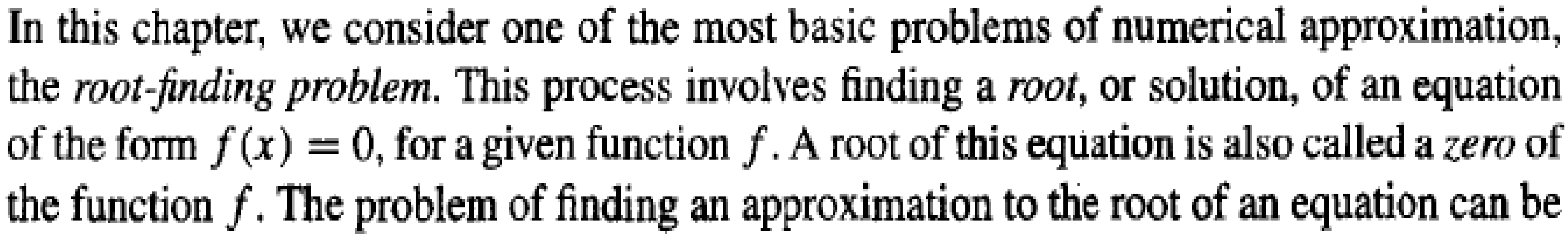
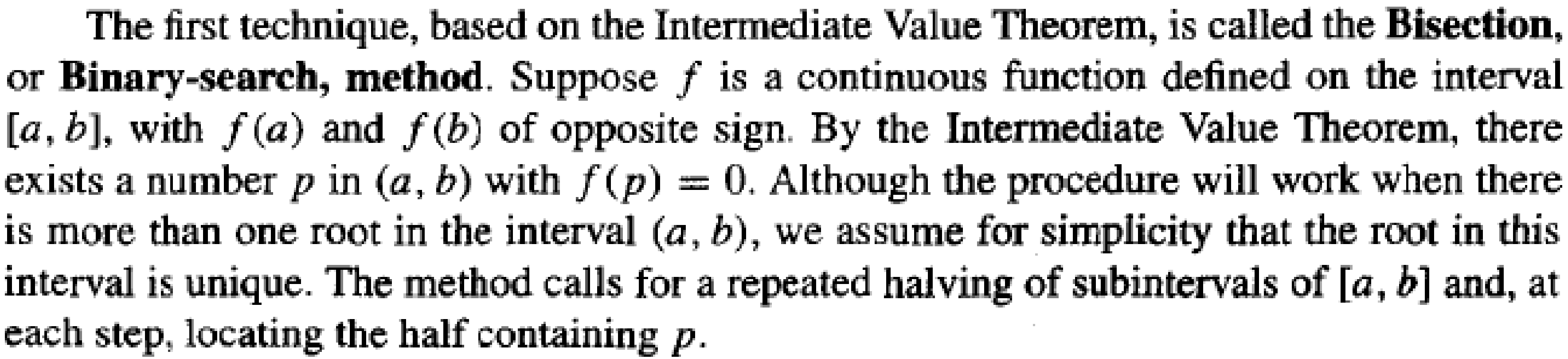
University of Science and Technology Beijing

Chap2 Solutions of Equations in One Variable

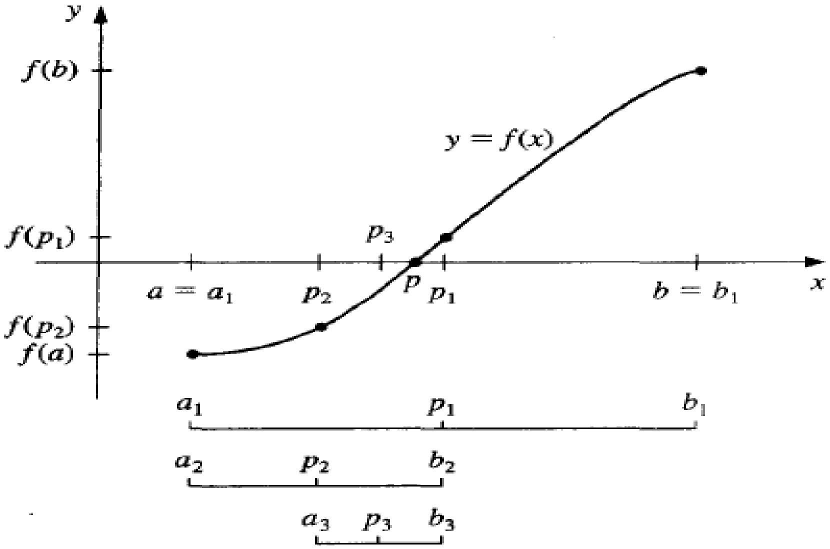


In this chapter, we consider one of the most basic problems of numerical approximation, the *root-finding problem*. This process involves finding a *root*, or solution, of an equation of the form $f(x) = 0$, for a given function f. A root of this equation is also called a *zero* of the function f. The problem of finding an approximation to the root of an equation can be traced back at least as far as 1700 B. C



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. By the Intermediate Value Theorem, there exists a number p 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 of subintervals of $[a, b]$ and, at each step, locating the half containing p.

