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 ***k*th 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:

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

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	$f[x_5]$			

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

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

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

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,

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1}{h}(f(x_1) - f(x_0)) = \frac{1}{h}\Delta f(x_0)$$

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

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

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

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

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

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:

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

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 (~~~). Notice that the fourth divided difference is used in both formulas.

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

□

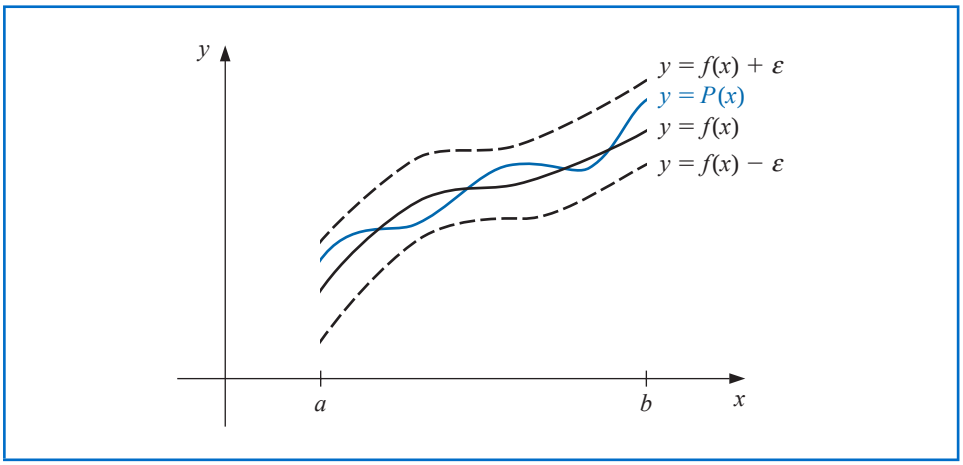
3.1 Interpolation and the Lagrange Polynomial

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.

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

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.

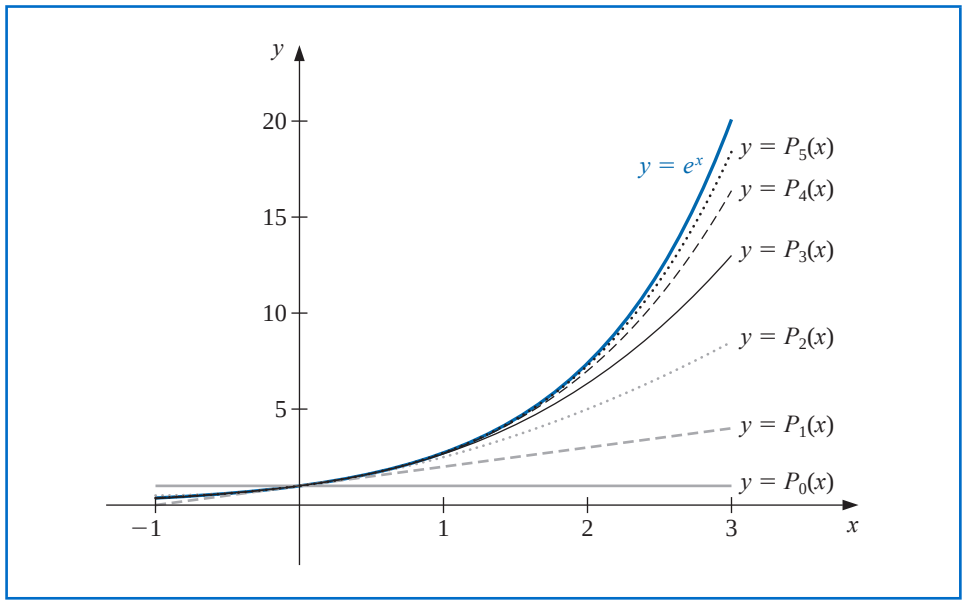
Very little of Weierstrass's work was published during his lifetime, but his lectures, particularly on the theory of functions, had significant influence on an entire generation of students.

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

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

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

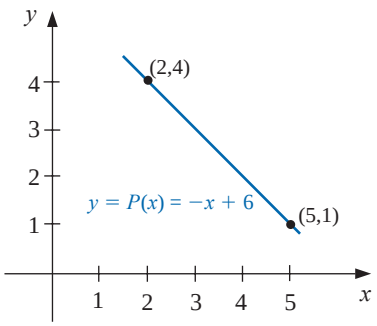
$$L_0(x) = \frac{x - 5}{2 - 5} = -\frac{1}{3}(x - 5) \quad \text{and} \quad L_1(x) = \frac{x - 2}{5 - 2} = \frac{1}{3}(x - 2),$$

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

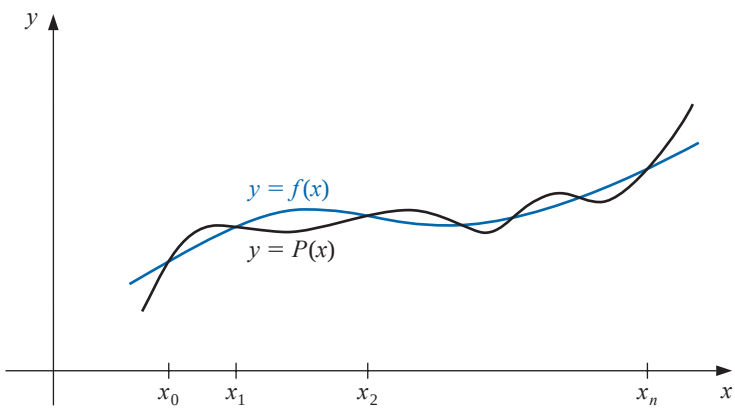


To generalize the concept of linear interpolation, consider the construction of a polynomial of degree at most n that passes through the $n + 1$ points

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

(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

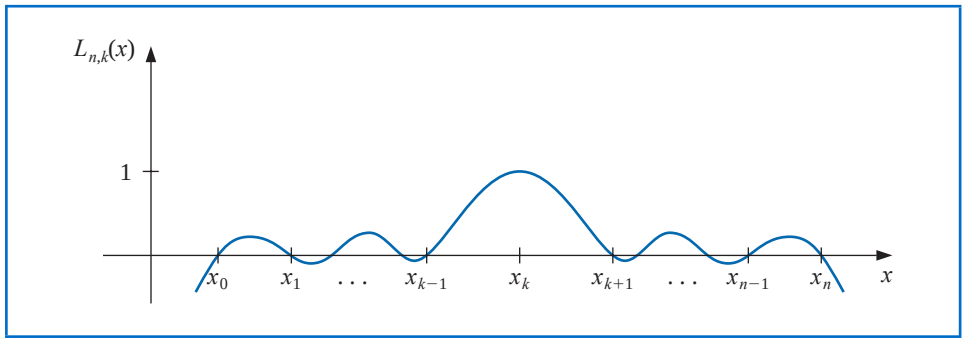
$$(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n).$$

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

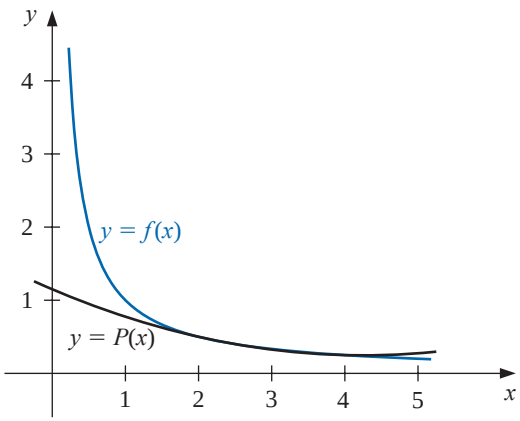
$$\begin{aligned} P(x) &= \sum_{k=0}^{2} f(x_k)L_k(x) \\ &= \frac{1}{3}(x - 2.75)(x - 4) - \frac{64}{165}(x - 2)(x - 4) + \frac{1}{10}(x - 2)(x - 2.75) \\ &= \frac{1}{22}x^2 - \frac{35}{88}x + \frac{49}{44}. \end{aligned}$$

(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

$$g(t) = f(t) - P(t) - [f(x) - P(x)]\frac{(t - x_0)(t - x_1)\cdots(t - x_n)}{(x - x_0)(x - x_1)\cdots(x - x_n)}$$

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

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

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

Hence, the maximum error is

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

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

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

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. $f(x) = \tan x$

c. $3x^2 - e^x = 0$, where g is the function in Exercise 12(c) of Section 2.2.

d. $x - \cos x = 0$, where g is the function in Exercise 12(d) of Section 2.2.

13. The following sequences converge to 0. Use Aitken's Δ^2 method to generate $\{\hat{p}_n\}$ until $|\hat{p}_n| \leq 5 \times 10^{-2}$:

a. $p_n = \dfrac{1}{n}, \quad n \geq 1$ **b.** $p_n = \dfrac{1}{n^2}, \quad n \geq 1$

14. A sequence $\{p_n\}$ is said to be **superlinearly convergent** to p if

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|} = 0.$$

a. Show that if $p_n \to p$ of order α for $\alpha > 1$, then $\{p_n\}$ is superlinearly convergent to p.

b. Show that $p_n = \frac{1}{n^n}$ is superlinearly convergent to 0 but does not converge to 0 of order α for any $\alpha > 1$.

15. Suppose that $\{p_n\}$ is superlinearly convergent to p. Show that

$$\lim_{n\to\infty} \frac{|p_{n+1} - p_n|}{|p_n - p|} = 1.$$

16. Prove Theorem 2.14. [*Hint:* Let $\delta_n = (p_{n+1} - p)/(p_n - p) - \lambda$, and show that $\lim_{n\to\infty} \delta_n = 0$. Then express $(\hat{p}_{n+1} - p)/(p_n - p)$ in terms of δ_n, δ_{n+1}, and λ.]

17. Let $P_n(x)$ be the nth Taylor polynomial for $f(x) = e^x$ expanded about $x_0 = 0$.

a. For fixed x, show that $p_n = P_n(x)$ satisfies the hypotheses of Theorem 2.14.

b. Let $x = 1$, and use Aitken's Δ^2 method to generate the sequence $\hat{p}_0, \ldots, \hat{p}_8$.

c. Does Aitken's method accelerate convergence in this situation?

2.6 Zeros of Polynomials and Müller's Method

A *polynomial of degree n* has the form

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

where the a_i's, called the *coefficients* of P, are constants and $a_n \neq 0$. The zero function, $P(x) = 0$ for all values of x, is considered a polynomial but is assigned no degree.

Algebraic Polynomials

Theorem 2.16 **(Fundamental Theorem of Algebra)**

If $P(x)$ is a polynomial of degree $n \geq 1$ with real or complex coefficients, then $P(x) = 0$ has at least one (possibly complex) root. ■

Although the Fundamental Theorem of Algebra is basic to any study of elementary functions, the usual proof requires techniques from the study of complex function theory. The reader is referred to [SaS], p. 155, for the culmination of a systematic development of the topics needed to prove the Theorem.

Example 1 Determine all the zeros of the polynomial $P(x) = x^3 - 5x^2 + 17x - 13$.

Solution It is easily verified that $P(1) = 1 - 5 + 17 - 13 = 0$. so $x = 1$ is a zero of P and $(x - 1)$ is a factor of the polynomial. Dividing $P(x)$ by $x - 1$ gives

$$P(x) = (x - 1)(x^2 - 4x + 13).$$

Carl Friedrich Gauss (1777–1855), one of the greatest mathematicians of all time, proved the Fundamental Theorem of Algebra in his doctoral dissertation and published it in 1799. He published different proofs of this result throughout his lifetime, in 1815, 1816, and as late as 1848. The result had been stated, without proof, by Albert Girard (1595–1632), and partial proofs had been given by Jean d'Alembert (1717–1783), Euler, and Lagrange.

To determine the zeros of $x^2 - 4x + 13$ we use the quadratic formula in its standard form, which gives the complex zeros

$$\frac{-(-4) \pm \sqrt{(-4)^2 - 4(1)(13)}}{2(1)} = \frac{4 \pm \sqrt{-36}}{2} = 2 \pm 3i.$$

Hence the third-degree polynomial $P(x)$ has three zeros, $x_1 = 1$, $x_2 = 2 - 3i$, and $x_2 = 2 + 3i$. ■

In the preceding example we found that the third-degree polynomial had three distinct zeros. An important consequence of the Fundamental Theorem of Algebra is the following corollary. It states that this is always the case, provided that when the zeros are not distinct we count the number of zeros according to their multiplicities.

Corollary 2.17 If $P(x)$ is a polynomial of degree $n \geq 1$ with real or complex coefficients, then there exist unique constants $x_1, x_2, \ldots, x_k$, possibly complex, and unique positive integers $m_1, m_2, \ldots, m_k$, such that $\sum_{i=1}^{k} m_i = n$ and

$$P(x) = a_n(x - x_1)^{m_1}(x - x_2)^{m_2} \cdots (x - x_k)^{m_k}.$$ ■

By Corollary 2.17 the collection of zeros of a polynomial is unique and, if each zero x_i is counted as many times as its multiplicity m_i, a polynomial of degree n has exactly n zeros.

The following corollary of the Fundamental Theorem of Algebra is used often in this section and in later chapters.

Corollary 2.18 Let $P(x)$ and $Q(x)$ be polynomials of degree at most n. If $x_1, x_2, \ldots, x_k$, with $k > n$, are distinct numbers with $P(x_i) = Q(x_i)$ for $i = 1, 2, \ldots, k$, then $P(x) = Q(x)$ for all values of x. ■

This result implies that to show that two polynomials of degree less than or equal to n are the same, we only need to show that they agree at $n + 1$ values. This will be frequently used, particularly in Chapters 3 and 8.

Horner's Method

William Horner (1786–1837) was a child prodigy who became headmaster of a school in Bristol at age 18. Horner's method for solving algebraic equations was published in 1819 in the Philosophical Transactions of the Royal Society.

To use Newton's method to locate approximate zeros of a polynomial $P(x)$, we need to evaluate $P(x)$ and $P'(x)$ at specified values. Since $P(x)$ and $P'(x)$ are both polynomials, computational efficiency requires that the evaluation of these functions be done in the nested manner discussed in Section 1.2. Horner's method incorporates this nesting technique, and, as a consequence, requires only n multiplications and n additions to evaluate an arbitrary nth-degree polynomial.

Theorem 2.19 **(Horner's Method)**

Let

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

Define $b_n = a_n$ and

$$b_k = a_k + b_{k+1}x_0, \quad \text{for } k = n - 1, n - 2, \ldots, 1, 0.$$

Then $b_0 = P(x_0)$. Moreover, if

$$Q(x) = b_n x^{n-1} + b_{n-1}x^{n-2} + \cdots + b_2 x + b_1,$$

then

$$P(x) = (x - x_0)Q(x) + b_0.$$ ■

Paolo Ruffini (1765–1822) had described a similar method which won him the gold medal from the Italian Mathematical Society for Science. Neither Ruffini nor Horner was the first to discover this method; it was known in China at least 500 years earlier.

Proof By the definition of $Q(x)$,

$$\begin{aligned}(x - x_0)Q(x) + b_0 &= (x - x_0)(b_n x^{n-1} + \cdots + b_2 x + b_1) + b_0 \\ &= (b_n x^n + b_{n-1}x^{n-1} + \cdots + b_2 x^2 + b_1 x) \\ &\quad - (b_n x_0 x^{n-1} + \cdots + b_2 x_0 x + b_1 x_0) + b_0 \\ &= b_n x^n + (b_{n-1} - b_n x_0)x^{n-1} + \cdots + (b_1 - b_2 x_0)x + (b_0 - b_1 x_0).\end{aligned}$$

By the hypothesis, $b_n = a_n$ and $b_k - b_{k+1}x_0 = a_k$, so

$$(x - x_0)Q(x) + b_0 = P(x) \quad \text{and} \quad b_0 = P(x_0).$$ ■ ■ ■

Example 2 Use Horner's method to evaluate $P(x) = 2x^4 - 3x^2 + 3x - 4$ at $x_0 = -2$.

