 0.8888194418I$$

and evaluating at this approximation gives $f(p3)$ as

$$-0.0112000001 + 3.014875548I$$

This is our first approximation, as seen in Table 2.12.

The illustration shows that Müller's method can approximate the roots of polynomials with a variety of starting values. In fact, Müller's method generally converges to the root of a polynomial for any initial approximation choice, although problems can be constructed for which convergence will not occur. For example, suppose that for some i we have $f(p_i) = f(p_{i+1}) = f(p_{i+2}) \neq 0$. The quadratic equation then reduces to a nonzero constant function and never intersects the x-axis. This is not usually the case, however, and general-purpose software packages using Müller's method request only one initial approximation per root and will even supply this approximation as an option.

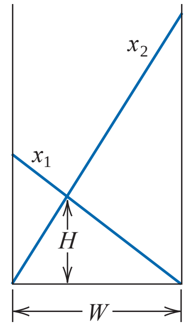
EXERCISE SET 2.6

1. Find the approximations to within 10^{-4} to all the real zeros of the following polynomials using Newton's method.
   - **a.** $f(x) = x^3 - 2x^2 - 5$
   - **b.** $f(x) = x^3 + 3x^2 - 1$
   - **c.** $f(x) = x^3 - x - 1$
   - **d.** $f(x) = x^4 + 2x^2 - x - 3$
   - **e.** $f(x) = x^3 + 4.001x^2 + 4.002x + 1.101$
   - **f.** $f(x) = x^5 - x^4 + 2x^3 - 3x^2 + x - 4$
2. Find approximations to within 10^{-5} to all the zeros of each of the following polynomials by first finding the real zeros using Newton's method and then reducing to polynomials of lower degree to determine any complex zeros.
   - **a.** $f(x) = x^4 + 5x^3 - 9x^2 - 85x - 136$
   - **b.** $f(x) = x^4 - 2x^3 - 12x^2 + 16x - 40$
   - **c.** $f(x) = x^4 + x^3 + 3x^2 + 2x + 2$
   - **d.** $f(x) = x^5 + 11x^4 - 21x^3 - 10x^2 - 21x - 5$
   - **e.** $f(x) = 16x^4 + 88x^3 + 159x^2 + 76x - 240$
   - **f.** $f(x) = x^4 - 4x^2 - 3x + 5$
   - **g.** $f(x) = x^4 - 2x^3 - 4x^2 + 4x + 4$
   - **h.** $f(x) = x^3 - 7x^2 + 14x - 6$
3. Repeat Exercise 1 using Müller's method.
4. Repeat Exercise 2 using Müller's method.
5. Use Newton's method to find, within 10^{-3}, the zeros and critical points of the following functions. Use this information to sketch the graph of f.
   - **a.** $f(x) = x^3 - 9x^2 + 12$
   - **b.** $f(x) = x^4 - 2x^3 - 5x^2 + 12x - 5$
6. $f(x) = 10x^3 - 8.3x^2 + 2.295x - 0.21141 = 0$ has a root at $x = 0.29$. Use Newton's method with an initial approximation $x_0 = 0.28$ to attempt to find this root. Explain what happens.
7. Use Maple to find a real zero of the polynomial $f(x) = x^3 + 4x - 4$.
8. Use Maple to find a real zero of the polynomial $f(x) = x^3 - 2x - 5$.
9. Use each of the following methods to find a solution in $[0.1, 1]$ accurate to within 10^{-4} for

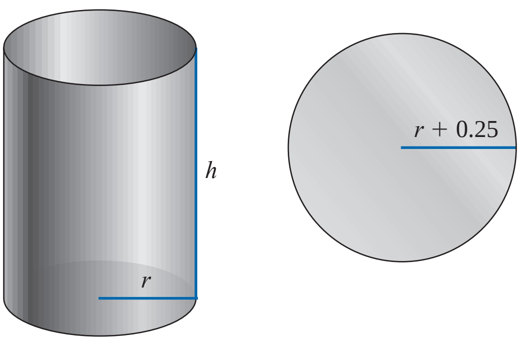
$$600x^4 - 550x^3 + 200x^2 - 20x - 1 = 0.$$

   - **a.** Bisection method
   - **b.** Newton's method
   - **c.** Secant method
   - **d.** method of False Position
   - **e.** Müller's method

10. Two ladders crisscross an alley of width W. Each ladder reaches from the base of one wall to some point on the opposite wall. The ladders cross at a height H above the pavement. Find W given that the lengths of the ladders are $x_1 = 20$ ft and $x_2 = 30$ ft, and that $H = 8$ ft.