2.1 The Bisection Method



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 $f(b_1)$. When $f(p_1)$ and $f(a_1)$ have the same sign, $p \in (p_1, b_1)$, and we set $a_2 = p_1$ and $b_2 = b_1$. When $f(p_1)$ and $f(a_1)$ have opposite signs, $p \in (a_1, p_1)$, and we set $a_2 = a_1$ and $b_2 = p_1$. We then reapply the process to the interval $[a_2, b_2]$. This produces the method described in Algorithm 2.1. (See Figure 2.1.)

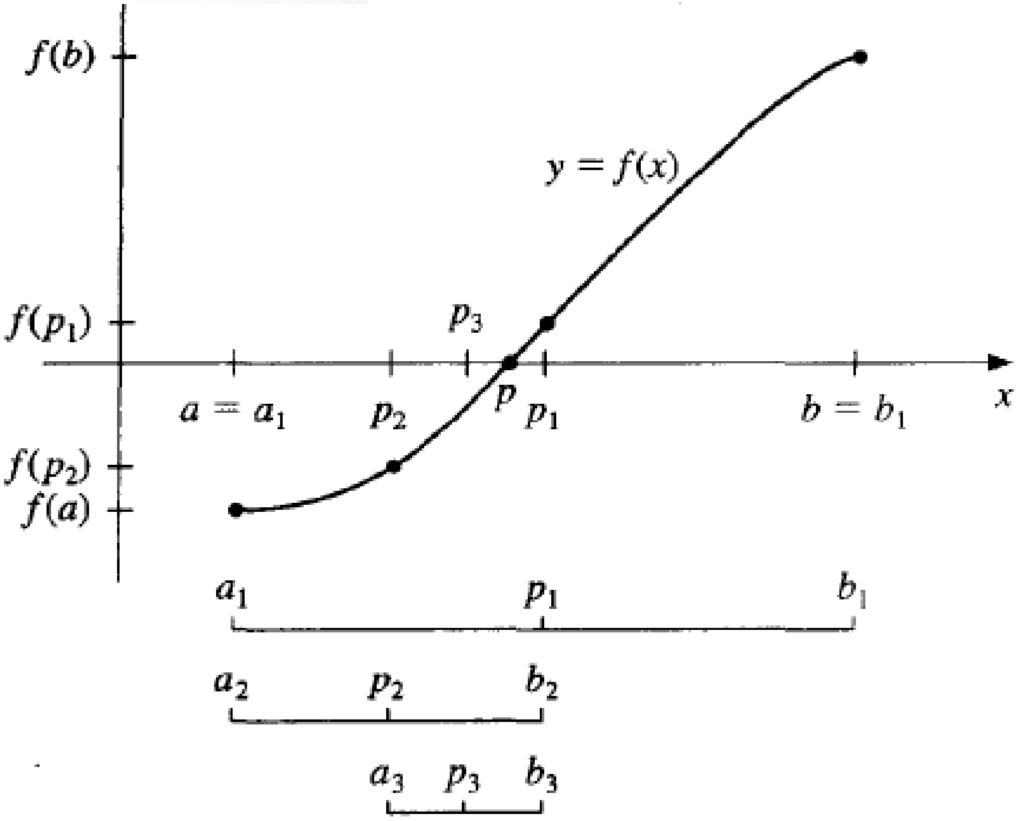


Figure 2.1

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 $i \leq N_0$ 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 $b = p$.

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 $\epsilon > 0$ and generate $p_1, \ldots, p_N$ until one of the following conditions is met:

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

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

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

When using a computer to generate approximations, it is good practice to set an upper bound on the number of iterations. This will eliminate 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 either of the intervals $[-4, 4]$ or $[0, 1]$. 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.

EXAMPLE 1 The equation $f(x) = x^3 + 4x^2 - 10 = 0$ has a root in $[1, 2]$ since $f(1) = -5$ and $f(2) = 14$. The Bisection Algorithm gives the values in Table 2.1.