Solution When we use hand calculation in Horner's method, we first construct a table, which suggests the *synthetic division* name that is often applied to the technique. For this problem, the table appears as follows:

	Coefficient of x^4	Coefficient of x^3	Coefficient of x^2	Coefficient of x	Constant term
$x_0 = -2$	$a_4 = 2$	$a_3 = 0$ $b_4x_0 = -4$	$a_2 = -3$ $b_3x_0 = 8$	$a_1 = 3$ $b_2x_0 = -10$	$a_0 = -4$ $b_1x_0 = 14$
	$b_4 = 2$	$b_3 = -4$	$b_2 = 5$	$b_1 = -7$	$b_0 = 10$

So,

$$P(x) = (x + 2)(2x^3 - 4x^2 + 5x - 7) + 10.$$ ■

The word synthetic has its roots in various languages. In standard English it generally provides the sense of something that is "false" or "substituted". But in mathematics it takes the form of something that is "grouped together". Synthetic geometry treats shapes as whole, rather than as individual objects, which is the style in analytic geometry. In synthetic division of polynomials, the various powers of the variables are not explicitly given but kept grouped together.

An additional advantage of using the Horner (or synthetic-division) procedure is that, since

$$P(x) = (x - x_0)Q(x) + b_0,$$

where

$$Q(x) = b_n x^{n-1} + b_{n-1}x^{n-2} + \cdots + b_2 x + b_1,$$

differentiating with respect to x gives

$$P'(x) = Q(x) + (x - x_0)Q'(x) \quad \text{and} \quad P'(x_0) = Q(x_0). \tag{2.16}$$

When the Newton-Raphson method is being used to find an approximate zero of a polynomial, $P(x)$ and $P'(x)$ can be evaluated in the same manner.

Example 3 Find an approximation to a zero of

$$P(x) = 2x^4 - 3x^2 + 3x - 4,$$

using Newton's method with $x_0 = -2$ and synthetic division to evaluate $P(x_n)$ and $P'(x_n)$ for each iterate x_n.

Solution With $x_0 = -2$ as an initial approximation, we obtained $P(-2)$ in Example 1 by

$x_0 = -2$	2	0	−3	3	−4	
		−4	8	−10	14	
	2	−4	5	−7	10	$= P(-2).$

Using Theorem 2.19 and Eq. (2.16),

$$Q(x) = 2x^3 - 4x^2 + 5x - 7 \quad \text{and} \quad P'(-2) = Q(-2),$$

so $P'(-2)$ can be found by evaluating $Q(-2)$ in a similar manner:

$x_0 = -2$	2	−4	5	−7	
		−4	16	−42	
	2	−8	21	−49	$= Q(-2) = P'(-2)$

and

$$x_1 = x_0 - \frac{P(x_0)}{P'(x_0)} = x_0 - \frac{P(x_0)}{Q(x_0)} = -2 - \frac{10}{-49} \approx -1.796.$$

Repeating the procedure to find x_2 gives

−1.796	2	0	−3	3	−4	
		−3.592	6.451	−6.197	5.742	
	2	−3.592	3.451	−3.197	1.742	$= P(x_1)$
		−3.592	12.902	−29.368		
	2	−7.184	16.353	−32.565	$= Q(x_1)$	$= P'(x_1).$

So $P(-1.796) = 1.742$, $P'(-1.796) = Q(-1.796) = -32.565$, and

$$x_2 = -1.796 - \frac{1.742}{-32.565} \approx -1.7425.$$

In a similar manner, $x_3 = -1.73897$, and an actual zero to five decimal places is -1.73896.

Note that the polynomial $Q(x)$ depends on the approximation being used and changes from iterate to iterate. ■

Algorithm 2.7 computes $P(x_0)$ and $P'(x_0)$ using Horner's method.

ALGORITHM
2.7

Horner's

To evaluate the polynomial

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = (x - x_0)Q(x) + b_0$$

and its derivative at x_0:

INPUT degree n; coefficients $a_0, a_1, \ldots, a_n; x_0$.

OUTPUT $y = P(x_0); z = P'(x_0)$.

Step 1 Set $y = a_n$; (*Compute b_n for P.*)
$z = a_n$. (*Compute b_{n-1} for Q.*)

Step 2 For $j = n - 1, n - 2, \ldots, 1$
set $y = x_0 y + a_j$; (*Compute b_j for P.*)
$z = x_0 z + y$. (*Compute b_{j-1} for Q.*)

Step 3 Set $y = x_0 y + a_0$. (*Compute b_0 for P.*)

Step 4 OUTPUT (y, z);
STOP. ■

If the Nth iterate, x_N, in Newton's method is an approximate zero for P, then

$$P(x) = (x - x_N)Q(x) + b_0 = (x - x_N)Q(x) + P(x_N) \approx (x - x_N)Q(x),$$

so $x - x_N$ is an approximate factor of $P(x)$. Letting $\hat{x}_1 = x_N$ be the approximate zero of P and $Q_1(x) \equiv Q(x)$ be the approximate factor gives

$$P(x) \approx (x - \hat{x}_1)Q_1(x).$$

We can find a second approximate zero of P by applying Newton's method to $Q_1(x)$.

If $P(x)$ is an nth-degree polynomial with n real zeros, this procedure applied repeatedly will eventually result in $(n-2)$ approximate zeros of P and an approximate quadratic factor $Q_{n-2}(x)$. At this stage, $Q_{n-2}(x) = 0$ can be solved by the quadratic formula to find the last two approximate zeros of P. Although this method can be used to find all the approximate zeros, it depends on repeated use of approximations and can lead to inaccurate results.

The procedure just described is called **deflation**. The accuracy difficulty with deflation is due to the fact that, when we obtain the approximate zeros of $P(x)$, Newton's method is used on the reduced polynomial $Q_k(x)$, that is, the polynomial having the property that

$$P(x) \approx (x - \hat{x}_1)(x - \hat{x}_2) \cdots (x - \hat{x}_k)Q_k(x).$$

An approximate zero $\hat{x}_{k+1}$ of Q_k will generally not approximate a root of $P(x) = 0$ as well as it does a root of the reduced equation $Q_k(x) = 0$, and inaccuracy increases as k increases. One way to eliminate this difficulty is to use the reduced equations to find approximations $\hat{x}_2$, $\hat{x}_3, \ldots, \hat{x}_k$ to the zeros of P, and then improve these approximations by applying Newton's method to the original polynomial $P(x)$.

Complex Zeros: Müller's Method

One problem with applying the Secant, False Position, or Newton's method to polynomials is the possibility of the polynomial having complex roots even when all the coefficients are

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.

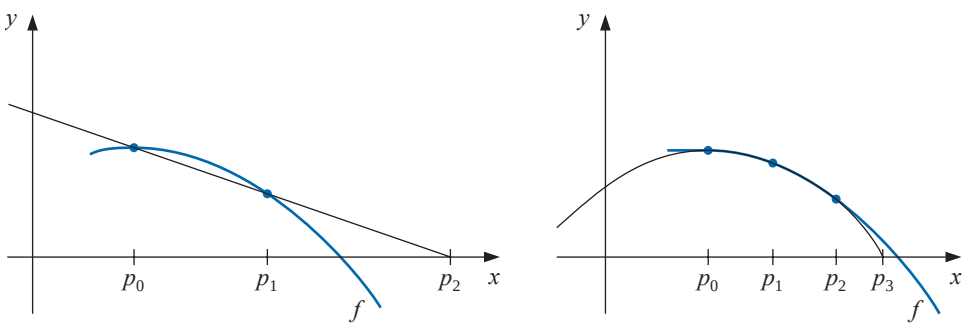
Theorem 2.20 If $z = a+bi$ is a complex zero of multiplicity m of the polynomial $P(x)$ with real coefficients, then $\bar{z} = a - bi$ is also a zero of multiplicity m of the polynomial $P(x)$, and $(x^2 - 2ax + a^2 + b^2)^m$ is a factor of $P(x)$. ■

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

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

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.

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

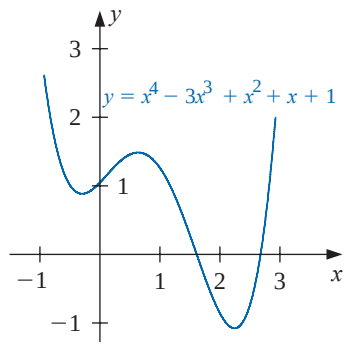
Step 6 If $|h| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

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 **c.** Secant method **e.** Müller's method

b. Newton's method **d.** method of False Position

2.4 Error Analysis for Iterative Methods

In this section we investigate the order of convergence of functional iteration schemes and, as a means of obtaining rapid convergence, rediscover Newton's method. We also consider ways of accelerating the convergence of Newton's method in special circumstances. First, however, we need a new procedure for measuring how rapidly a sequence converges.

Order of Convergence

Definition 2.7 Suppose $\{p_n\}_{n=0}^{\infty}$ is a sequence that converges to p, with $p_n \neq p$ for all n. If positive constants λ and α exist with

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}} = \lambda,$$

then $\{p_n\}_{n=0}^{\infty}$ **converges to p of order α, with asymptotic error constant λ.** ■

An iterative technique of the form $p_n = g(p_{n-1})$ is said to be of *order* α if the sequence $\{p_n\}_{n=0}^{\infty}$ converges to the solution $p = g(p)$ of order α.

In general, a sequence with a high order of convergence converges more rapidly than a sequence with a lower order. The asymptotic constant affects the speed of convergence but not to the extent of the order. Two cases of order are given special attention.

(i) If $\alpha = 1$ (and $\lambda < 1$), the sequence is **linearly convergent**.

(ii) If $\alpha = 2$, the sequence is **quadratically convergent**.

The next illustration compares a linearly convergent sequence to one that is quadratically convergent. It shows why we try to find methods that produce higher-order convergent sequences.

Illustration Suppose that $\{p_n\}_{n=0}^{\infty}$ is linearly convergent to 0 with

$$\lim_{n\to\infty} \frac{|p_{n+1}|}{|p_n|} = 0.5$$

and that $\{\tilde{p}_n\}_{n=0}^{\infty}$ is quadratically convergent to 0 with the same asymptotic error constant,

$$\lim_{n\to\infty} \frac{|\tilde{p}_{n+1}|}{|\tilde{p}_n|^2} = 0.5.$$

For simplicity we assume that for each n we have

$$\frac{|p_{n+1}|}{|p_n|} \approx 0.5 \quad \text{and} \quad \frac{|\tilde{p}_{n+1}|}{|\tilde{p}_n|^2} \approx 0.5.$$

For the linearly convergent scheme, this means that

$$|p_n - 0| = |p_n| \approx 0.5|p_{n-1}| \approx (0.5)^2|p_{n-2}| \approx \cdots \approx (0.5)^n|p_0|,$$

whereas the quadratically convergent procedure has

$$\begin{aligned}|\tilde{p}_n - 0| = |\tilde{p}_n| &\approx 0.5|\tilde{p}_{n-1}|^2 \approx (0.5)[0.5|\tilde{p}_{n-2}|^2]^2 = (0.5)^3|\tilde{p}_{n-2}|^4 \\ &\approx (0.5)^3[(0.5)|\tilde{p}_{n-3}|^2]^4 = (0.5)^7|\tilde{p}_{n-3}|^8 \\ &\approx \cdots \approx (0.5)^{2^n-1}|\tilde{p}_0|^{2^n}.\end{aligned}$$

Table 2.7 illustrates the relative speed of convergence of the sequences to 0 if $|p_0| = |\tilde{p}_0| = 1$.

Table 2.7

n	Linear Convergence Sequence $\{p_n\}_{n=0}^{\infty}$ $(0.5)^n$	Quadratic Convergence Sequence $\{\tilde{p}_n\}_{n=0}^{\infty}$ $(0.5)^{2^n-1}$
1	5.0000×10^{-1}	5.0000×10^{-1}
2	2.5000×10^{-1}	1.2500×10^{-1}
3	1.2500×10^{-1}	7.8125×10^{-3}
4	6.2500×10^{-2}	3.0518×10^{-5}
5	3.1250×10^{-2}	4.6566×10^{-10}
6	1.5625×10^{-2}	1.0842×10^{-19}
7	7.8125×10^{-3}	5.8775×10^{-39}

The quadratically convergent sequence is within 10^{-38} of 0 by the seventh term. At least 126 terms are needed to ensure this accuracy for the linearly convergent sequence. □

Quadratically convergent sequences are expected to converge much quicker than those that converge only linearly, but the next result implies that an arbitrary technique that generates a convergent sequences does so only linearly.

Theorem 2.8 Let $g \in C[a, b]$ be such that $g(x) \in [a, b]$, for all $x \in [a, b]$. Suppose, in addition, that g' is continuous on (a, b) and a positive constant $k < 1$ exists with

$$|g'(x)| \leq k, \quad \text{for all } x \in (a, b).$$

If $g'(p) \neq 0$, then for any number $p_0 \neq p$ in $[a, b]$, the sequence

$$p_n = g(p_{n-1}), \quad \text{for } n \geq 1,$$

converges only linearly to the unique fixed point p in $[a, b]$. ■

Proof We know from the Fixed-Point Theorem 2.4 in Section 2.2 that the sequence converges to p. Since g' exists on (a, b), we can apply the Mean Value Theorem to g to show that for any n,

$$p_{n+1} - p = g(p_n) - g(p) = g'(\xi_n)(p_n - p),$$

where ξ_n is between p_n and p. Since $\{p_n\}_{n=0}^{\infty}$ converges to p, we also have $\{\xi_n\}_{n=0}^{\infty}$ converging to p. Since g' is continuous on (a, b), we have

$$\lim_{n\to\infty} g'(\xi_n) = g'(p).$$

Thus

$$\lim_{n\to\infty} \frac{p_{n+1} - p}{p_n - p} = \lim_{n\to\infty} g'(\xi_n) = g'(p) \quad \text{and} \quad \lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|} = |g'(p)|.$$

Hence, if $g'(p) \neq 0$, fixed-point iteration exhibits linear convergence with asymptotic error constant $|g'(p)|$. ■ ■ ■

Theorem 2.8 implies that higher-order convergence for fixed-point methods of the form $g(p) = p$ can occur only when $g'(p) = 0$. The next result describes additional conditions that ensure the quadratic convergence we seek.

Theorem 2.9 Let p be a solution of the equation $x = g(x)$. Suppose that $g'(p) = 0$ and g'' is continuous with $|g''(x)| < M$ on an open interval I containing p. Then there exists a $\delta > 0$ such that, for $p_0 \in [p - \delta, p + \delta]$, the sequence defined by $p_n = g(p_{n-1})$, when $n \geq 1$, converges at least quadratically to p. Moreover, for sufficiently large values of n,

$$|p_{n+1} - p| < \frac{M}{2}|p_n - p|^2.$$

■

Proof Choose k in $(0, 1)$ and $\delta > 0$ such that on the interval $[p-\delta, p+\delta]$, contained in I, we have $|g'(x)| \leq k$ and g'' continuous. Since $|g'(x)| \leq k < 1$, the argument used in the proof of Theorem 2.6 in Section 2.3 shows that the terms of the sequence $\{p_n\}_{n=0}^{\infty}$ are contained in $[p - \delta, p + \delta]$. Expanding $g(x)$ in a linear Taylor polynomial for $x \in [p - \delta, p + \delta]$ gives

$$g(x) = g(p) + g'(p)(x - p) + \frac{g''(\xi)}{2}(x - p)^2,$$

where ξ lies between x and p. The hypotheses $g(p) = p$ and $g'(p) = 0$ imply that

$$g(x) = p + \frac{g''(\xi)}{2}(x - p)^2.$$

In particular, when $x = p_n$,

$$p_{n+1} = g(p_n) = p + \frac{g''(\xi_n)}{2}(p_n - p)^2,$$

with ξ_n between p_n and p. Thus,

$$p_{n+1} - p = \frac{g''(\xi_n)}{2}(p_n - p)^2.$$

Since $|g'(x)| \leq k < 1$ on $[p - \delta, p + \delta]$ and g maps $[p - \delta, p + \delta]$ into itself, it follows from the Fixed-Point Theorem that $\{p_n\}_{n=0}^{\infty}$ converges to p. But ξ_n is between p and p_n for each n, so $\{\xi_n\}_{n=0}^{\infty}$ also converges to p, and

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^2} = \frac{|g''(p)|}{2}.$$

This result implies that the sequence $\{p_n\}_{n=0}^{\infty}$ is quadratically convergent if $g''(p) \neq 0$ and of higher-order convergence if $g''(p) = 0$.