11. A can in the shape of a right circular cylinder is to be constructed to contain 1000 cm^3. The circular top and bottom of the can must have a radius of 0.25 cm more than the radius of the can so that the excess can be used to form a seal with the side. The sheet of material being formed into the side of the can must also be 0.25 cm longer than the circumference of the can so that a seal can be formed. Find, to within 10^{-4}, the minimal amount of material needed to construct the can.



12. In 1224, Leonardo of Pisa, better known as Fibonacci, answered a mathematical challenge of John of Palermo in the presence of Emperor Frederick II: find a root of the equation $x^3 + 2x^2 + 10x = 20$. He first showed that the equation had no rational roots and no Euclidean irrational root—that is, no root in any of the forms $a \pm \sqrt{b}$, $\sqrt{a} \pm \sqrt{b}$, $\sqrt{a \pm \sqrt{b}}$, or $\sqrt{\sqrt{a} \pm \sqrt{b}}$, where a and b are rational numbers. He then approximated the only real root, probably using an algebraic technique of Omar Khayyam involving the intersection of a circle and a parabola. His answer was given in the base-60 number system as

$$1 + 22\left(\frac{1}{60}\right) + 7\left(\frac{1}{60}\right)^2 + 42\left(\frac{1}{60}\right)^3 + 33\left(\frac{1}{60}\right)^4 + 4\left(\frac{1}{60}\right)^5 + 40\left(\frac{1}{60}\right)^6.$$

How accurate was his approximation?

2.7 Survey of Methods and Software

In this chapter we have considered the problem of solving the equation $f(x) = 0$, where f is a given continuous function. All the methods begin with initial approximations and generate a sequence that converges to a root of the equation, if the method is successful. If $[a, b]$ is an interval on which $f(a)$ and $f(b)$ are of opposite sign, then the Bisection method and the method of False Position will converge. However, the convergence of these methods might be slow. Faster convergence is generally obtained using the Secant method or Newton's method. Good initial approximations are required for these methods, two for the Secant method and one for Newton's method, so the root-bracketing techniques such as Bisection or the False Position method can be used as starter methods for the Secant or Newton's method.

Müller's method will give rapid convergence without a particularly good initial approximation. It is not quite as efficient as Newton's method; its order of convergence near a root is approximately $\alpha = 1.84$, compared to the quadratic, $\alpha = 2$, order of Newton's method. However, it is better than the Secant method, whose order is approximately $\alpha = 1.62$, and it has the added advantage of being able to approximate complex roots.

Deflation is generally used with Müller's method once an approximate root of a polynomial has been determined. After an approximation to the root of the deflated equation has been determined, use either Müller's method or Newton's method in the original polynomial with this root as the initial approximation. This procedure will ensure that the root being approximated is a solution to the true equation, not to the deflated equation. We recommended Müller's method for finding all the zeros of polynomials, real or complex. Müller's method can also be used for an arbitrary continuous function.

Other high-order methods are available for determining the roots of polynomials. If this topic is of particular interest, we recommend that consideration be given to Laguerre's method, which gives cubic convergence and also approximates complex roots (see [Ho], pp. 176–179 for a complete discussion), the Jenkins-Traub method (see [JT]), and Brent's method (see [Bre]).

Another method of interest, Cauchy's method, is similar to Müller's method but avoids the failure problem of Müller's method when $f(x_i) = f(x_{i+1}) = f(x_{i+2})$, for some i. For an interesting discussion of this method, as well as more detail on Müller's method, we recommend [YG], Sections 4.10, 4.11, and 5.4.

Given a specified function f and a tolerance, an efficient program should produce an approximation to one or more solutions of $f(x) = 0$, each having an absolute or relative error within the tolerance, and the results should be generated in a reasonable amount of time. If the program cannot accomplish this task, it should at least give meaningful explanations of why success was not obtained and an indication of how to remedy the cause of failure.

IMSL has subroutines that implement Müller's method with deflation. Also included in this package is a routine due to R. P. Brent that uses a combination of linear interpolation, an inverse quadratic interpolation similar to Müller's method, and the Bisection method. Laguerre's method is also used to find zeros of a real polynomial. Another routine for finding the zeros of real polynomials uses a method of Jenkins-Traub, which is also used to find zeros of a complex polynomial.

The NAG library has a subroutine that uses a combination of the Bisection method, linear interpolation, and extrapolation to approximate a real zero of a function on a given interval. NAG also supplies subroutines to approximate all zeros of a real polynomial or complex polynomial, respectively. Both subroutines use a modified Laguerre method.

The netlib library contains a subroutine that uses a combination of the Bisection and Secant method developed by T. J. Dekker to approximate a real zero of a function in the interval. It requires specifying an interval that contains a root and returns an interval with a width that is within a specified tolerance. Another subroutine uses a combination of the bisection method, interpolation, and extrapolation to find a real zero of the function on the interval.