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

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 $|a_{14}| < |p|$,

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

so the approximation is correct to at least four significant digits. 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 since $|f(p_9)| < |f(p_{13})|$, but we cannot be sure of this unless the true answer is known. ■

The Bisection method, though conceptually clear, has significant drawbacks. It is slow to converge (that is, N may become quite large before $|p - p_N|$ is sufficiently small), and a good intermediate approximation can 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 present 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

$$|p_n - p| \le \frac{b-a}{2^n}, \quad \text{when} \quad n \ge 1. \qquad \blacksquare$$

Proof For each $n \ge 1$, 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 $n \ge 1$, it follows that

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

Since

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

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 may 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:

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

To 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 $b_1 = 2$ requires finding an integer N that satisfies

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

To determine N we will use logarithms. Although logarithms to any base would suffice, we will use base-10 logarithms since 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 on page 51 shows that the value of $p_9 = 1.365234375$ is accurate to within 10^{-4}. Again, it

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, consideration must be given to the effects of roundoff error. For example, the computation of the midpoint of the interval $[a_n, b_n]$ should be found from the equation

$$p_n = a_n + \frac{b_n - a_n}{2}$$

instead of from the algebraically equivalent equation

$$p_n = \frac{a_n + b_n}{2}.$$

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 $[a_n, b_n]$.

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 $f(p_n)$.

2.2 Fixed-Point Iteration

A number p is a **fixed point** for a given function g if $g(p) = p$. 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 $g(x) = x - f(x)$ or as $g(x) = x + 3f(x)$. 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.

The function $g(x) = x^2 - 2$, for $-2 \leq x \leq 3$, has fixed points at $x = -1$ and $x = 2$ since