Because g'' is continuous and strictly bounded by M on the interval $[p - \delta, p + \delta]$, this also implies that, for sufficiently large values of n,

$$|p_{n+1} - p| < \frac{M}{2}|p_n - p|^2.$$

■ ■ ■

Theorems 2.8 and 2.9 tell us that our search for quadratically convergent fixed-point methods should point in the direction of functions whose derivatives are zero at the fixed point. That is:

- For a fixed point method to converge quadratically we need to have both $g(p) = p$, and $g'(p) = 0$.

The easiest way to construct a fixed-point problem associated with a root-finding problem $f(x) = 0$ is to add or subtract a multiple of $f(x)$ from x. Consider the sequence

$$p_n = g(p_{n-1}), \quad \text{for } n \geq 1,$$

for g in the form

$$g(x) = x - \phi(x) f(x),$$

where ϕ is a differentiable function that will be chosen later.

For the iterative procedure derived from g to be quadratically convergent, we need to have $g'(p) = 0$ when $f(p) = 0$. Because

$$g'(x) = 1 - \phi'(x) f(x) - f'(x)\phi(x),$$

and $f(p) = 0$, we have

$$g'(p) = 1 - \phi'(p) f(p) - f'(p)\phi(p) = 1 - \phi'(p) \cdot 0 - f'(p)\phi(p) = 1 - f'(p)\phi(p),$$

and $g'(p) = 0$ if and only if $\phi(p) = 1/f'(p)$.

If we let $\phi(x) = 1/f'(x)$, then we will ensure that $\phi(p) = 1/f'(p)$ and produce the quadratically convergent procedure

$$p_n = g(p_{n-1}) = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}.$$

This, of course, is simply Newton's method. Hence

- If $f(p) = 0$ and $f'(p) \neq 0$, then for starting values sufficiently close to p, Newton's method will converge at least quadratically.

Multiple Roots

In the preceding discussion, the restriction was made that $f'(p) \neq 0$, where p is the solution to $f(x) = 0$. In particular, Newton's method and the Secant method will generally give problems if $f'(p) = 0$ when $f(p) = 0$. To examine these difficulties in more detail, we make the following definition.

Definition 2.10 A solution p of $f(x) = 0$ is a **zero of multiplicity** m of f if for $x \neq p$, we can write $f(x) = (x - p)^m q(x)$, where $\lim_{x \to p} q(x) \neq 0$. ■

For polynomials, p is a zero of multiplicity m of f if $f(x) = (x - p)^m q(x)$, where $q(p) \neq 0$.

In essence, $q(x)$ represents that portion of $f(x)$ that does not contribute to the zero of f. The following result gives a means to easily identify **simple** zeros of a function, those that have multiplicity one.

Theorem 2.11 The function $f \in C^1[a, b]$ has a simple zero at p in (a, b) if and only if $f(p) = 0$, but $f'(p) \neq 0$. ■

Proof If f has a simple zero at p, then $f(p) = 0$ and $f(x) = (x - p)q(x)$, where $\lim_{x \to p} q(x) \neq 0$. Since $f \in C^1[a, b]$,

$$f'(p) = \lim_{x \to p} f'(x) = \lim_{x \to p} [q(x) + (x - p)q'(x)] = \lim_{x \to p} q(x) \neq 0.$$

Conversely, if $f(p) = 0$, but $f'(p) \neq 0$, expand f in a zeroth Taylor polynomial about p. Then

$$f(x) = f(p) + f'(\xi(x))(x - p) = (x - p) f'(\xi(x)),$$

where $\xi(x)$ is between x and p. Since $f \in C^1[a, b]$,

$$\lim_{x \to p} f'(\xi(x)) = f'\left(\lim_{x \to p} \xi(x)\right) = f'(p) \neq 0.$$

Letting $q = f' \circ \xi$ gives $f(x) = (x - p)q(x)$, where $\lim_{x \to p} q(x) \neq 0$. Thus f has a simple zero at p. ■ ■ ■

The following generalization of Theorem 2.11 is considered in Exercise 12.

Theorem 2.12 The function $f \in C^m[a, b]$ has a zero of multiplicity m at p in (a, b) if and only if

$$0 = f(p) = f'(p) = f''(p) = \cdots = f^{(m-1)}(p), \quad \text{but} \quad f^{(m)}(p) \neq 0. \qquad ■$$

The result in Theorem 2.12 implies that an interval about p exists where Newton's method converges quadratically to p for any initial approximation $p_0 = p$, provided that p is a simple zero. The following example shows that quadratic convergence might not occur if the zero is not simple.

Example 1 Let $f(x) = e^x - x - 1$. **(a)** Show that f has a zero of multiplicity 2 at $x = 0$. **(b)** Show that Newton's method with $p_0 = 1$ converges to this zero but not quadratically.

Solution **(a)** We have

$$f(x) = e^x - x - 1, \qquad f'(x) = e^x - 1 \qquad \text{and} \qquad f''(x) = e^x,$$

so

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

Theorem 2.12 implies that f has a zero of multiplicity 2 at $x = 0$.

(b) The first two terms generated by Newton's method applied to f with $p_0 = 1$ are

$$p_1 = p_0 - \frac{f(p_0)}{f'(p_0)} = 1 - \frac{e - 2}{e - 1} \approx 0.58198,$$

and

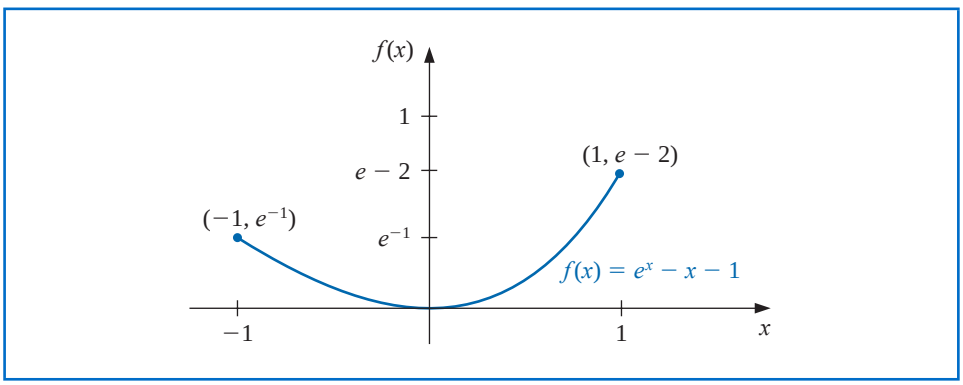
$$p_2 = p_1 - \frac{f(p_1)}{f'(p_1)} \approx 0.58198 - \frac{0.20760}{0.78957} \approx 0.31906.$$

The first sixteen terms of the sequence generated by Newton's method are shown in Table 2.8. The sequence is clearly converging to 0, but not quadratically. The graph of f is shown in Figure 2.12. ■

Table 2.8

n	p_n
0	1.0
1	0.58198
2	0.31906
3	0.16800
4	0.08635
5	0.04380
6	0.02206
7	0.01107
8	0.005545
9	2.7750×10^{-3}
10	1.3881×10^{-3}
11	6.9411×10^{-4}
12	3.4703×10^{-4}
13	1.7416×10^{-4}
14	8.8041×10^{-5}
15	4.2610×10^{-5}
16	1.9142×10^{-6}

Figure 2.12



One method of handling the problem of multiple roots of a function f is to define

$$\mu(x) = \frac{f(x)}{f'(x)}.$$

If p is a zero of f of multiplicity m with $f(x) = (x-p)^m q(x)$, then

$$\mu(x) = \frac{(x-p)^m q(x)}{m(x-p)^{m-1}q(x) + (x-p)^m q'(x)}$$
$$= (x-p)\frac{q(x)}{mq(x) + (x-p)q'(x)}$$

also has a zero at p. However, $q(p) \neq 0$, so

$$\frac{q(p)}{mq(p) + (p-p)q'(p)} = \frac{1}{m} \neq 0,$$

and p is a simple zero of $\mu(x)$. Newton's method can then be applied to $\mu(x)$ to give

$$g(x) = x - \frac{\mu(x)}{\mu'(x)} = x - \frac{f(x)/f'(x)}{\{[f'(x)]^2 - [f(x)][f''(x)]\}/[f'(x)]^2}$$

which simplifies to

$$g(x) = x - \frac{f(x)f'(x)}{[f'(x)]^2 - f(x)f''(x)}. \tag{2.13}$$

If g has the required continuity conditions, functional iteration applied to g will be quadratically convergent regardless of the multiplicity of the zero of f. Theoretically, the only drawback to this method is the additional calculation of $f''(x)$ and the more laborious procedure of calculating the iterates. In practice, however, multiple roots can cause serious round-off problems because the denominator of (2.13) consists of the difference of two numbers that are both close to 0.

Example 2 In Example 1 it was shown that $f(x) = e^x - x - 1$ has a zero of multiplicity 2 at $x = 0$ and that Newton's method with $p_0 = 1$ converges to this zero but not quadratically. Show that the modification of Newton's method as given in Eq. (2.13) improves the rate of convergence.

Solution Modified Newton's method gives

$$p_1 = p_0 - \frac{f(p_0)f'(p_0)}{f'(p_0)^2 - f(p_0)f''(p_0)} = 1 - \frac{(e-2)(e-1)}{(e-1)^2 - (e-2)e} \approx -2.3421061 \times 10^{-1}.$$

This is considerably closer to 0 than the first term using Newton's method, which was 0.58918. Table 2.9 lists the first five approximations to the double zero at $x = 0$. The results were obtained using a system with ten digits of precision. The relative lack of improvement in the last two entries is due to the fact that using this system both the numerator and the denominator approach 0. Consequently there is a loss of significant digits of accuracy as the approximations approach 0. ■

Table 2.9

n	p_n
1	$-2.3421061 \times 10^{-1}$
2	$-8.4582788 \times 10^{-3}$
3	$-1.1889524 \times 10^{-5}$
4	$-6.8638230 \times 10^{-6}$
5	$-2.8085217 \times 10^{-7}$

The following illustrates that the modified Newton's method converges quadratically even when in the case of a simple zero.

Illustration In Section 2.2 we found that a zero of $f(x) = x^3 + 4x^2 - 10 = 0$ is $p = 1.36523001$. Here we will compare convergence for a simple zero using both Newton's method and the modified Newton's method listed in Eq. (2.13). Let

(i) $p_n = p_{n-1} - \dfrac{p_{n-1}^3 + 4p_{n-1}^2 - 10}{3p_{n-1}^2 + 8p_{n-1}}$, from Newton's method

and, from the Modified Newton's method given by Eq. (2.13),

(ii) $p_n = p_{n-1} - \dfrac{(p_{n-1}^3 + 4p_{n-1}^2 - 10)(3p_{n-1}^2 + 8p_{n-1})}{(3p_{n-1}^2 + 8p_{n-1})^2 - (p_{n-1}^3 + 4p_{n-1}^2 - 10)(6p_{n-1} + 8)}$.

With $p_0 = 1.5$, we have

Newton's method

$$p_1 = 1.37333333, \quad p_2 = 1.36526201, \quad \text{and} \quad p_3 = 1.36523001.$$

Modified Newton's method

$$p_1 = 1.35689898, \quad p_2 = 1.36519585, \quad \text{and} \quad p_3 = 1.36523001.$$

Both methods are rapidly convergent to the actual zero, which is given by both methods as p_3. Note, however, that in the case of a simple zero the original Newton's method requires substantially less computation. □

Maple contains Modified Newton's method as described in Eq. (2.13) in its *Numerical-Analysis* package. The options for this command are the same as those for the Bisection method. To obtain results similar to those in Table 2.9 we can use

with(*Student*[*NumericalAnalysis*])

$f := e^x - x - 1$

ModifiedNewton $\left(f, x = 1.0, tolerance = 10^{-10}, output = sequence, maxiterations = 20\right)$

Remember that there is sensitivity to round-off error in these calculations, so you might need to reset *Digits* in Maple to get the exact values in Table 2.9.

EXERCISE SET 2.4

1. Use Newton's method to find solutions accurate to within 10^{-5} to the following problems.

a. $x^2 - 2xe^{-x} + e^{-2x} = 0, \quad \text{for } 0 \le x \le 1$

b. $\cos(x + \sqrt{2}) + x(x/2 + \sqrt{2}) = 0, \quad \text{for } -2 \le x \le -1$

c. $x^3 - 3x^2(2^{-x}) + 3x(4^{-x}) - 8^{-x} = 0, \quad \text{for } 0 \le x \le 1$

d. $e^{6x} + 3(\ln 2)^2 e^{2x} - (\ln 8)e^{4x} - (\ln 2)^3 = 0, \quad \text{for } -1 \le x \le 0$

2. Use Newton's method to find solutions accurate to within 10^{-5} to the following problems.

a. $1 - 4x\cos x + 2x^2 + \cos 2x = 0, \quad \text{for } 0 \le x \le 1$

b. $x^2 + 6x^5 + 9x^4 - 2x^3 - 6x^2 + 1 = 0, \quad \text{for } -3 \le x \le -2$

c. $\sin 3x + 3e^{-2x}\sin x - 3e^{-x}\sin 2x - e^{-3x} = 0, \quad \text{for } 3 \le x \le 4$

d. $e^{3x} - 27x^6 + 27x^4 e^x - 9x^2 e^{2x} = 0, \quad \text{for } 3 \le x \le 5$

3. Repeat Exercise 1 using the modified Newton's method described in Eq. (2.13). Is there an improvement in speed or accuracy over Exercise 1?

14. Use a fixed-point iteration method to determine a solution accurate to within 10^{-4} for $x = \tan x$, for x in $[4, 5]$.

15. Use a fixed-point iteration method to determine a solution accurate to within 10^{-2} for $2 \sin \pi x + x = 0$ on $[1, 2]$. Use $p_0 = 1$.

16. Let A be a given positive constant and $g(x) = 2x - Ax^2$.

a. Show that if fixed-point iteration converges to a nonzero limit, then the limit is $p = 1/A$, so the inverse of a number can be found using only multiplications and subtractions.

b. Find an interval about $1/A$ for which fixed-point iteration converges, provided p_0 is in that interval.

17. Find a function g defined on $[0, 1]$ that satisfies none of the hypotheses of Theorem 2.3 but still has a unique fixed point on $[0, 1]$.

18. **a.** Show that Theorem 2.2 is true if the inequality $|g'(x)| \leq k$ is replaced by $g'(x) \leq k$, for all $x \in (a, b)$. [*Hint:* Only uniqueness is in question.]

b. Show that Theorem 2.3 may not hold if inequality $|g'(x)| \leq k$ is replaced by $g'(x) \leq k$. [*Hint:* Show that $g(x) = 1 - x^2$, for x in $[0, 1]$, provides a counterexample.]

19. **a.** Use Theorem 2.4 to show that the sequence defined by

$$x_n = \frac{1}{2}x_{n-1} + \frac{1}{x_{n-1}}, \quad \text{for } n \geq 1,$$

converges to $\sqrt{2}$ whenever $x_0 > \sqrt{2}$.

b. Use the fact that $0 < (x_0 - \sqrt{2})^2$ whenever $x_0 \neq \sqrt{2}$ to show that if $0 < x_0 < \sqrt{2}$, then $x_1 > \sqrt{2}$.

c. Use the results of parts (a) and (b) to show that the sequence in (a) converges to $\sqrt{2}$ whenever $x_0 > 0$.

20. **a.** Show that if A is any positive number, then the sequence defined by

$$x_n = \frac{1}{2}x_{n-1} + \frac{A}{2x_{n-1}}, \quad \text{for } n \geq 1,$$

converges to $\sqrt{A}$ whenever $x_0 > 0$.

b. What happens if $x_0 < 0$?

21. Replace the assumption in Theorem 2.4 that "a positive number $k < 1$ exists with $|g'(x)| \leq k$" with "g satisfies a Lipschitz condition on the interval $[a, b]$ with Lipschitz constant $L < 1$." (See Exercise 27, Section 1.1.) Show that the conclusions of this theorem are still valid.

22. Suppose that g is continuously differentiable on some interval (c, d) that contains the fixed point p of g. Show that if $|g'(p)| < 1$, then there exists a $\delta > 0$ such that if $|p_0 - p| \leq \delta$, then the fixed-point iteration converges.