MATLAB has a routine to compute all the roots, both real and complex, of a polynomial, and one that computes a zero near a specified initial approximation to within a specified tolerance.

Notice that in spite of the diversity of methods, the professionally written packages are based primarily on the methods and principles discussed in this chapter. You should be able to use these packages by reading the manuals accompanying the packages to better understand the parameters and the specifications of the results that are obtained.

There are three books that we consider to be classics on the solution of nonlinear equations: those by Traub [Tr], by Ostrowski [Os], and by Householder [Ho]. In addition, the book by Brent [Bre] served as the basis for many of the currently used root-finding methods.

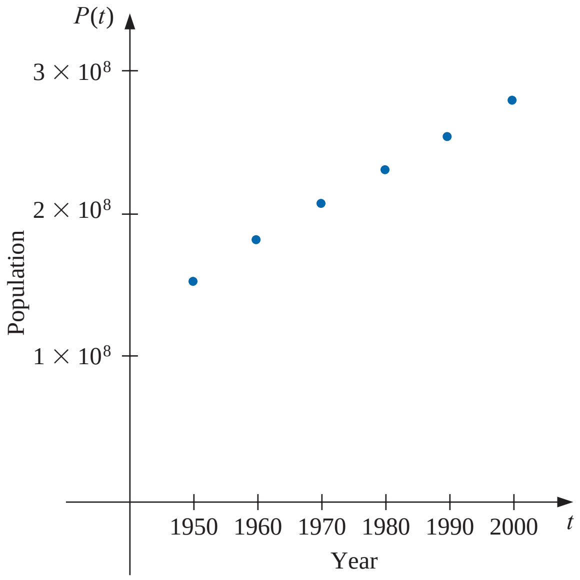
CHAPTER

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.

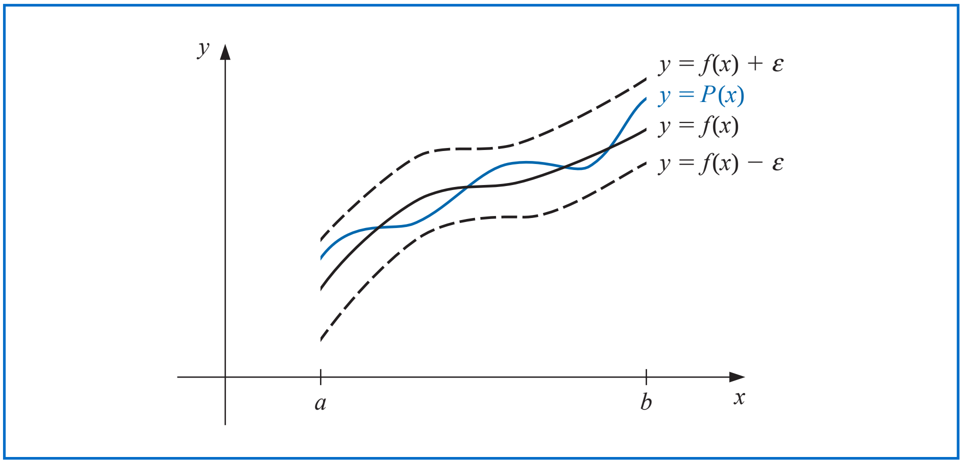
3.1 Interpolation and the Lagrange Polynomial

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

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

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

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.

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.

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

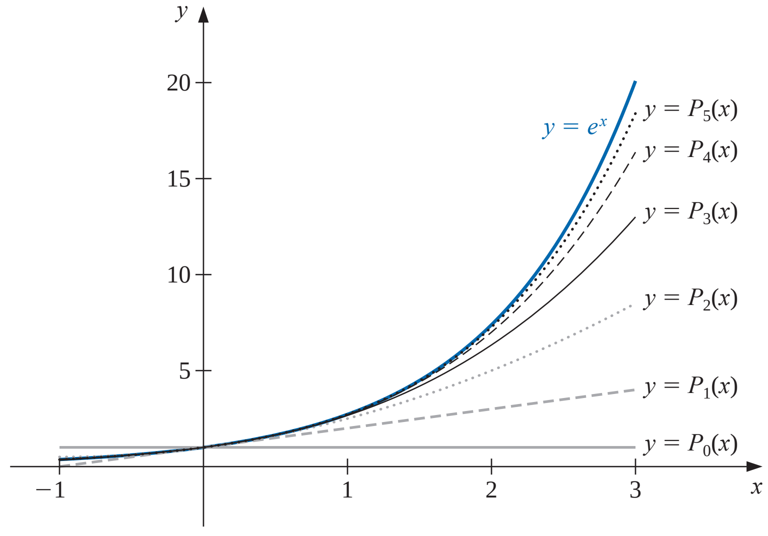
Very little of Weierstrass's work was published during his lifetime, but his lectures, particularly on the theory of functions, had significant influence on an entire generation of students.