$$g(-1) = (-1)^2 - 2 = -1 \quad \text{and} \quad g(2) = 2^2 - 2 = 2.$$

This can be seen in Figure 2.2. ■

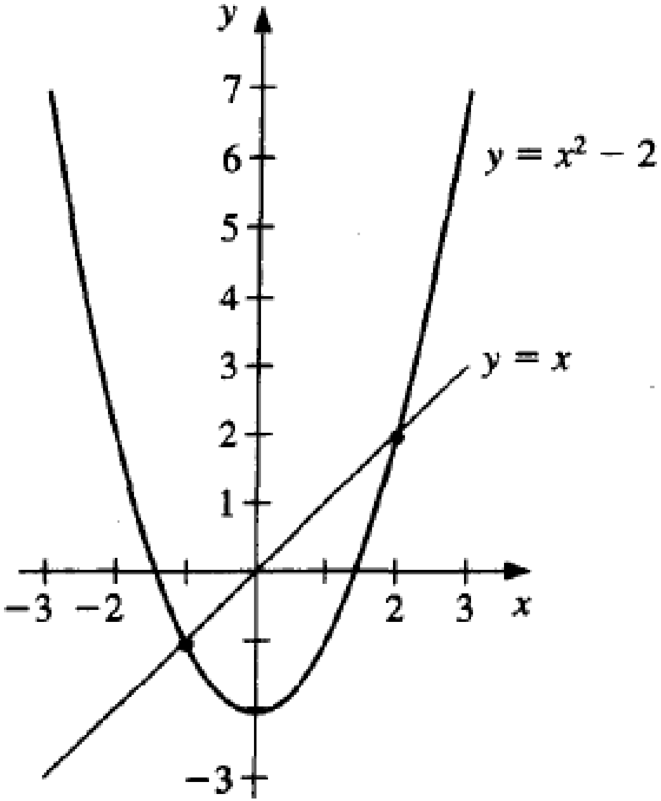


Figure 2.2

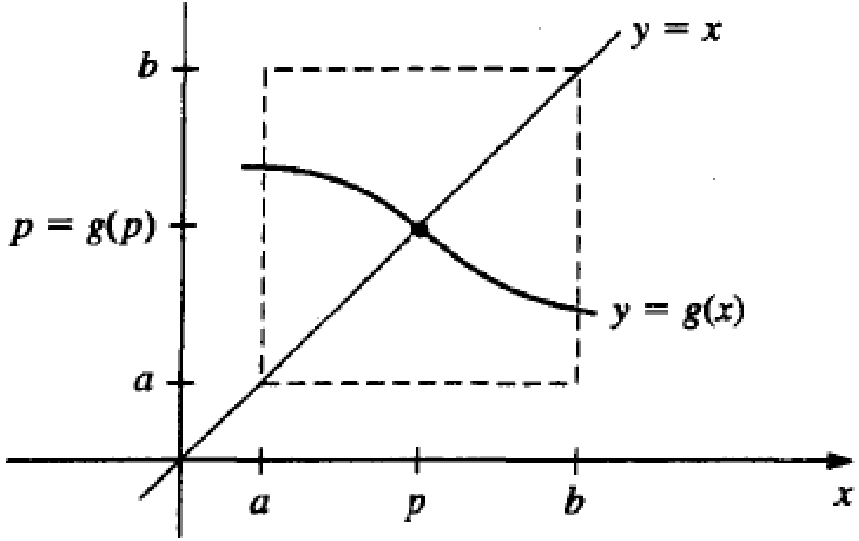
Theorem 2.2

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

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

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

then the fixed point in $[a, b]$ is unique. (See Figure 2.3.) ■



Proof

a. 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 since

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

b. Suppose, in addition, that $|g'(x)| \le k < 1$ 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,

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

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

a. Let $g(x) = (x^2 - 1)/3$ on $[-1, 1]$. The Extreme Value Theorem implies that the absolute minimum of g occurs at $x = 0$ and $g(0) = -\frac{1}{3}$. Similarly, the absolute maximum of g occurs at $x = \pm 1$ and has the value $g(\pm 1) = 0$. Moreover, g is continuous and

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

So g satisfies all the hypotheses of Theorem 2.2 and has a unique fixed point in $[-1, 1]$.

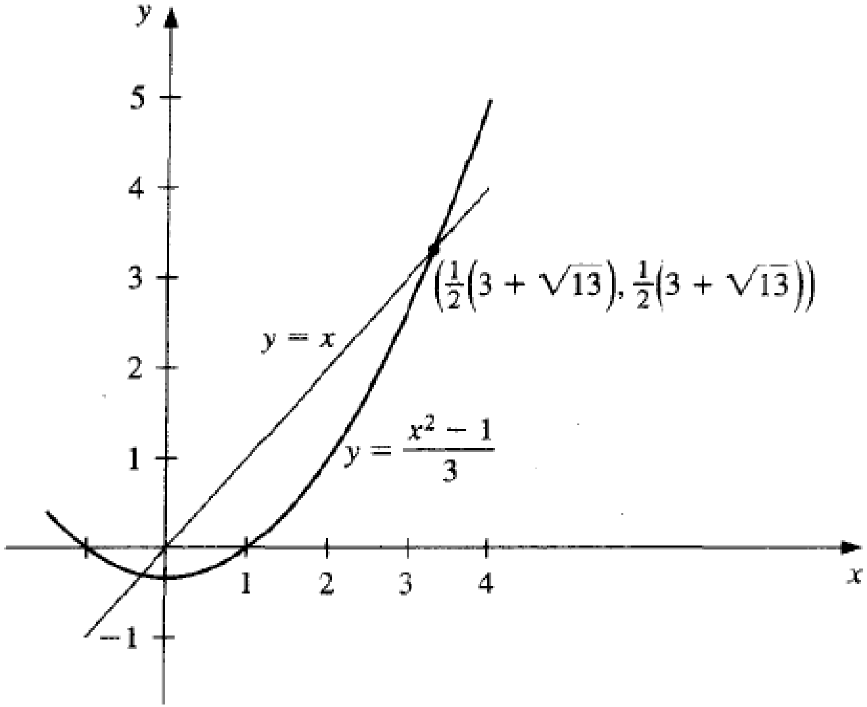
In this example, the unique fixed point p in the interval $[-1, 1]$ can be determined algebraically. If

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

which, by the quadratic formula, implies 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 2.2 on $[3, 4]$. Hence, the hypotheses of Theorem 2.2 are sufficient to guarantee a unique fixed point but are not necessary. (See Figure 2.4.)

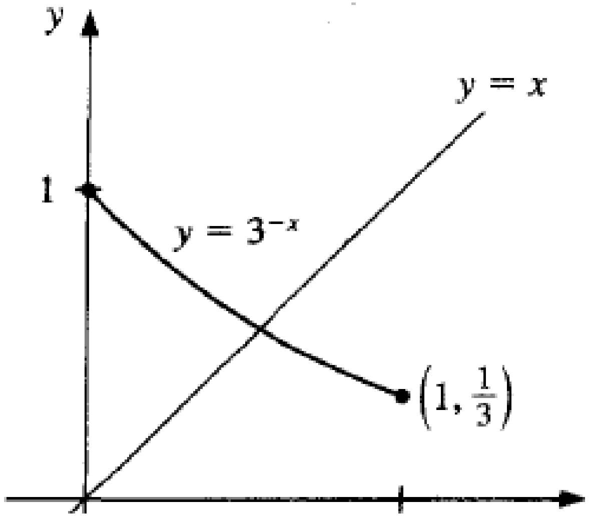


b. Let $g(x) = 3^{-x}$. Since $g'(x) = -3^{-x} \ln 3 < 0$ on $[0, 1]$, the function g is decreasing on $[0, 1]$. So