23. An object falling vertically through the air is subjected to viscous resistance as well as to the force of gravity. Assume that an object with mass m is dropped from a height s_0 and that the height of the object after t seconds is

$$s(t) = s_0 - \frac{mg}{k}t + \frac{m^2 g}{k^2}(1 - e^{-kt/m}),$$

where $g = 32.17$ ft/s^2 and k represents the coefficient of air resistance in lb-s/ft. Suppose $s_0 = 300$ ft, $m = 0.25$ lb, and $k = 0.1$ lb-s/ft. Find, to within 0.01 s, the time it takes this quarter-pounder to hit the ground.

24. Let $g \in C^1[a, b]$ and p be in (a, b) with $g(p) = p$ and $|g'(p)| > 1$. Show that there exists a $\delta > 0$ such that if $0 < |p_0 - p| < \delta$, then $|p_0 - p| < |p_1 - p|$. Thus, no matter how close the initial approximation p_0 is to p, the next iterate p_1 is farther away, so the fixed-point iteration does not converge if $p_0 \neq p$.

2.3 Newton's Method and Its Extensions

Isaac Newton (1641–1727) was one of the most brilliant scientists of all time. The late 17th century was a vibrant period for science and mathematics and Newton's work touched nearly every aspect of mathematics. His method for solving was introduced to find a root of the equation $y^3 - 2y - 5 = 0$. Although he demonstrated the method only for polynomials, it is clear that he realized its broader applications.

Newton's (or the *Newton-Raphson*) **method** is one of the most powerful and well-known numerical methods for solving a root-finding problem. There are many ways of introducing Newton's method.

Newton's Method

If we only want an algorithm, we can consider the technique graphically, as is often done in calculus. Another possibility is to derive Newton's method as a technique to obtain faster convergence than offered by other types of functional iteration, as is done in Section 2.4. A third means of introducing Newton's method, which is discussed next, is based on Taylor polynomials. We will see there that this particular derivation produces not only the method, but also a bound for the error of the approximation.

Suppose that $f \in C^2[a, b]$. Let $p_0 \in [a, b]$ be an approximation to p such that $f'(p_0) \neq 0$ and $|p - p_0|$ is "small." Consider the first Taylor polynomial for $f(x)$ expanded about p_0 and evaluated at $x = p$.

$$f(p) = f(p_0) + (p - p_0)f'(p_0) + \frac{(p - p_0)^2}{2} f''(\xi(p)),$$

where $\xi(p)$ lies between p and p_0. Since $f(p) = 0$, this equation gives

$$0 = f(p_0) + (p - p_0)f'(p_0) + \frac{(p - p_0)^2}{2} f''(\xi(p)).$$

Newton's method is derived by assuming that since $|p - p_0|$ is small, the term involving $(p - p_0)^2$ is much smaller, so

$$0 \approx f(p_0) + (p - p_0)f'(p_0).$$

Solving for p gives

$$p \approx p_0 - \frac{f(p_0)}{f'(p_0)} \equiv p_1.$$

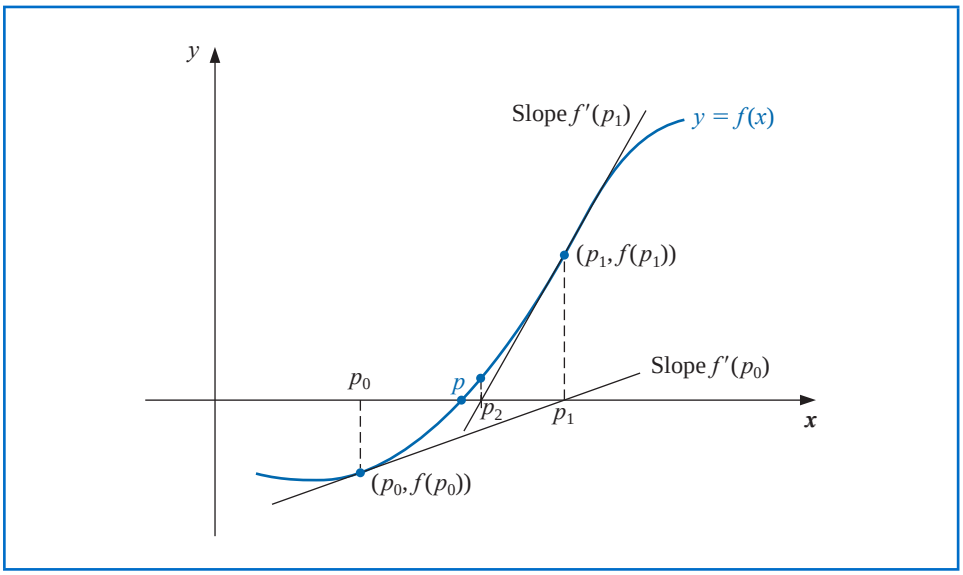
Joseph Raphson (1648–1715) gave a description of the method attributed to Isaac Newton in 1690, acknowledging Newton as the source of the discovery. Neither Newton nor Raphson explicitly used the derivative in their description since both considered only polynomials. Other mathematicians, particularly James Gregory (1636–1675), were aware of the underlying process at or before this time.

This sets the stage for Newton's method, which starts with an initial approximation p_0 and generates the sequence $\{p_n\}_{n=0}^{\infty}$, by

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{for } n \geq 1. \tag{2.7}$$

Figure 2.8 on page 68 illustrates how the approximations are obtained using successive tangents. (Also see Exercise 15.) Starting with the initial approximation p_0, the approximation p_1 is the x-intercept of the tangent line to the graph of f at $(p_0, f(p_0))$. The approximation p_2 is the x-intercept of the tangent line to the graph of f at $(p_1, f(p_1))$ and so on. Algorithm 2.3 follows this procedure.

Figure 2.8



ALGORITHM
2.3

Newton's

To find a solution to $f(x) = 0$ given an initial approximation p_0:

INPUT initial approximation p_0; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$.

Step 2 While $i \le N_0$ do Steps 3–6.

Step 3 Set $p = p_0 - f(p_0)/f'(p_0)$. (*Compute* p_i.)

Step 4 If $|p - p_0| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 5 Set $i = i + 1$.

Step 6 Set $p_0 = p$. (*Update* p_0.)

Step 7 OUTPUT ('The method failed after N_0 iterations, $N_0 =$', N_0);
(*The procedure was unsuccessful.*)
STOP. ■

The stopping-technique inequalities given with the Bisection method are applicable to Newton's method. That is, select a tolerance $\varepsilon > 0$, and construct $p_1, \ldots p_N$ until

$$|p_N - p_{N-1}| < \varepsilon, \tag{2.8}$$

$$\frac{|p_N - p_{N-1}|}{|p_N|} < \varepsilon, \quad p_N \neq 0, \tag{2.9}$$

or

$$|f(p_N)| < \varepsilon. \tag{2.10}$$

A form of Inequality (2.8) is used in Step 4 of Algorithm 2.3. Note that none of the inequalities (2.8), (2.9), or (2.10) give precise information about the actual error $|p_N - p|$. (See Exercises 16 and 17 in Section 2.1.)

Newton's method is a functional iteration technique with $p_n = g(p_{n-1})$, for which

$$g(p_{n-1}) = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{for } n \geq 1. \tag{2.11}$$

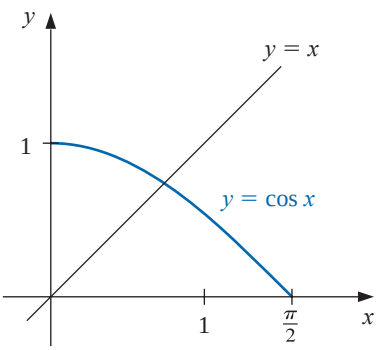
In fact, this is the functional iteration technique that was used to give the rapid convergence we saw in column (**e**) of Table 2.2 in Section 2.2.

It is clear from Equation (2.7) that Newton's method cannot be continued if $f'(p_{n-1}) = 0$ for some n. In fact, we will see that the method is most effective when f' is bounded away from zero near p.

Example 1 Consider the function $f(x) = \cos x - x = 0$. Approximate a root of f using **(a)** a fixed-point method, and **(b)** Newton's Method

Solution **(a)** A solution to this root-finding problem is also a solution to the fixed-point problem $x = \cos x$, and the graph in Figure 2.9 implies that a single fixed-point p lies in $[0, \pi/2]$.

Figure 2.9



Note that the variable in the trigonometric function is in radian measure, not degrees. This will always be the case unless specified otherwise.

Table 2.3

n	p_n
0	0.7853981635
1	0.7071067810
2	0.7602445972
3	0.7246674808
4	0.7487198858
5	0.7325608446
6	0.7434642113
7	0.7361282565

Table 2.4

	Newton's Method
n	p_n
0	0.7853981635
1	0.7395361337
2	0.7390851781
3	0.7390851332
4	0.7390851332

Table 2.3 shows the results of fixed-point iteration with $p_0 = \pi/4$. The best we could conclude from these results is that $p \approx 0.74$.

(b) To apply Newton's method to this problem we need $f'(x) = -\sin x - 1$. Starting again with $p_0 = \pi/4$, we generate the sequence defined, for $n \geq 1$, by

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f(p'_{n-1})} = p_{n-1} - \frac{\cos p_{n-1} - p_{n-1}}{-\sin p_{n-1} - 1}.$$

This gives the approximations in Table 2.4. An excellent approximation is obtained with $n = 3$. Because of the agreement of p_3 and p_4 we could reasonably expect this result to be accurate to the places listed. ■

Convergence using Newton's Method

Example 1 shows that Newton's method can provide extremely accurate approximations with very few iterations. For that example, only one iteration of Newton's method was needed to give better accuracy than 7 iterations of the fixed-point method. It is now time to examine Newton's method more carefully to discover why it is so effective.

The Taylor series derivation of Newton's method at the beginning of the section points out the importance of an accurate initial approximation. The crucial assumption is that the term involving $(p - p_0)^2$ is, by comparison with $|p - p_0|$, so small that it can be deleted. This will clearly be false unless p_0 is a good approximation to p. If p_0 is not sufficiently close to the actual root, there is little reason to suspect that Newton's method will converge to the root. However, in some instances, even poor initial approximations will produce convergence. (Exercises 20 and 21 illustrate some of these possibilities.)

The following convergence theorem for Newton's method illustrates the theoretical importance of the choice of p_0.

Theorem 2.6 Let $f \in C^2[a, b]$. If $p \in (a, b)$ is such that $f(p) = 0$ and $f'(p) \neq 0$, then there exists a $\delta > 0$ such that Newton's method generates a sequence $\{p_n\}_{n=1}^{\infty}$ converging to p for any initial approximation $p_0 \in [p - \delta, p + \delta]$. ■

Proof The proof is based on analyzing Newton's method as the functional iteration scheme $p_n = g(p_{n-1})$, for $n \geq 1$, with

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

Let k be in $(0, 1)$. We first find an interval $[p - \delta, p + \delta]$ that g maps into itself and for which $|g'(x)| \leq k$, for all $x \in (p - \delta, p + \delta)$.

Since f' is continuous and $f'(p) \neq 0$, part (**a**) of Exercise 29 in Section 1.1 implies that there exists a $\delta_1 > 0$, such that $f'(x) \neq 0$ for $x \in [p - \delta_1, p + \delta_1] \subseteq [a, b]$. Thus g is defined and continuous on $[p - \delta_1, p + \delta_1]$. Also

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

for $x \in [p - \delta_1, p + \delta_1]$, and, since $f \in C^2[a, b]$, we have $g \in C^1[p - \delta_1, p + \delta_1]$.

By assumption, $f(p) = 0$, so

$$g'(p) = \frac{f(p)f''(p)}{[f'(p)]^2} = 0.$$

Since g' is continuous and $0 < k < 1$, part (**b**) of Exercise 29 in Section 1.1 implies that there exists a δ, with $0 < \delta < \delta_1$, and

$$|g'(x)| \leq k, \quad \text{for all } x \in [p - \delta, p + \delta].$$

It remains to show that g maps $[p - \delta, p + \delta]$ into $[p - \delta, p + \delta]$. If $x \in [p - \delta, p + \delta]$, the Mean Value Theorem implies that for some number ξ between x and p, $|g(x) - g(p)| = |g'(\xi)||x - p|$. So

$$|g(x) - p| = |g(x) - g(p)| = |g'(\xi)||x - p| \leq k|x - p| < |x - p|.$$

Since $x \in [p - \delta, p + \delta]$, it follows that $|x - p| < \delta$ and that $|g(x) - p| < \delta$. Hence, g maps $[p - \delta, p + \delta]$ into $[p - \delta, p + \delta]$.

All the hypotheses of the Fixed-Point Theorem 2.4 are now satisfied, so the sequence $\{p_n\}_{n=1}^{\infty}$, defined by

$$p_n = g(p_{n-1}) = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{for } n \geq 1,$$

converges to p for any $p_0 \in [p - \delta, p + \delta]$. ■ ■ ■

Theorem 2.6 states that, under reasonable assumptions, Newton's method converges provided a sufficiently accurate initial approximation is chosen. It also implies that the constant k that bounds the derivative of g, and, consequently, indicates the speed of convergence of the method, decreases to 0 as the procedure continues. This result is important for the theory of Newton's method, but it is seldom applied in practice because it does not tell us how to determine δ.

In a practical application, an initial approximation is selected and successive approximations are generated by Newton's method. These will generally either converge quickly to the root, or it will be clear that convergence is unlikely.

The Secant Method

Newton's method is an extremely powerful technique, but it has a major weakness: the need to know the value of the derivative of f at each approximation. Frequently, $f'(x)$ is far more difficult and needs more arithmetic operations to calculate than $f(x)$.

To circumvent the problem of the derivative evaluation in Newton's method, we introduce a slight variation. By definition,

$$f'(p_{n-1}) = \lim_{x \to p_{n-1}} \frac{f(x) - f(p_{n-1})}{x - p_{n-1}}.$$

If p_{n-2} is close to p_{n-1}, then

$$f'(p_{n-1}) \approx \frac{f(p_{n-2}) - f(p_{n-1})}{p_{n-2} - p_{n-1}} = \frac{f(p_{n-1}) - f(p_{n-2})}{p_{n-1} - p_{n-2}}.$$

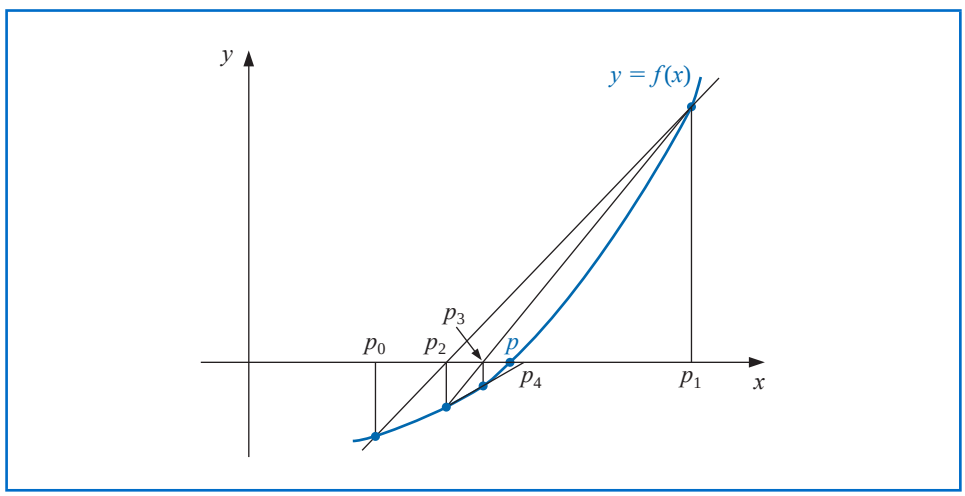
Using this approximation for $f'(p_{n-1})$ in Newton's formula gives

$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}. \tag{2.12}$$

The word secant is derived from the Latin word *secan*, which means to cut. The secant method uses a secant line, a line joining two points that cut the curve, to approximate a root.

This technique is called the **Secant method** and is presented in Algorithm 2.4. (See Figure 2.10.) Starting with the two initial approximations p_0 and p_1, the approximation p_2 is the x-intercept of the line joining $(p_0, f(p_0))$ and $(p_1, f(p_1))$. The approximation p_3 is the x-intercept of the line joining $(p_1, f(p_1))$ and $(p_2, f(p_2))$, and so on. Note that only one function evaluation is needed per step for the Secant method after p_2 has been determined. In contrast, each step of Newton's method requires an evaluation of both the function and its derivative.