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

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

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

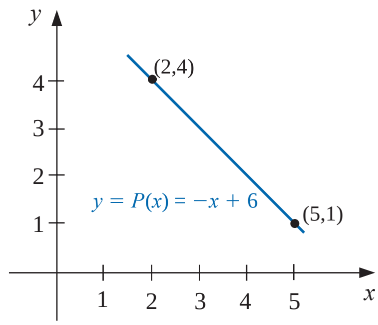
$$L_0(x) = \frac{x - 5}{2 - 5} = -\frac{1}{3}(x - 5) \quad \text{and} \quad L_1(x) = \frac{x - 2}{5 - 2} = \frac{1}{3}(x - 2),$$

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

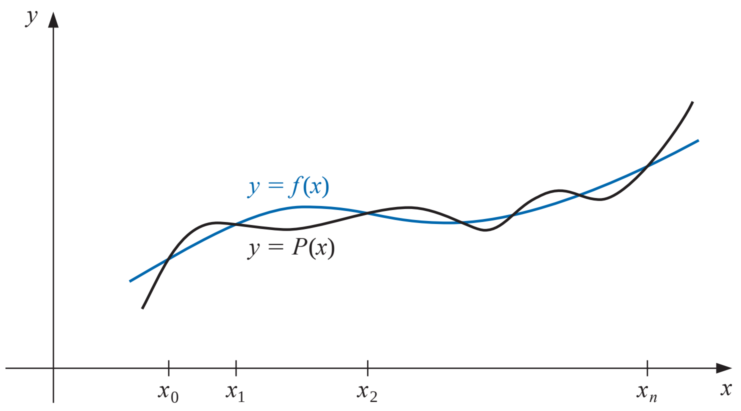


To generalize the concept of linear interpolation, consider the construction of a polynomial of degree at most n that passes through the $n + 1$ points

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

(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

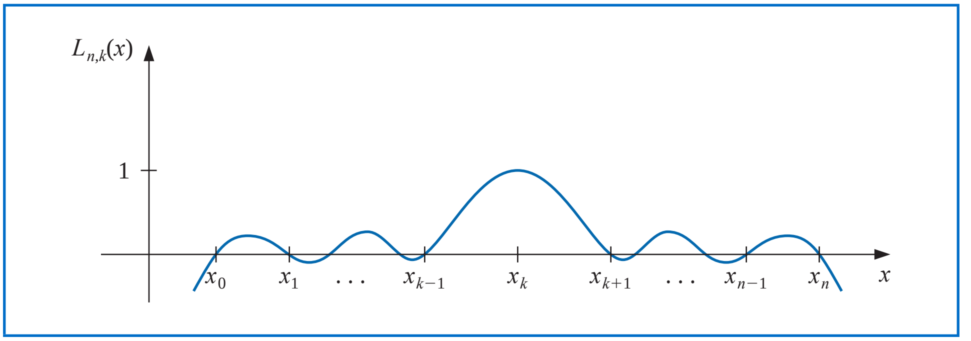
$$(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n).$$

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

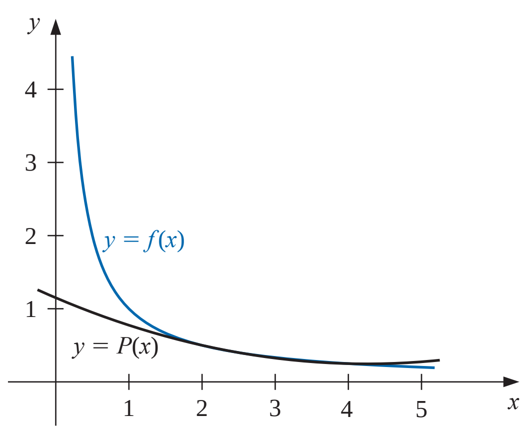
$$\begin{aligned}P(x) &= \sum_{k=0}^{2} f(x_k)L_k(x)\\ &= \frac{1}{3}(x-2.75)(x-4) - \frac{64}{165}(x-2)(x-4) + \frac{1}{10}(x-2)(x-2.75)\\ &= \frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44}.\end{aligned}$$

(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

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

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

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

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

■ ■ ■

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

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

Hence, the maximum error is

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

■

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

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

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

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

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.]