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

Thus, for $x \in [0, 1]$, we have $g(x) \in [0, 1]$, and g has a fixed point in $[0, 1]$. Since

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

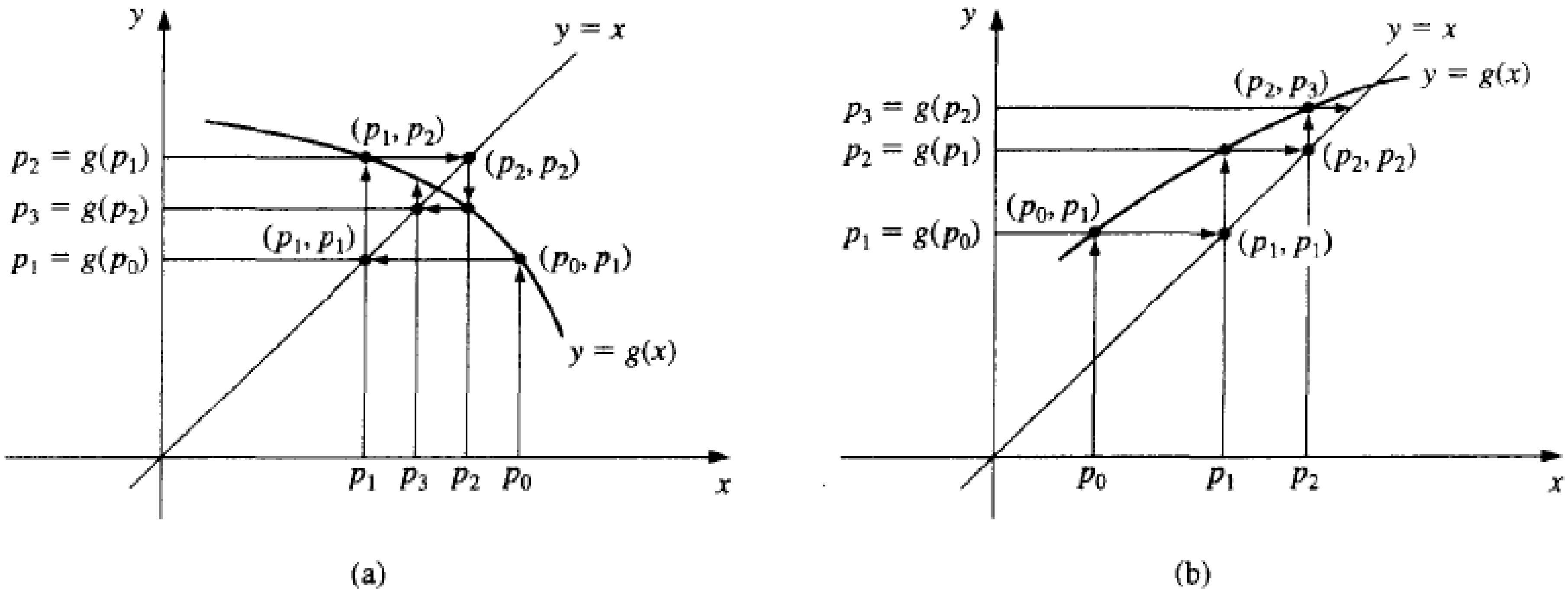


$|g'(x)| \not\le 1$ on $(0, 1)$, and Theorem 2.2 cannot be used to determine uniqueness. However, g is always decreasing, and it is clear from Figure 2.5 that the fixed point must be unique. ■

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 iteration**, or **functional iteration**. The procedure is detailed in Algorithm 2.2 and illustrated in Figure 2.6.