Figure 2.10



ALGORITHM
2.4

Secant

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

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

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 2$;
$q_0 = f(p_0)$;
$q_1 = f(p_1)$.

Step 2 While $i \le N_0$ do Steps 3–6.

Step 3 Set $p = p_1 - q_1(p_1 - p_0)/(q_1 - q_0)$. (*Compute* p_i.)

Step 4 If $|p - p_1| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 5 Set $i = i + 1$.

Step 6 Set $p_0 = p_1$; (*Update* p_0, q_0, p_1, q_1.)
$q_0 = q_1$;
$p_1 = p$;
$q_1 = f(p)$.

Step 7 OUTPUT ('The method failed after N_0 iterations, $N_0 =$', N_0);
(*The procedure was unsuccessful.*)
STOP. ■

The next example involves a problem considered in Example 1, where we used Newton's method with $p_0 = \pi/4$.

Example 2 Use the Secant method to find a solution to $x = \cos x$, and compare the approximations with those given in Example 1 which applied Newton's method.

Solution In Example 1 we compared fixed-point iteration and Newton's method starting with the initial approximation $p_0 = \pi/4$. For the Secant method we need two initial approximations. Suppose we use $p_0 = 0.5$ and $p_1 = \pi/4$. Succeeding approximations are generated by the formula

$$p_n = p_{n-1} - \frac{(p_{n-1} - p_{n-2})(\cos p_{n-1} - p_{n-1})}{(\cos p_{n-1} - p_{n-1}) - (\cos p_{n-2} - p_{n-2})}, \quad \text{for } n \ge 2.$$

These give the results in Table 2.5. ■

Table 2.5

	Secant
n	p_n
0	0.5
1	0.7853981635
2	0.7363841388
3	0.7390581392
4	0.7390851493
5	0.7390851332

	Newton
n	p_n
0	0.7853981635
1	0.7395361337
2	0.7390851781
3	0.7390851332
4	0.7390851332

Comparing the results in Table 2.5 from the Secant method and Newton's method, we see that the Secant method approximation p_5 is accurate to the tenth decimal place, whereas Newton's method obtained this accuracy by p_3. For this example, the convergence of the Secant method is much faster than functional iteration but slightly slower than Newton's method. This is generally the case. (See Exercise 14 of Section 2.4.)

Newton's method or the Secant method is often used to refine an answer obtained by another technique, such as the Bisection method, since these methods require good first approximations but generally give rapid convergence.

The Method of False Position

Each successive pair of approximations in the Bisection method brackets a root p of the equation; that is, for each positive integer n, a root lies between a_n and b_n. This implies that, for each n, the Bisection method iterations satisfy

$$|p_n - p| < \frac{1}{2}|a_n - b_n|,$$

which provides an easily calculated error bound for the approximations.

Root bracketing is not guaranteed for either Newton's method or the Secant method. In Example 1, Newton's method was applied to $f(x) = \cos x - x$, and an approximate root was found to be 0.7390851332. Table 2.5 shows that this root is not bracketed by either p_0 and p_1 or p_1 and p_2. The Secant method approximations for this problem are also given in Table 2.5. In this case the initial approximations p_0 and p_1 bracket the root, but the pair of approximations p_3 and p_4 fail to do so.

The term *Regula Falsi*, literally a false rule or false position, refers to a technique that uses results that are known to be false, but in some specific manner, to obtain convergence to a true result. False position problems can be found on the Rhind papyrus, which dates from about 1650 B.C.E.

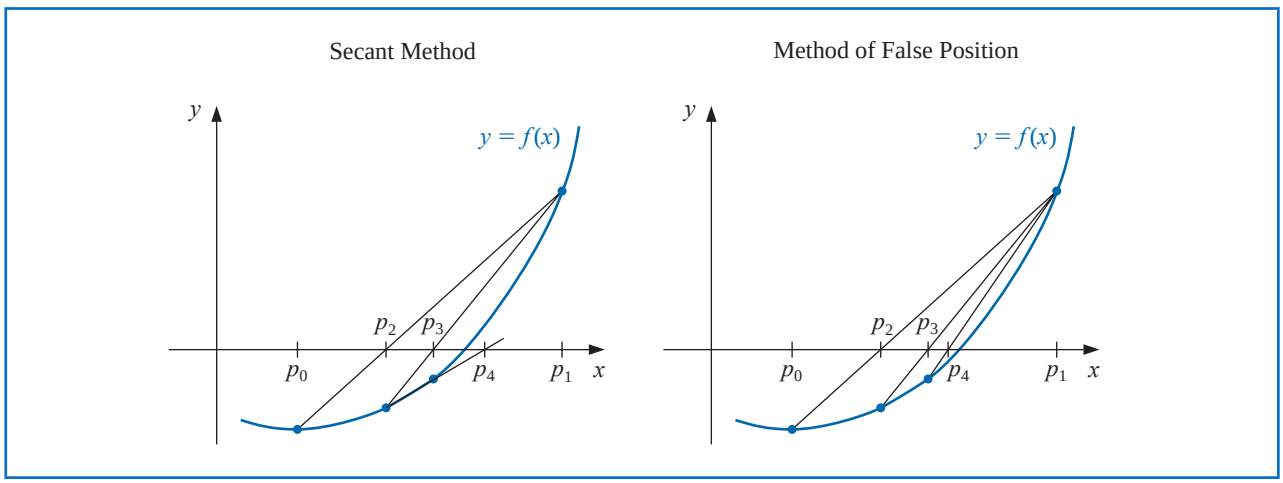
The **method of False Position** (also called *Regula Falsi*) generates approximations in the same manner as the Secant method, but it includes a test to ensure that the root is always bracketed between successive iterations. Although it is not a method we generally recommend, it illustrates how bracketing can be incorporated.

First choose initial approximations p_0 and p_1 with $f(p_0) \cdot f(p_1) < 0$. The approximation p_2 is chosen in the same manner as in the Secant method, as the x-intercept of the line joining $(p_0, f(p_0))$ and $(p_1, f(p_1))$. To decide which secant line to use to compute p_3, consider $f(p_2) \cdot f(p_1)$, or more correctly $\operatorname{sgn} f(p_2) \cdot \operatorname{sgn} f(p_1)$.

- If $\operatorname{sgn} f(p_2) \cdot \operatorname{sgn} f(p_1) < 0$, then p_1 and p_2 bracket a root. Choose p_3 as the x-intercept of the line joining $(p_1, f(p_1))$ and $(p_2, f(p_2))$.
- If not, choose p_3 as the x-intercept of the line joining $(p_0, f(p_0))$ and $(p_2, f(p_2))$, and then interchange the indices on p_0 and p_1.

In a similar manner, once p_3 is found, the sign of $f(p_3) \cdot f(p_2)$ determines whether we use p_2 and p_3 or p_3 and p_1 to compute p_4. In the latter case a relabeling of p_2 and p_1 is performed. The relabeling ensures that the root is bracketed between successive iterations. The process is described in Algorithm 2.5, and Figure 2.11 shows how the iterations can differ from those of the Secant method. In this illustration, the first three approximations are the same, but the fourth approximations differ.

Figure 2.11



ALGORITHM
2.5

False Position

To find a solution to $f(x) = 0$ given the continuous function f on the interval $[p_0, p_1]$ where $f(p_0)$ and $f(p_1)$ have opposite signs:

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

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 2$;
$q_0 = f(p_0)$;
$q_1 = f(p_1)$.

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

Step 3 Set $p = p_1 - q_1(p_1 - p_0)/(q_1 - q_0)$. (*Compute* p_i.)

Step 4 If $|p - p_1| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 5 Set $i = i + 1$;
$q = f(p)$.

Step 6 If $q \cdot q_1 < 0$ then set $p_0 = p_1$;
$q_0 = q_1$.

Step 7 Set $p_1 = p$;
$q_1 = q$.

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

Example 3 Use the method of False Position to find a solution to $x = \cos x$, and compare the approximations with those given in Example 1 which applied fixed-point iteration and Newton's method, and to those found in Example 2 which applied the Secant method.

Solution To make a reasonable comparison we will use the same initial approximations as in the Secant method, that is, $p_0 = 0.5$ and $p_1 = \pi/4$. Table 2.6 shows the results of the method of False Position applied to $f(x) = \cos x - x$ together with those we obtained using the Secant and Newton's methods. Notice that the False Position and Secant approximations agree through p_3 and that the method of False Position requires an additional iteration to obtain the same accuracy as the Secant method. ■

Table 2.6

	False Position	Secant	Newton
n	p_n	p_n	p_n
0	0.5	0.5	0.7853981635
1	0.7853981635	0.7853981635	0.7395361337
2	0.7363841388	0.7363841388	0.7390851781
3	0.7390581392	0.7390581392	0.7390851332
4	0.7390848638	0.7390851493	0.7390851332
5	0.7390851305	0.7390851332	
6	0.7390851332		

The added insurance of the method of False Position commonly requires more calculation than the Secant method, just as the simplification that the Secant method provides over Newton's method usually comes at the expense of additional iterations. Further examples of the positive and negative features of these methods can be seen by working Exercises 17 and 18.

Maple has Newton's method, the Secant method, and the method of False Position implemented in its *NumericalAnalysis* package. The options that were available for the Bisection method are also available for these techniques. For example, to generate the results in Tables 2.4, 2.5, and 2.6 we could use the commands

with(*Student*[*NumericalAnalysis*])

$f := \cos(x) - x$

$Newton\left(f, x = \frac{\pi}{4.0}, tolerance = 10^{-8}, output = sequence, maxiterations = 20\right)$

$Secant\left(f, x = \left[0.5, \frac{\pi}{4.0}\right], tolerance = 10^{-8}, output = sequence, maxiterations = 20\right)$

and

$FalsePosition\left(f, x = \left[0.5, \frac{\pi}{4.0}\right], tolerance = 10^{-8}, output = sequence, maxiterations = 20\right)$

EXERCISE SET 2.3

1. Let $f(x) = x^2 - 6$ and $p_0 = 1$. Use Newton's method to find p_2.

2. Let $f(x) = -x^3 - \cos x$ and $p_0 = -1$. Use Newton's method to find p_2. Could $p_0 = 0$ be used?

3. Let $f(x) = x^2 - 6$. With $p_0 = 3$ and $p_1 = 2$, find p_3.

a. Use the Secant method.

b. Use the method of False Position.

c. Which of **a.** or **b.** is closer to $\sqrt{6}$?

4. Let $f(x) = -x^3 - \cos x$. With $p_0 = -1$ and $p_1 = 0$, find p_3.

a. Use the Secant method.

b. Use the method of False Position.

5. Use Newton's method to find solutions accurate to within 10^{-4} for the following problems.

a. $x^3 - 2x^2 - 5 = 0, \quad [1, 4]$

b. $x^3 + 3x^2 - 1 = 0, \quad [-3, -2]$

c. $x - \cos x = 0, \quad [0, \pi/2]$

d. $x - 0.8 - 0.2 \sin x = 0, \quad [0, \pi/2]$

6. Use Newton's method to find solutions accurate to within 10^{-5} for the following problems.

a. $e^x + 2^{-x} + 2\cos x - 6 = 0 \quad$ for $1 \le x \le 2$

b. $\ln(x - 1) + \cos(x - 1) = 0 \quad$ for $1.3 \le x \le 2$

c. $2x \cos 2x - (x - 2)^2 = 0 \quad$ for $2 \le x \le 3$ and $3 \le x \le 4$

d. $(x - 2)^2 - \ln x = 0 \quad$ for $1 \le x \le 2$ and $e \le x \le 4$

e. $e^x - 3x^2 = 0 \quad$ for $0 \le x \le 1$ and $3 \le x \le 5$

f. $\sin x - e^{-x} = 0 \quad$ for $0 \le x \le 1 \;\; 3 \le x \le 4$ and $6 \le x \le 7$

7. Repeat Exercise 5 using the Secant method.

8. Repeat Exercise 6 using the Secant method.

9. Repeat Exercise 5 using the method of False Position.

10. Repeat Exercise 6 using the method of False Position.

11. Use all three methods in this Section to find solutions to within 10^{-5} for the following problems.

a. $3xe^x = 0 \quad$ for $1 \le x \le 2$

b. $2x + 3\cos x - e^x = 0 \quad$ for $0 \le x \le 1$

Suppose a certain population contains $N(0) = 1{,}000{,}000$ individuals initially, that 435,000 individuals immigrate into the community in the first year, and that $N(1) = 1{,}564{,}000$ individuals are present at the end of one year. To determine the birth rate of this population, we need to find λ in the equation

$$1{,}564{,}000 = 1{,}000{,}000e^{\lambda} + \frac{435{,}000}{\lambda}(e^{\lambda} - 1).$$

It is not possible to solve explicitly for λ in this equation, but numerical methods discussed in this chapter can be used to approximate solutions of equations of this type to an arbitrarily high accuracy. The solution to this particular problem is considered in Exercise 24 of Section 2.3.

2.1 The Bisection Method

In this chapter we consider one of the most basic problems of numerical approximation, the **root-finding problem**. This process involves finding a **root**, or solution, of an equation of the form $f(x) = 0$, for a given function f. A root of this equation is also called a **zero** of the function f.

The problem of finding an approximation to the root of an equation can be traced back at least to 1700 B.C.E. A cuneiform table in the Yale Babylonian Collection dating from that period gives a sexigesimal (base-60) number equivalent to 1.414222 as an approximation to $\sqrt{2}$, a result that is accurate to within 10^{-5}. This approximation can be found by applying a technique described in Exercise 19 of Section 2.2.

Bisection Technique

In computer science, the process of dividing a set continually in half to search for the solution to a problem, as the bisection method does, is known as a *binary search* procedure.

The first technique, based on the Intermediate Value Theorem, is called the **Bisection**, or **Binary-search, method**.

Suppose f is a continuous function defined on the interval $[a, b]$, with $f(a)$ and $f(b)$ of opposite sign. The Intermediate Value Theorem implies that a number p exists in (a, b) with $f(p) = 0$. Although the procedure will work when there is more than one root in the interval (a, b), we assume for simplicity that the root in this interval is unique. The method calls for a repeated halving (or bisecting) of subintervals of $[a, b]$ and, at each step, locating the half containing p.

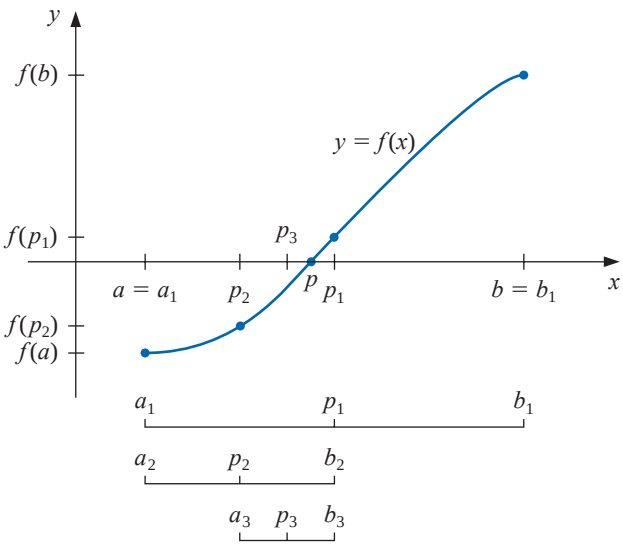
To begin, set $a_1 = a$ and $b_1 = b$, and let p_1 be the midpoint of $[a, b]$; that is,

$$p_1 = a_1 + \frac{b_1 - a_1}{2} = \frac{a_1 + b_1}{2}.$$

- If $f(p_1) = 0$, then $p = p_1$, and we are done.
- If $f(p_1) \neq 0$, then $f(p_1)$ has the same sign as either $f(a_1)$ or $f(b_1)$.
  - If $f(p_1)$ and $f(a_1)$ have the same sign, $p \in (p_1, b_1)$. Set $a_2 = p_1$ and $b_2 = b_1$.
  - If $f(p_1)$ and $f(a_1)$ have opposite signs, $p \in (a_1, p_1)$. Set $a_2 = a_1$ and $b_2 = p_1$.

Then reapply the process to the interval $[a_2, b_2]$. This produces the method described in Algorithm 2.1. (See Figure 2.1.)

Figure 2.1



ALGORITHM
2.1

Bisection

To find a solution to $f(x) = 0$ given the continuous function f on the interval $[a, b]$, where $f(a)$ and $f(b)$ have opposite signs:

INPUT endpoints a, b; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$;
$FA = f(a)$.

Step 2 While $i \le N_0$ do Steps 3–6.

Step 3 Set $p = a + (b - a)/2$; (*Compute* p_i.)
$FP = f(p)$.

Step 4 If $FP = 0$ or $(b - a)/2 < TOL$ then
OUTPUT (p); (*Procedure completed successfully.*)
STOP.

Step 5 Set $i = i + 1$.

Step 6 If $FA \cdot FP > 0$ then set $a = p$; (*Compute* a_i, b_i.)
$FA = FP$
else set $b = p$. (*FA is unchanged.*)

Step 7 OUTPUT ('Method failed after N_0 iterations, $N_0 =$', N_0);
(*The procedure was unsuccessful.*)
STOP.

■

Other stopping procedures can be applied in Step 4 of Algorithm 2.1 or in any of the iterative techniques in this chapter. For example, we can select a tolerance $\varepsilon > 0$ and generate $p_1, \ldots, p_N$ until one of the following conditions is met:

$$|p_N - p_{N-1}| < \varepsilon, \tag{2.1}$$

$$\frac{|p_N - p_{N-1}|}{|p_N|} < \varepsilon, \quad p_N \neq 0, \quad \text{or} \tag{2.2}$$

$$|f(p_N)| < \varepsilon. \tag{2.3}$$

Unfortunately, difficulties can arise using any of these stopping criteria. For example, there are sequences $\{p_n\}_{n=0}^{\infty}$ with the property that the differences $p_n - p_{n-1}$ converge to zero while the sequence itself diverges. (See Exercise 17.) It is also possible for $f(p_n)$ to be close to zero while p_n differs significantly from p. (See Exercise 16.) Without additional knowledge about f or p, Inequality (2.2) is the best stopping criterion to apply because it comes closest to testing relative error.

When using a computer to generate approximations, it is good practice to set an upper bound on the number of iterations. This eliminates the possibility of entering an infinite loop, a situation that can arise when the sequence diverges (and also when the program is incorrectly coded). This was done in Step 2 of Algorithm 2.1 where the bound N_0 was set and the procedure terminated if $i > N_0$.

Note that to start the Bisection Algorithm, an interval $[a, b]$ must be found with $f(a) \cdot f(b) < 0$. At each step the length of the interval known to contain a zero of f is reduced by a factor of 2; hence it is advantageous to choose the initial interval $[a, b]$ as small as possible. For example, if $f(x) = 2x^3 - x^2 + x - 1$, we have both

$$f(-4) \cdot f(4) < 0 \quad \text{and} \quad f(0) \cdot f(1) < 0,$$

so the Bisection Algorithm could be used on $[-4, 4]$ or on $[0, 1]$. Starting the Bisection Algorithm on $[0, 1]$ instead of $[-4, 4]$ will reduce by 3 the number of iterations required to achieve a specified accuracy.

The following example illustrates the Bisection Algorithm. The iteration in this example is terminated when a bound for the relative error is less than 0.0001. This is ensured by having

$$\frac{|p - p_n|}{\min\{|a_n|, |b_n|\}} < 10^{-4}.$$

Example 1 Show that $f(x) = x^3 + 4x^2 - 10 = 0$ has a root in $[1, 2]$, and use the Bisection method to determine an approximation to the root that is accurate to at least within 10^{-4}.

Solution Because $f(1) = -5$ and $f(2) = 14$ the Intermediate Value Theorem 1.11 ensures that this continuous function has a root in $[1, 2]$.

For the first iteration of the Bisection method we use the fact that at the midpoint of $[1, 2]$ we have $f(1.5) = 2.375 > 0$. This indicates that we should select the interval $[1, 1.5]$ for our second iteration. Then we find that $f(1.25) = -1.796875$ so our new interval becomes $[1.25, 1.5]$, whose midpoint is 1.375. Continuing in this manner gives the values in Table 2.1. After 13 iterations, $p_{13} = 1.365112305$ approximates the root p with an error

$$|p - p_{13}| < |b_{14} - a_{14}| = |1.365234375 - 1.365112305| = 0.000122070.$$

Since $|a_{14}| < |p|$, we have

$$\frac{|p - p_{13}|}{|p|} < \frac{|b_{14} - a_{14}|}{|a_{14}|} \leq 9.0 \times 10^{-5},$$

Table 2.1

n	a_n	b_n	p_n	$f(p_n)$
1	1.0	2.0	1.5	2.375
2	1.0	1.5	1.25	−1.79687
3	1.25	1.5	1.375	0.16211
4	1.25	1.375	1.3125	−0.84839
5	1.3125	1.375	1.34375	−0.35098
6	1.34375	1.375	1.359375	−0.09641
7	1.359375	1.375	1.3671875	0.03236
8	1.359375	1.3671875	1.36328125	−0.03215
9	1.36328125	1.3671875	1.365234375	0.000072
10	1.36328125	1.365234375	1.364257813	−0.01605
11	1.364257813	1.365234375	1.364746094	−0.00799
12	1.364746094	1.365234375	1.364990235	−0.00396
13	1.364990235	1.365234375	1.365112305	−0.00194

so the approximation is correct to at least within 10^{-4}. The correct value of p to nine decimal places is $p = 1.365230013$. Note that p_9 is closer to p than is the final approximation p_{13}. You might suspect this is true because $|f(p_9)| < |f(p_{13})|$, but we cannot be sure of this unless the true answer is known. ■

The Bisection method, though conceptually clear, has significant drawbacks. It is relatively slow to converge (that is, N may become quite large before $|p - p_N|$ is sufficiently small), and a good intermediate approximation might be inadvertently discarded. However, the method has the important property that it always converges to a solution, and for that reason it is often used as a starter for the more efficient methods we will see later in this chapter.

Theorem 2.1 Suppose that $f \in C[a, b]$ and $f(a) \cdot f(b) < 0$. The Bisection method generates a sequence $\{p_n\}_{n=1}^{\infty}$ approximating a zero p of f with

$$|p_n - p| \leq \frac{b-a}{2^n}, \quad \text{when} \quad n \geq 1.$$ ■

Proof For each $n \geq 1$, we have

$$b_n - a_n = \frac{1}{2^{n-1}}(b-a) \quad \text{and} \quad p \in (a_n, b_n).$$

Since $p_n = \frac{1}{2}(a_n + b_n)$ for all $n \geq 1$, it follows that

$$|p_n - p| \leq \frac{1}{2}(b_n - a_n) = \frac{b-a}{2^n}.$$ ■ ■ ■

Because

$$|p_n - p| \leq (b-a)\frac{1}{2^n},$$

the sequence $\{p_n\}_{n=1}^{\infty}$ converges to p with rate of convergence $O\left(\frac{1}{2^n}\right)$; that is,

$$p_n = p + O\left(\frac{1}{2^n}\right).$$

It is important to realize that Theorem 2.1 gives only a bound for approximation error and that this bound might be quite conservative. For example, this bound applied to the problem in Example 1 ensures only that

$$|p - p_9| \leq \frac{2-1}{2^9} \approx 2 \times 10^{-3},$$

but the actual error is much smaller:

$$|p - p_9| = |1.365230013 - 1.365234375| \approx 4.4 \times 10^{-6}.$$

Example 2 Determine the number of iterations necessary to solve $f(x) = x^3 + 4x^2 - 10 = 0$ with accuracy 10^{-3} using $a_1 = 1$ and $b_1 = 2$.

Solution We we will use logarithms to find an integer N that satisfies

$$|p_N - p| \leq 2^{-N}(b-a) = 2^{-N} < 10^{-3}.$$

Logarithms to any base would suffice, but we will use base-10 logarithms because the tolerance is given as a power of 10. Since $2^{-N} < 10^{-3}$ implies that $\log_{10} 2^{-N} < \log_{10} 10^{-3} = -3$, we have

$$-N \log_{10} 2 < -3 \quad \text{and} \quad N > \frac{3}{\log_{10} 2} \approx 9.96.$$

Hence, ten iterations will ensure an approximation accurate to within 10^{-3}.

Table 2.1 shows that the value of $p_9 = 1.365234375$ is accurate to within 10^{-4}. Again, it is important to keep in mind that the error analysis gives only a bound for the number of iterations. In many cases this bound is much larger than the actual number required. ■

Maple has a *NumericalAnalysis* package that implements many of the techniques we will discuss, and the presentation and examples in the package are closely aligned with this text. The Bisection method in this package has a number of options, some of which we will now consider. In what follows, Maple code is given in *black italic* type and Maple response in cyan.

Load the *NumericalAnalysis* package with the command

with(*Student*[*NumericalAnalysis*])

which gives access to the procedures in the package. Define the function with

$f := x^3 + 4x^2 - 10$

and use

Bisection $(f, x = [1, 2], tolerance = 0.005)$

Maple returns

1.363281250

Note that the value that is output is the same as p_8 in Table 2.1.

The sequence of bisection intervals can be output with the command

Bisection $(f, x = [1, 2], tolerance = 0.005, output = sequence)$

and Maple returns the intervals containing the solution together with the solution

[1., 2.], [1., 1.500000000], [1.250000000, 1.500000000], [1.250000000, 1.375000000],

[1.312500000, 1.375000000], [1.343750000, 1.375000000], [1.359375000, 1.375000000],

[1.359375000, 1.367187500], 1.363281250

The stopping criterion can also be based on relative error by choosing the option

Bisection $(f, x = [1, 2], tolerance = 0.005, stoppingcriterion = relative)$

Now Maple returns

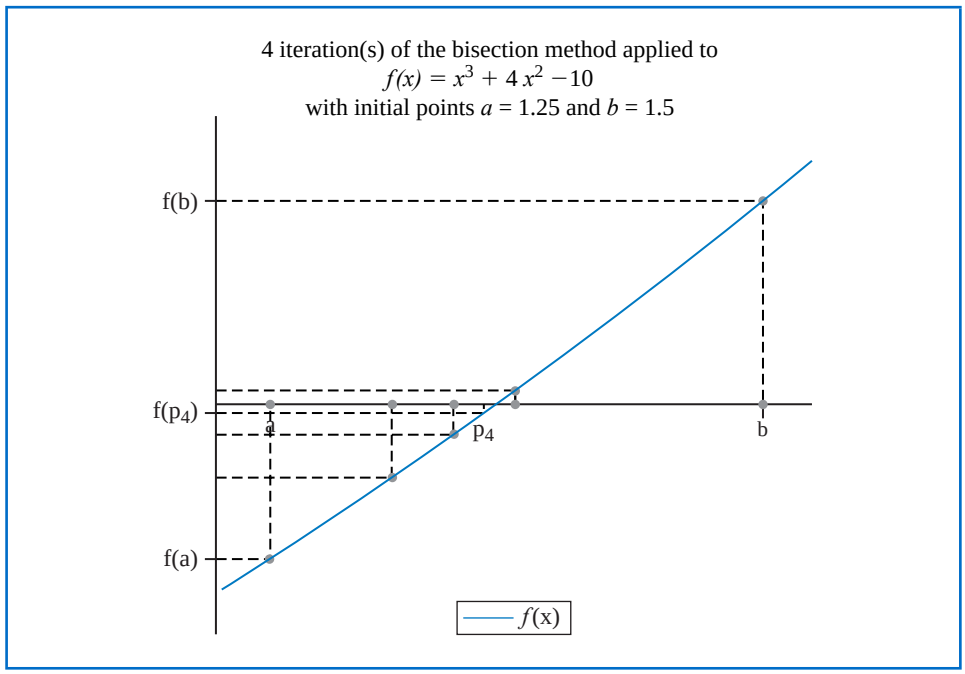
$$1.363281250$$

The option *output* = *plot* given in

$$Bisection\,(f, x = [1.25, 1.5], output = plot, tolerance = 0.02)$$

produces the plot shown in Figure 2.2.

Figure 2.2



We can also set the maximum number of iterations with the option *maxiterations* = . An error message will be displayed if the stated tolerance is not met within the specified number of iterations.

The results from Bisection method can also be obtained using the command Roots. For example,

$$Roots\left(f, x = [1.0, 2.0], method = bisection, tolerance = \frac{1}{100}, output = information\right)$$

uses the Bisection method to produce the information

n	a_n	b_n	p_n	$f\,(p_n)$	*relative error*
1	1.0	2.0	1.500000000	2.37500000	0.3333333333
2	1.0	1.500000000	1.250000000	−1.796875000	0.2000000000
3	1.250000000	1.500000000	1.375000000	0.16210938	0.09090909091
4	1.250000000	1.375000000	1.312500000	−0.848388672	0.04761904762
5	1.312500000	1.375000000	1.343750000	−0.350982668	0.02325581395
6	1.343750000	1.375000000	1.359375000	−0.096408842	0.01149425287
7	1.359375000	1.375000000	1.367187500	0.03235578	0.005714285714

The bound for the number of iterations for the Bisection method assumes that the calculations are performed using infinite-digit arithmetic. When implementing the method on a computer, we need to consider the effects of round-off error. For example, the computation of the midpoint of the interval $[a_n, b_n]$ should be found from the equation

$$p_n = a_n + \frac{b_n - a_n}{2} \quad \text{instead of} \quad p_n = \frac{a_n + b_n}{2}.$$

The first equation adds a small correction, $(b_n - a_n)/2$, to the known value a_n. When $b_n - a_n$ is near the maximum precision of the machine, this correction might be in error, but the error would not significantly affect the computed value of p_n. However, when $b_n - a_n$ is near the maximum precision of the machine, it is possible for $(a_n + b_n)/2$ to return a midpoint that is not even in the interval $[a_n, b_n]$.

The Latin word *signum* means "token" or "sign". So the signum function quite naturally returns the sign of a number (unless the number is 0).

As a final remark, to determine which subinterval of $[a_n, b_n]$ contains a root of f, it is better to make use of the **signum** function, which is defined as

$$\operatorname{sgn}(x) = \begin{cases} -1, & \text{if } x < 0, \\ 0, & \text{if } x = 0, \\ 1, & \text{if } x > 0. \end{cases}$$

The test

$$\operatorname{sgn}\,(f(a_n))\,\operatorname{sgn}\,(f(p_n)) < 0 \qquad \text{instead of} \qquad f(a_n)f(p_n) < 0$$

gives the same result but avoids the possibility of overflow or underflow in the multiplication of $f(a_n)$ and $f(p_n)$.

EXERCISE SET 2.1

1. Use the Bisection method to find p_3 for $f(x) = \sqrt{x} - \cos x$ on $[0, 1]$.

2. Let $f(x) = 3(x + 1)(x - \frac{1}{2})(x - 1)$. Use the Bisection method on the following intervals to find p_3.
a. $[-2, 1.5]$ **b.** $[-1.25, 2.5]$

3. Use the Bisection method to find solutions accurate to within 10^{-2} for $x^3 - 7x^2 + 14x - 6 = 0$ on each interval.
a. $[0, 1]$ **b.** $[1, 3.2]$ **c.** $[3.2, 4]$

4. Use the Bisection method to find solutions accurate to within 10^{-2} for $x^4 - 2x^3 - 4x^2 + 4x + 4 = 0$ on each interval.
a. $[-2, -1]$ **b.** $[0, 2]$ **c.** $[2, 3]$ **d.** $[-1, 0]$

5. Use the Bisection method to find solutions accurate to within 10^{-5} for the following problems.
a. $x - 2^{-x} = 0$ for $0 \le x \le 1$
b. $e^x - x^2 + 3x - 2 = 0$ for $0 \le x \le 1$
c. $2x\cos(2x) - (x + 1)^2 = 0$ for $-3 \le x \le -2$ and $-1 \le x \le 0$
d. $x\cos x - 2x^2 + 3x - 1 = 0$ for $0.2 \le x \le 0.3$ and $1.2 \le x \le 1.3$

6. Use the Bisection method to find solutions, accurate to within 10^{-5} for the following problems.
a. $3x - e^x = 0$ for $1 \le x \le 2$
b. $2x + 3\cos x - e^x = 0$ for $0 \le x \le 1$
c. $x^2 - 4x + 4 - \ln x = 0$ for $1 \le x \le 2$ and $2 \le x \le 4$
d. $x + 1 - 2\sin \pi x = 0$ for $0 \le x \le 0.5$ and $0.5 \le x \le 1$

7. **a.** Sketch the graphs of $y = x$ and $y = 2\sin x$.

b. Use the Bisection method to find an approximation to within 10^{-5} to the first positive value of x with $x = 2\sin x$.