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.
```

EXAMPLE

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:

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

and

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

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

$$x^3 + 4x^2 - 10 = 0$$

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

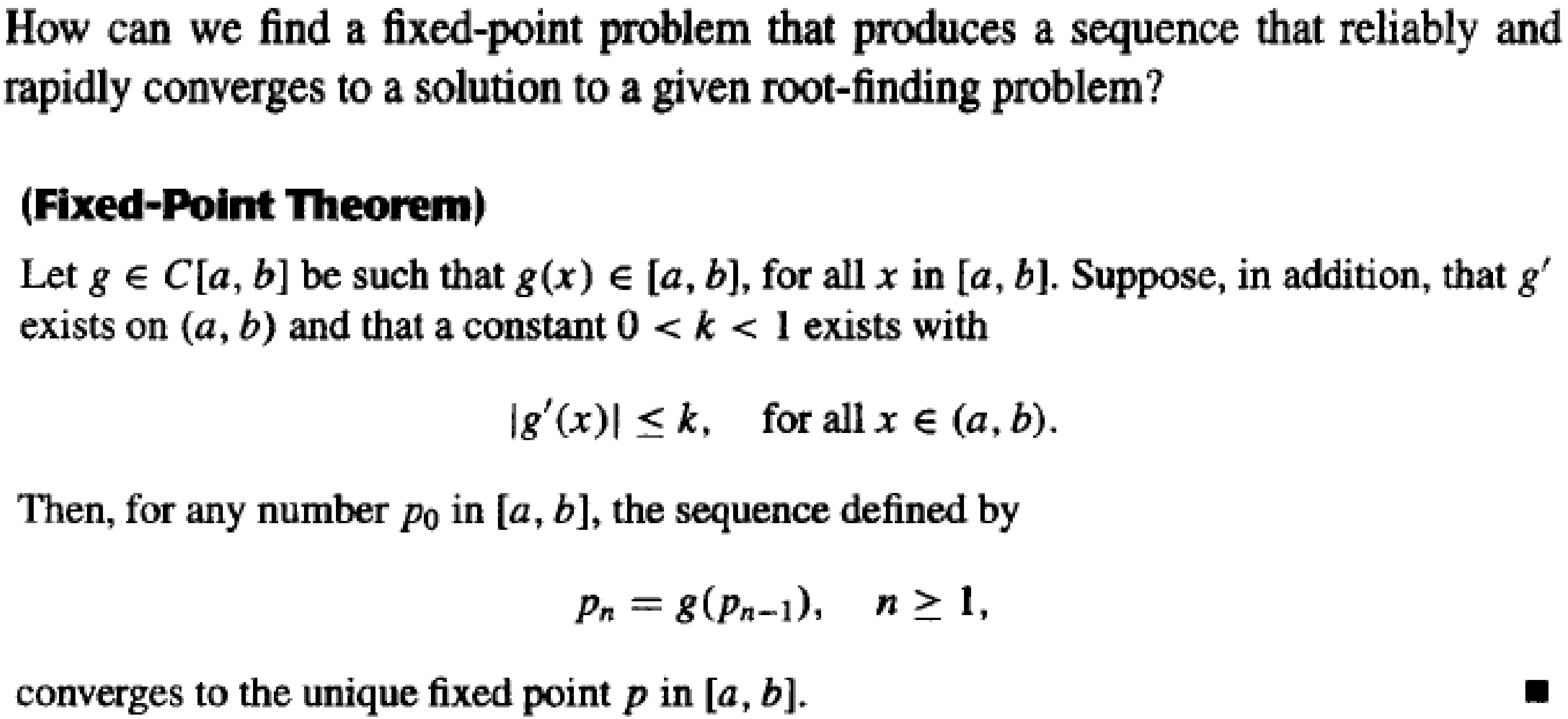
e. $x = g_5(x) = x - \frac{x^3 + 4x^2 - 10}{3x^2 + 8x}$

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

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), since 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. ■

Even though the various functions in Example 3 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 the true question that needs to be answered:

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		



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?

(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

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

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

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

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

If g satisfies the hypotheses of Theorem 2.3, then bounds for the error involved in using p_n to approximate p are given by

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

and

$$|p_n - p| \le \frac{k^n}{1-k}|p_1 - p_0|, \quad \text{for all} \quad n \ge 1. \qquad \blacksquare$$

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. In the following example, the fixed-point methods in Example 3 are reconsidered in light of the results presented in Theorem 2.3 and its corollary.

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 2.3 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

$$|g_2'(x)| < 1, \quad \text{since} \quad |g_2'(p)| \approx 3.4.$$

There is no reason to expect that this method will converge.