8. **a.** Sketch the graphs of $y = x$ and $y = \tan x$.

b. Use the Bisection method to find an approximation to within 10^{-5} to the first positive value of x with $x = \tan x$.

9. **a.** Sketch the graphs of $y = e^x - 2$ and $y = \cos(e^x - 2)$.

b. Use the Bisection method to find an approximation to within 10^{-5} to a value in $[0.5, 1.5]$ with $e^x - 2 = \cos(e^x - 2)$.

10. Let $f(x) = (x+2)(x+1)^2x(x-1)^3(x-2)$. To which zero of f does the Bisection method converge when applied on the following intervals?

a. $[-1.5, 2.5]$ **b.** $[-0.5, 2.4]$ **c.** $[-0.5, 3]$ **d.** $[-3, -0.5]$

11. Let $f(x) = (x+2)(x+1)x(x-1)^3(x-2)$. To which zero of f does the Bisection method converge when applied on the following intervals?

a. $[-3, 2.5]$ **b.** $[-2.5, 3]$ **c.** $[-1.75, 1.5]$ **d.** $[-1.5, 1.75]$

12. Find an approximation to $\sqrt{3}$ correct to within 10^{-4} using the Bisection Algorithm. [*Hint:* Consider $f(x) = x^2 - 3$.]

13. Find an approximation to $\sqrt[3]{25}$ correct to within 10^{-4} using the Bisection Algorithm.

14. Use Theorem 2.1 to find a bound for the number of iterations needed to achieve an approximation with accuracy 10^{-3} to the solution of $x^3 + x - 4 = 0$ lying in the interval $[1, 4]$. Find an approximation to the root with this degree of accuracy.

15. Use Theorem 2.1 to find a bound for the number of iterations needed to achieve an approximation with accuracy 10^{-4} to the solution of $x^3 - x - 1 = 0$ lying in the interval $[1, 2]$. Find an approximation to the root with this degree of accuracy.

16. Let $f(x) = (x-1)^{10}$, $p = 1$, and $p_n = 1 + 1/n$. Show that $|f(p_n)| < 10^{-3}$ whenever $n > 1$ but that $|p - p_n| < 10^{-3}$ requires that $n > 1000$.

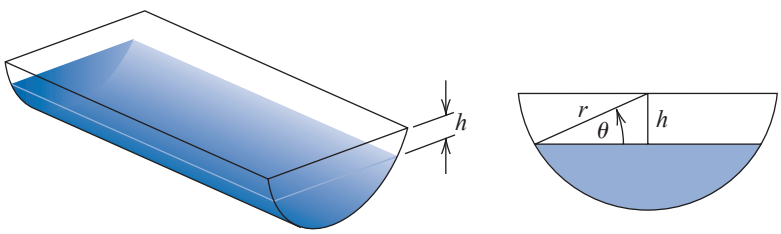
17. Let $\{p_n\}$ be the sequence defined by $p_n = \sum_{k=1}^{n} \frac{1}{k}$. Show that $\{p_n\}$ diverges even though $\lim_{n\to\infty}(p_n - p_{n-1}) = 0$.

18. The function defined by $f(x) = \sin \pi x$ has zeros at every integer. Show that when $-1 < a < 0$ and $2 < b < 3$, the Bisection method converges to

a. 0, if $a + b < 2$ **b.** 2, if $a + b > 2$ **c.** 1, if $a + b = 2$

19. A trough of length L has a cross section in the shape of a semicircle with radius r. (See the accompanying figure.) When filled with water to within a distance h of the top, the volume V of water is

$$V = L\left[0.5\pi r^2 - r^2 \arcsin(h/r) - h(r^2 - h^2)^{1/2}\right].$$



Suppose $L = 10$ ft, $r = 1$ ft, and $V = 12.4$ ft^3. Find the depth of water in the trough to within 0.01 ft.

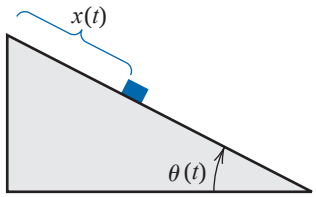
20. A particle starts at rest on a smooth inclined plane whose angle θ is changing at a constant rate

$$\frac{d\theta}{dt} = \omega < 0.$$

At the end of t seconds, the position of the object is given by

$$x(t) = -\frac{g}{2\omega^2}\left(\frac{e^{wt} - e^{-wt}}{2} - \sin \omega t\right).$$

Suppose the particle has moved 1.7 ft in 1 s. Find, to within 10^{-5}, the rate ω at which θ changes. Assume that $g = 32.17$ ft/s^2.



2.2 Fixed-Point Iteration

A *fixed point* for a function is a number at which the value of the function does not change when the function is applied.

Definition 2.2 The number p is a **fixed point** for a given function g if $g(p) = p$. ■

Fixed-point results occur in many areas of mathematics, and are a major tool of economists for proving results concerning equilibria. Although the idea behind the technique is old, the terminology was first used by the Dutch mathematician L. E. J. Brouwer (1882–1962) in the early 1900s.

In this section we consider the problem of finding solutions to fixed-point problems and the connection between the fixed-point problems and the root-finding problems we wish to solve. Root-finding problems and fixed-point problems are equivalent classes in the following sense:

- Given a root-finding problem $f(p) = 0$, we can define functions g with a fixed point at p in a number of ways, for example, as

$$g(x) = x - f(x) \quad \text{or as} \quad g(x) = x + 3f(x).$$

- Conversely, if the function g has a fixed point at p, then the function defined by

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

has a zero at p.

Although the problems we wish to solve are in the root-finding form, the fixed-point form is easier to analyze, and certain fixed-point choices lead to very powerful root-finding techniques.

We first need to become comfortable with this new type of problem, and to decide when a function has a fixed point and how the fixed points can be approximated to within a specified accuracy.

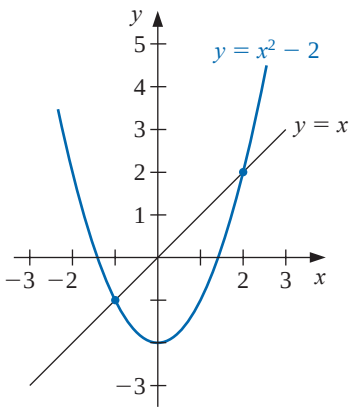
Example 1 Determine any fixed points of the function $g(x) = x^2 - 2$.

Solution A fixed point p for g has the property that

$$p = g(p) = p^2 - 2 \quad \text{which implies that} \quad 0 = p^2 - p - 2 = (p + 1)(p - 2).$$

A fixed point for g occurs precisely when the graph of $y = g(x)$ intersects the graph of $y = x$, so g has two fixed points, one at $p = -1$ and the other at $p = 2$. These are shown in Figure 2.3. ■

Figure 2.3



The following theorem gives sufficient conditions for the existence and uniqueness of a fixed point.

Theorem 2.3

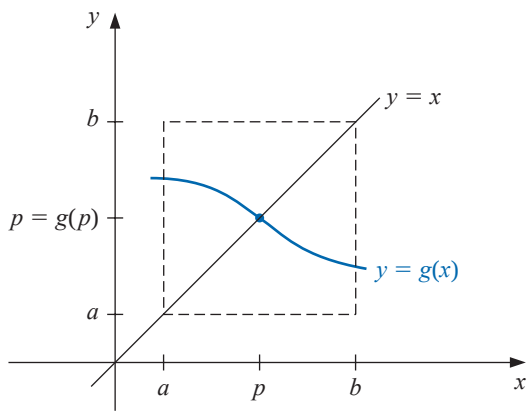
(i) If $g \in C[a, b]$ and $g(x) \in [a, b]$ for all $x \in [a, b]$, then g has at least one fixed point in $[a, b]$.

(ii) If, in addition, $g'(x)$ exists on (a, b) and a positive constant $k < 1$ exists with

$$|g'(x)| \leq k, \quad \text{for all } x \in (a, b),$$

then there is exactly one fixed point in $[a, b]$. (See Figure 2.4.) ■

Figure 2.4



Proof

(i) If $g(a) = a$ or $g(b) = b$, then g has a fixed point at an endpoint. If not, then $g(a) > a$ and $g(b) < b$. The function $h(x) = g(x) - x$ is continuous on $[a, b]$, with

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

The Intermediate Value Theorem implies that there exists $p \in (a, b)$ for which $h(p) = 0$. This number p is a fixed point for g because

$$0 = h(p) = g(p) - p \quad \text{implies that} \quad g(p) = p.$$

(ii) Suppose, in addition, that $|g'(x)| \leq k < 1$ and that p and q are both fixed points in $[a, b]$. If $p \neq q$, then the Mean Value Theorem implies that a number ξ exists between p and q, and hence in $[a, b]$, with

$$\frac{g(p) - g(q)}{p - q} = g'(\xi).$$

Thus

$$|p - q| = |g(p) - g(q)| = |g'(\xi)||p - q| \leq k|p - q| < |p - q|,$$

which is a contradiction. This contradiction must come from the only supposition, $p \neq q$. Hence, $p = q$ and the fixed point in $[a, b]$ is unique. ■ ■ ■

Example 2 Show that $g(x) = (x^2 - 1)/3$ has a unique fixed point on the interval $[-1, 1]$.

Solution The maximum and minimum values of $g(x)$ for x in $[-1, 1]$ must occur either when x is an endpoint of the interval or when the derivative is 0. Since $g'(x) = 2x/3$, the function g is continuous and $g'(x)$ exists on $[-1, 1]$. The maximum and minimum values of $g(x)$ occur at $x = -1$, $x = 0$, or $x = 1$. But $g(-1) = 0$, $g(1) = 0$, and $g(0) = -1/3$, so an absolute maximum for $g(x)$ on $[-1, 1]$ occurs at $x = -1$ and $x = 1$, and an absolute minimum at $x = 0$.

Moreover

$$|g'(x)| = \left|\frac{2x}{3}\right| \leq \frac{2}{3}, \quad \text{for all } x \in (-1, 1).$$

So g satisfies all the hypotheses of Theorem 2.3 and has a unique fixed point in $[-1, 1]$. ■

For the function in Example 2, the unique fixed point p in the interval $[-1, 1]$ can be determined algebraically. If

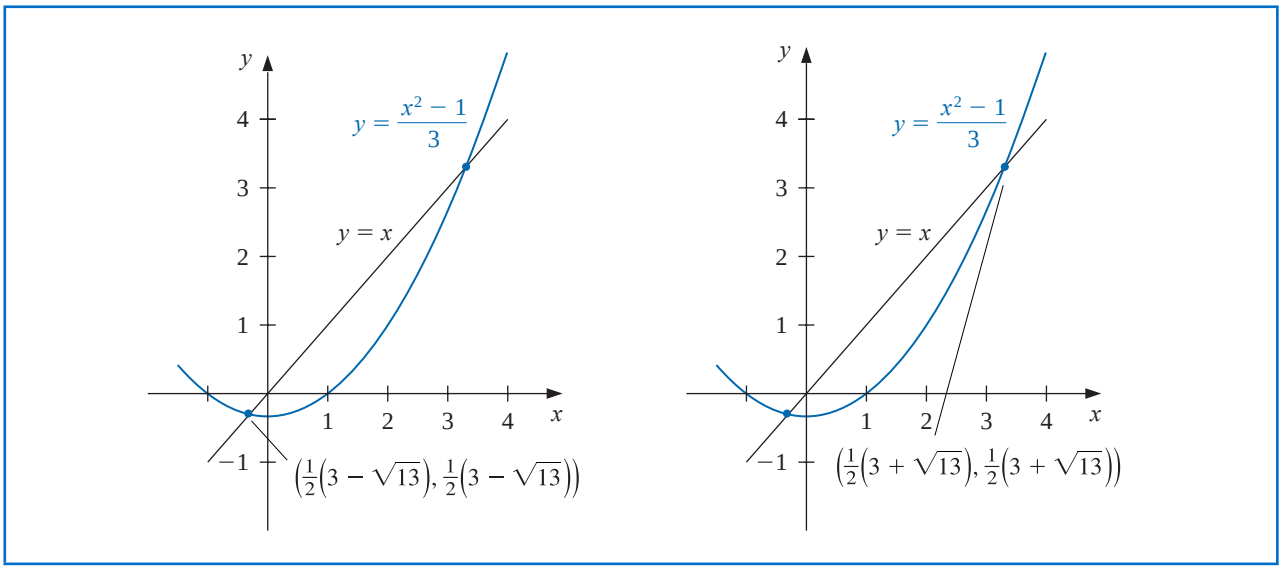
$$p = g(p) = \frac{p^2 - 1}{3}, \quad \text{then} \quad p^2 - 3p - 1 = 0,$$

which, by the quadratic formula, implies, as shown on the left graph in Figure 2.4, that

$$p = \frac{1}{2}(3 - \sqrt{13}).$$

Note that g also has a unique fixed point $p = \frac{1}{2}(3 + \sqrt{13})$ for the interval $[3, 4]$. However, $g(4) = 5$ and $g'(4) = \frac{8}{3} > 1$, so g does not satisfy the hypotheses of Theorem 2.3 on $[3, 4]$. This demonstrates that the hypotheses of Theorem 2.3 are sufficient to guarantee a unique fixed point but are not necessary. (See the graph on the right in Figure 2.5.)

Figure 2.5



Example 3 Show that Theorem 2.3 does not ensure a unique fixed point of $g(x) = 3^{-x}$ on the interval $[0, 1]$, even though a unique fixed point on this interval does exist.

Solution $g'(x) = -3^{-x} \ln 3 < 0$ on $[0, 1]$, the function g is strictly decreasing on $[0, 1]$. So

$$g(1) = \frac{1}{3} \le g(x) \le 1 = g(0), \quad \text{for} \quad 0 \le x \le 1.$$

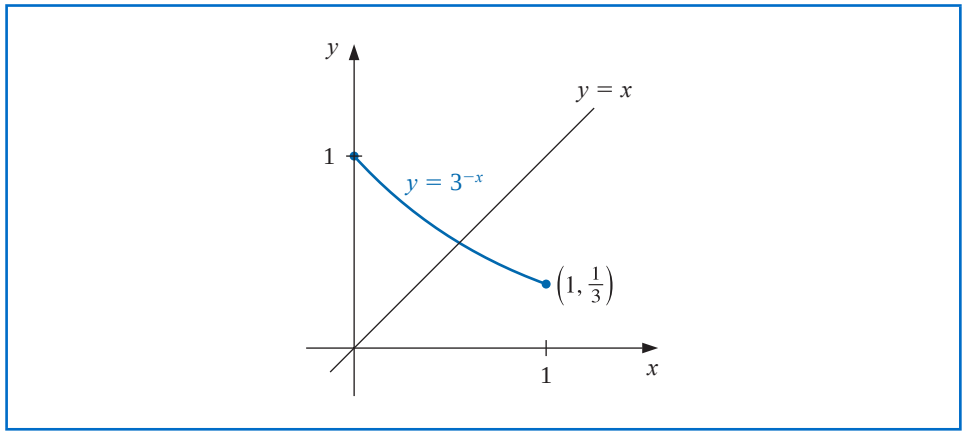
Thus, for $x \in [0, 1]$, we have $g(x) \in [0, 1]$. The first part of Theorem 2.3 ensures that there is at least one fixed point in $[0, 1]$.