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

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

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. Since it is also true that $|g_3'(x)| \leq |g_3'(1.5)| \approx 0.66$ on this interval, Theorem 2.3 confirms the convergence of which we were already aware.

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

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

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. ■

2.3 Newton's Method

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. 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.

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

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

where $\xi(x)$ lies between x and $\overline{x}$. Since $f(p) = 0$, this equation with $x = p$ gives

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

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

$$0 \approx f(\overline{x}) + (p - \overline{x})f'(\overline{x}).$$

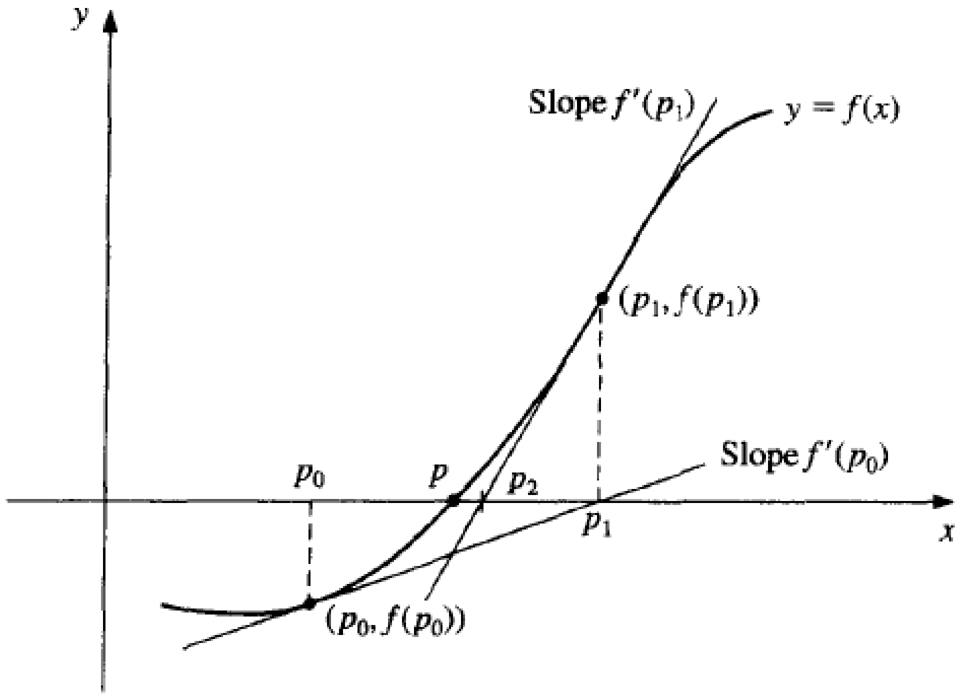
Solving for p gives

$$p \approx \overline{x} - \frac{f(\overline{x})}{f'(\overline{x})}.$$

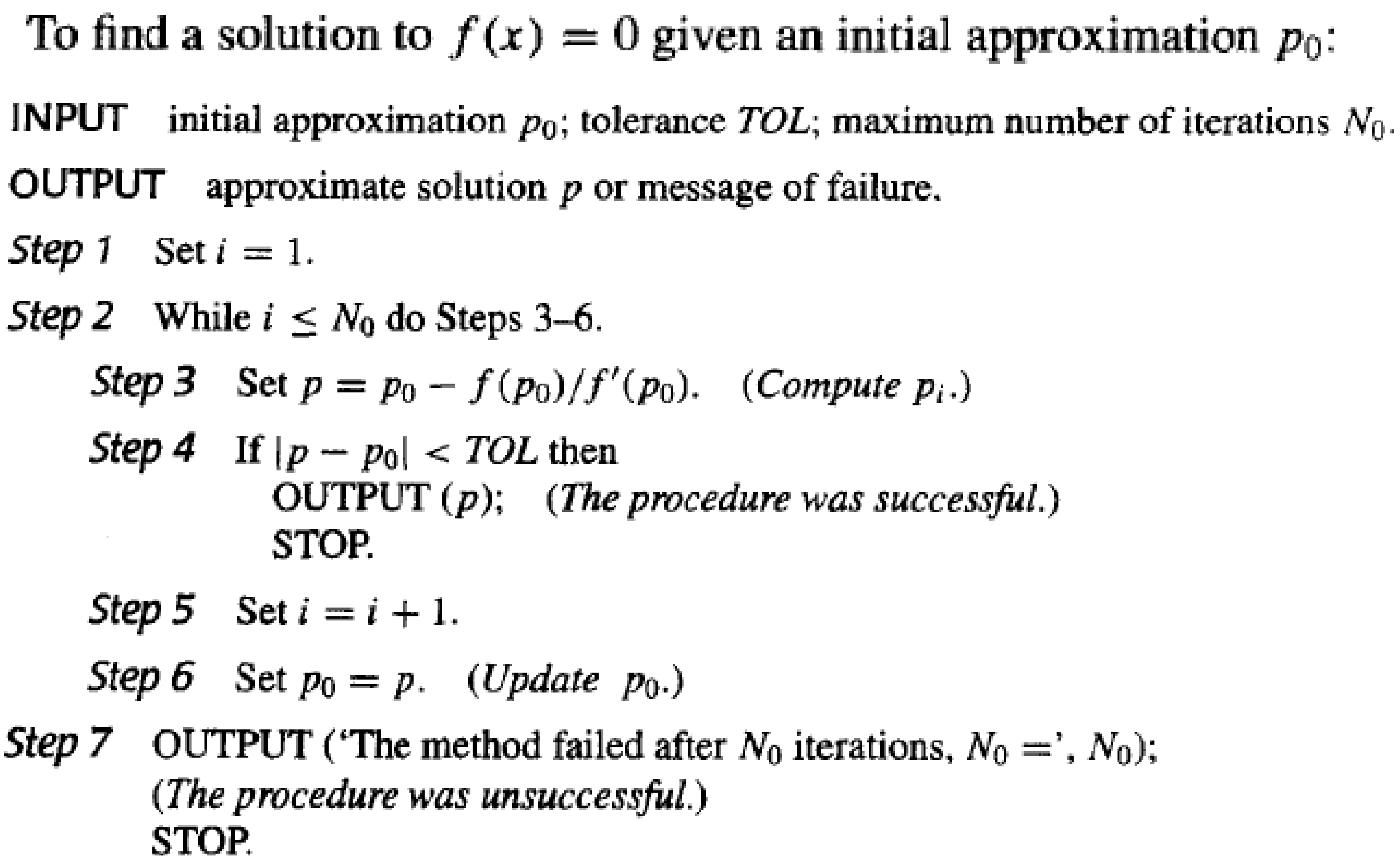
This sets the stage for Newton's method, which starts with an initial approximation p_0 and generates the sequence $\{p_n\}_{n=0}^{\infty}$, by

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{for } n \geq 1. \tag{2.5}$$

Figure 2.7 illustrates how the approximations are obtained using successive tangents. (Also see Exercise 11.) 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.



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 $i \leq N_0$ 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

$$|p_N - p_{N-1}| < \varepsilon, \tag{2.6}$$

$$\frac{|p_N - p_{N-1}|}{|p_N|} < \varepsilon, \quad p_N \neq 0, \tag{2.7}$$

or

$$|f(p_N)| < \varepsilon. \tag{2.8}$$

A form of Inequality (2.6) is used in Step 4 of Algorithm 2.3. Note that inequality (2.8) may not give much information about the actual error $|p_N - p|$. (See Exercise 14 in Section 2.1.)