However,

$$g'(0) = -\ln 3 = -1.098612289,$$

so $|g'(x)| \not\le 1$ on $(0, 1)$, and Theorem 2.3 cannot be used to determine uniqueness. But g is always decreasing, and it is clear from Figure 2.6 that the fixed point must be unique. ■

Figure 2.6



Fixed-Point Iteration

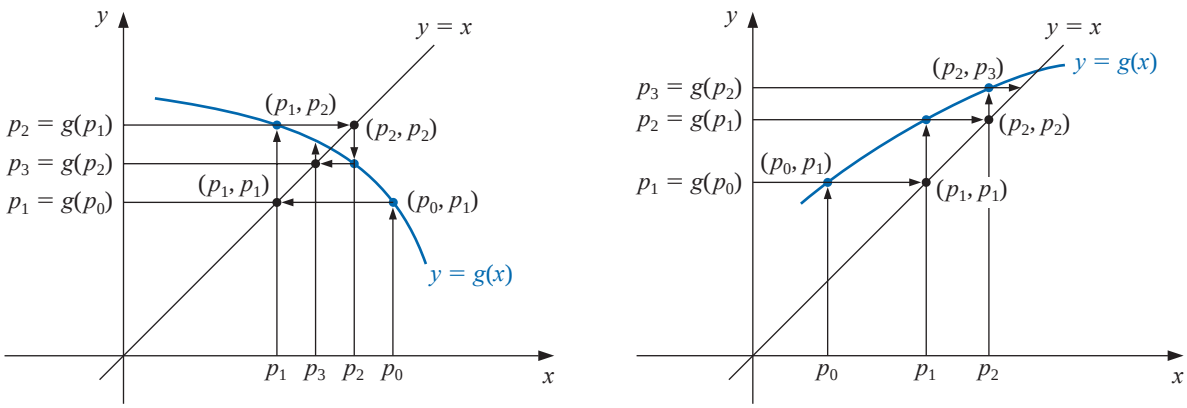
We cannot explicitly determine the fixed point in Example 3 because we have no way to solve for p in the equation $p = g(p) = 3^{-p}$. We can, however, determine approximations to this fixed point to any specified degree of accuracy. We will now consider how this can be done.

To approximate the fixed point of a function g, we choose an initial approximation p_0 and generate the sequence $\{p_n\}_{n=0}^{\infty}$ by letting $p_n = g(p_{n-1})$, for each $n \geq 1$. If the sequence converges to p and g is continuous, then

$$p = \lim_{n\to\infty} p_n = \lim_{n\to\infty} g(p_{n-1}) = g\left(\lim_{n\to\infty} p_{n-1}\right) = g(p),$$

and a solution to $x = g(x)$ is obtained. This technique is called **fixed-point**, or **functional iteration**. The procedure is illustrated in Figure 2.7 and detailed in Algorithm 2.2.

Figure 2.7



ALGORITHM
2.2

Fixed-Point Iteration

To find a solution to $p = g(p)$ given an initial approximation p_0:

INPUT initial approximation p_0; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$.

Step 2 While $i \leq N_0$ do Steps 3–6.

Step 3 Set $p = g(p_0)$. (*Compute* p_i.)

Step 4 If $|p - p_0| < TOL$ then
OUTPUT (p); (*The procedure was successful.*)
STOP.

Step 5 Set $i = i + 1$.

Step 6 Set $p_0 = p$. (*Update* p_0.)

Step 7 OUTPUT ('The method failed after N_0 iterations, $N_0 =$', N_0);
(*The procedure was unsuccessful.*)
STOP. ■

The following illustrates some features of functional iteration.

Illustration The equation $x^3 + 4x^2 - 10 = 0$ has a unique root in $[1, 2]$. There are many ways to change the equation to the fixed-point form $x = g(x)$ using simple algebraic manipulation. For example, to obtain the function g described in part (c), we can manipulate the equation $x^3 + 4x^2 - 10 = 0$ as follows:

$$4x^2 = 10 - x^3, \quad \text{so} \quad x^2 = \frac{1}{4}(10 - x^3), \quad \text{and} \quad x = \pm\frac{1}{2}(10 - x^3)^{1/2}.$$

To obtain a positive solution, $g_3(x)$ is chosen. It is not important for you to derive the functions shown here, but you should verify that the fixed point of each is actually a solution to the original equation, $x^3 + 4x^2 - 10 = 0$.

(a) $x = g_1(x) = x - x^3 - 4x^2 + 10$

(b) $x = g_2(x) = \left(\frac{10}{x} - 4x\right)^{1/2}$

(c) $x = g_3(x) = \frac{1}{2}(10 - x^3)^{1/2}$

(d) $x = g_4(x) = \left(\frac{10}{4 + x}\right)^{1/2}$

(e) $x = g_5(x) = x - \frac{x^3 + 4x^2 - 10}{3x^2 + 8x}$

With $p_0 = 1.5$, Table 2.2 lists the results of the fixed-point iteration for all five choices of g.

Table 2.2

n	(a)	(b)	(c)	(d)	(e)
0	1.5	1.5	1.5	1.5	1.5
1	−0.875	0.8165	1.286953768	1.348399725	1.373333333
2	6.732	2.9969	1.402540804	1.367376372	1.365262015
3	−469.7	$(-8.65)^{1/2}$	1.345458374	1.364957015	1.365230014
4	1.03×10^8		1.375170253	1.365264748	1.365230013
5			1.360094193	1.365225594	
6			1.367846968	1.365230576	
7			1.363887004	1.365229942	
8			1.365916734	1.365230022	
9			1.364878217	1.365230012	
10			1.365410062	1.365230014	
15			1.365223680	1.365230013	
20			1.365230236		
25			1.365230006		
30			1.365230013		

The actual root is 1.365230013, as was noted in Example 1 of Section 2.1. Comparing the results to the Bisection Algorithm given in that example, it can be seen that excellent results have been obtained for choices (c), (d), and (e) (the Bisection method requires 27 iterations for this accuracy). It is interesting to note that choice (a) was divergent and that (b) became undefined because it involved the square root of a negative number. □

Although the various functions we have given are fixed-point problems for the same root-finding problem, they differ vastly as techniques for approximating the solution to the root-finding problem. Their purpose is to illustrate what needs to be answered:

- Question: How can we find a fixed-point problem that produces a sequence that reliably and rapidly converges to a solution to a given root-finding problem?

The following theorem and its corollary give us some clues concerning the paths we should pursue and, perhaps more importantly, some we should reject.

Theorem 2.4 **(Fixed-Point Theorem)**

Let $g \in C[a, b]$ be such that $g(x) \in [a, b]$, for all x in $[a, b]$. Suppose, in addition, that g' exists on (a, b) and that a constant $0 < k < 1$ exists with

$$|g'(x)| \leq k, \quad \text{for all } x \in (a, b).$$

Then for any number p_0 in $[a, b]$, the sequence defined by

$$p_n = g(p_{n-1}), \quad n \geq 1,$$

converges to the unique fixed point p in $[a, b]$. ■

Proof Theorem 2.3 implies that a unique point p exists in $[a, b]$ with $g(p) = p$. Since g maps $[a, b]$ into itself, the sequence $\{p_n\}_{n=0}^{\infty}$ is defined for all $n \geq 0$, and $p_n \in [a, b]$ for all n. Using the fact that $|g'(x)| \leq k$ and the Mean Value Theorem 1.8, we have, for each n,

$$|p_n - p| = |g(p_{n-1}) - g(p)| = |g'(\xi_n)||p_{n-1} - p| \leq k|p_{n-1} - p|,$$

where $\xi_n \in (a, b)$. Applying this inequality inductively gives

$$|p_n - p| \leq k|p_{n-1} - p| \leq k^2|p_{n-2} - p| \leq \cdots \leq k^n|p_0 - p|. \tag{2.4}$$

Since $0 < k < 1$, we have $\lim_{n\to\infty} k^n = 0$ and

$$\lim_{n\to\infty} |p_n - p| \leq \lim_{n\to\infty} k^n|p_0 - p| = 0.$$

Hence $\{p_n\}_{n=0}^{\infty}$ converges to p. ■ ■ ■

Corollary 2.5 If g satisfies the hypotheses of Theorem 2.4, then bounds for the error involved in using p_n to approximate p are given by

$$|p_n - p| \leq k^n \max\{p_0 - a, b - p_0\} \tag{2.5}$$

and

$$|p_n - p| \leq \frac{k^n}{1 - k}|p_1 - p_0|, \quad \text{for all} \quad n \geq 1. \tag{2.6}$$

■

Proof Because $p \in [a, b]$, the first bound follows from Inequality (2.4):

$$|p_n - p| \leq k^n|p_0 - p| \leq k^n \max\{p_0 - a, b - p_0\}.$$

For $n \geq 1$, the procedure used in the proof of Theorem 2.4 implies that

$$|p_{n+1} - p_n| = |g(p_n) - g(p_{n-1})| \leq k|p_n - p_{n-1}| \leq \cdots \leq k^n|p_1 - p_0|.$$

Thus for $m > n \geq 1$,

$$\begin{aligned}|p_m - p_n| &= |p_m - p_{m-1} + p_{m-1} - \cdots + p_{n+1} - p_n| \\ &\leq |p_m - p_{m-1}| + |p_{m-1} - p_{m-2}| + \cdots + |p_{n+1} - p_n| \\ &\leq k^{m-1}|p_1 - p_0| + k^{m-2}|p_1 - p_0| + \cdots + k^n|p_1 - p_0| \\ &= k^n|p_1 - p_0|\left(1 + k + k^2 + \cdots + k^{m-n-1}\right).\end{aligned}$$

By Theorem 2.3, $\lim_{m\to\infty} p_m = p$, so

$$|p - p_n| = \lim_{m\to\infty} |p_m - p_n| \leq \lim_{m\to\infty} k^n|p_1 - p_0| \sum_{i=0}^{m-n-1} k^i \leq k^n|p_1 - p_0| \sum_{i=0}^{\infty} k^i.$$

But $\sum_{i=0}^{\infty} k^i$ is a geometric series with ratio k and $0 < k < 1$. This sequence converges to $1/(1-k)$, which gives the second bound:

$$|p - p_n| \leq \frac{k^n}{1-k}|p_1 - p_0|. \qquad \blacksquare\ \blacksquare\ \blacksquare$$

Both inequalities in the corollary relate the rate at which $\{p_n\}_{n=0}^{\infty}$ converges to the bound k on the first derivative. The rate of convergence depends on the factor k^n. The smaller the value of k, the faster the convergence, which may be very slow if k is close to 1.

Illustration Let us reconsider the various fixed-point schemes described in the preceding illustration in light of the Fixed-point Theorem 2.4 and its Corollary 2.5.

(a) For $g_1(x) = x - x^3 - 4x^2 + 10$, we have $g_1(1) = 6$ and $g_1(2) = -12$, so g_1 does not map $[1, 2]$ into itself. Moreover, $g_1'(x) = 1 - 3x^2 - 8x$, so $|g_1'(x)| > 1$ for all x in $[1, 2]$. Although Theorem 2.4 does not guarantee that the method must fail for this choice of g, there is no reason to expect convergence.

(b) With $g_2(x) = [(10/x) - 4x]^{1/2}$, we can see that g_2 does not map $[1, 2]$ into $[1, 2]$, and the sequence $\{p_n\}_{n=0}^{\infty}$ is not defined when $p_0 = 1.5$. Moreover, there is no interval containing $p \approx 1.365$ such that $|g_2'(x)| < 1$, because $|g_2'(p)| \approx 3.4$. There is no reason to expect that this method will converge.

(c) For the function $g_3(x) = \frac{1}{2}(10 - x^3)^{1/2}$, we have

$$g_3'(x) = -\frac{3}{4}x^2(10 - x^3)^{-1/2} < 0 \quad \text{on } [1, 2],$$

so g_3 is strictly decreasing on $[1, 2]$. However, $|g_3'(2)| \approx 2.12$, so the condition $|g_3'(x)| \leq k < 1$ fails on $[1, 2]$. A closer examination of the sequence $\{p_n\}_{n=0}^{\infty}$ starting with $p_0 = 1.5$ shows that it suffices to consider the interval $[1, 1.5]$ instead of $[1, 2]$. On this interval it is still true that $g_3'(x) < 0$ and g_3 is strictly decreasing, but, additionally,

$$1 < 1.28 \approx g_3(1.5) \leq g_3(x) \leq g_3(1) = 1.5,$$

for all $x \in [1, 1.5]$. This shows that g_3 maps the interval $[1, 1.5]$ into itself. It is also true that $|g_3'(x)| \leq |g_3'(1.5)| \approx 0.66$ on this interval, so Theorem 2.4 confirms the convergence of which we were already aware.

(d) For $g_4(x) = (10/(4 + x))^{1/2}$, we have

$$|g_4'(x)| = \left|\frac{-5}{\sqrt{10}(4 + x)^{3/2}}\right| \leq \frac{5}{\sqrt{10}(5)^{3/2}} < 0.15, \quad \text{for all} \quad x \in [1, 2].$$

The bound on the magnitude of $g_4'(x)$ is much smaller than the bound (found in (**c**)) on the magnitude of $g_3'(x)$, which explains the more rapid convergence using g_4.

(e) The sequence defined by

$$g_5(x) = x - \frac{x^3 + 4x^2 - 10}{3x^2 + 8x}$$

converges much more rapidly than our other choices. In the next sections we will see where this choice came from and why it is so effective. □

From what we have seen,

- Question: How can we find a fixed-point problem that produces a sequence that reliably and rapidly converges to a solution to a given root-finding problem?

might have

- Answer: Manipulate the root-finding problem into a fixed point problem that satisfies the conditions of Fixed-Point Theorem 2.4 and has a derivative that is as small as possible near the fixed point.

In the next sections we will examine this in more detail.

Maple has the fixed-point algorithm implemented in its *NumericalAnalysis* package. The options for the Bisection method are also available for fixed-point iteration. We will show only one option. After accessing the package using *with*(*Student*[*NumericalAnalysis*]): we enter the function

$$g := x - \frac{(x^3 + 4x^2 - 10)}{3x^2 + 8x}$$

and Maple returns

$$x - \frac{x^3 + 4x^2 - 10}{3x^2 + 8x}$$

Enter the command

FixedPointIteration(*fixedpointiterator* = *g*, $x = 1.5$, tolerance $= 10^{-8}$, *output* = *sequence*, *maxiterations* $= 20$)

and Maple returns

$$1.5, 1.373333333, 1.365262015, 1.365230014, 1.365230013$$

EXERCISE SET 2.2

1. Use algebraic manipulation to show that each of the following functions has a fixed point at p precisely when $f(p) = 0$, where $f(x) = x^4 + 2x^2 - x - 3$.

a. $g_1(x) = \left(3 + x - 2x^2\right)^{1/4}$ **b.** $g_2(x) = \left(\frac{x + 3 - x^4}{2}\right)^{1/2}$

This produces

$$0.99995000042$$

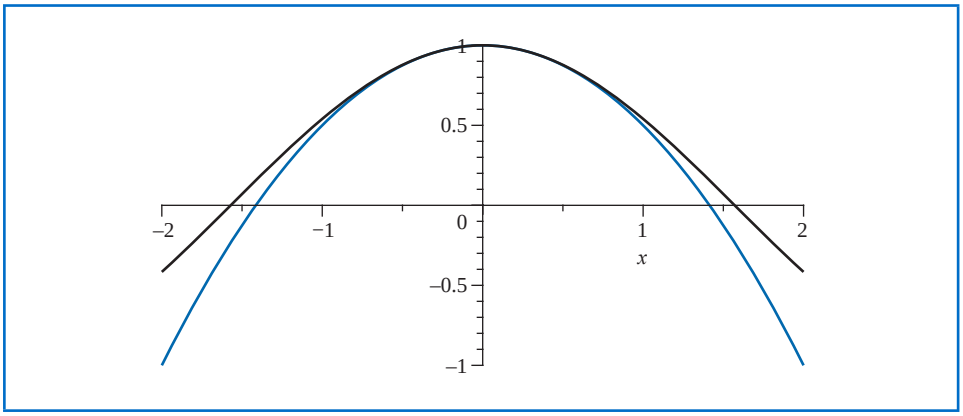
$$0.99995000000$$

To show both the function (in black) and the polynomial (in cyan) near $x_0 = 0$, we enter

$plot((f, p3), x = -2 \ldots 2)$

and obtain the Maple plot shown in Figure 1.11.

Figure 1.11



The integrals of f and the polynomial are given by

$q1 := int(f, x = 0 \ldots 0.1); q2 := int(p3, x = 0 \ldots 0.1)$

$$0.099833416647$$

$$0.099833333333$$

We assigned the names $q1$ and $q2$ to these values so that we could easily determine the error with the command

$err := |q1 - q2|$

$$8.3314\ 10^{-8}$$

There is an alternate method for generating the Taylor polynomials within the *NumericalAnalysis* subpackage of Maple's *Student* package. This subpackage will be discussed in Chapter 2.

EXERCISE SET 1.1

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

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

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