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

$$g(p_{n-1}) = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{for } n \geq 1. \tag{2.9}$$

In fact, this is the functional iteration technique that was used to give the rapid convergence we saw in part (e) of Example 3 in Section 2.2.

It is clear from equation (2.9) 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.

Suppose we would like to approximate a fixed point of $g(x) = \cos x$. The graph in Figure 2.8 implies that a single fixed-point p lies in $[0, \pi/2]$.

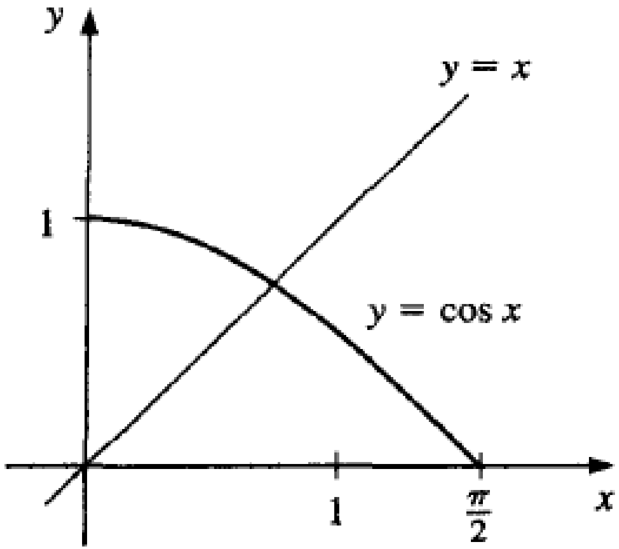


Table 2.3 shows the results of fixed-point iteration with $p_0 = \pi/4$. The best we could conclude from these results is that $p \approx 0.74$.

To approach this problem differently, define $f(x) = \cos x - x$ and apply Newton's method. Since $f'(x) = -\sin x - 1$, the sequence is generated by

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

With $p_0 = \pi/4$, the approximations in Table 2.4 are generated. An excellent approximation is obtained with $n = 3$. We would expect this result to be accurate to the places listed because of the agreement of p_3 and p_4. ■

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

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

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 $(p - \overline{x})^2$ is, by comparison with $|p - \overline{x}|$, so small that it can be deleted. This will clearly be false unless $\overline{x}$ 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 12 and 16 illustrate some of these possibilities.)

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

Theorem 2.5

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 $p_0 \in [p - \delta, p + \delta]$. ■

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,

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

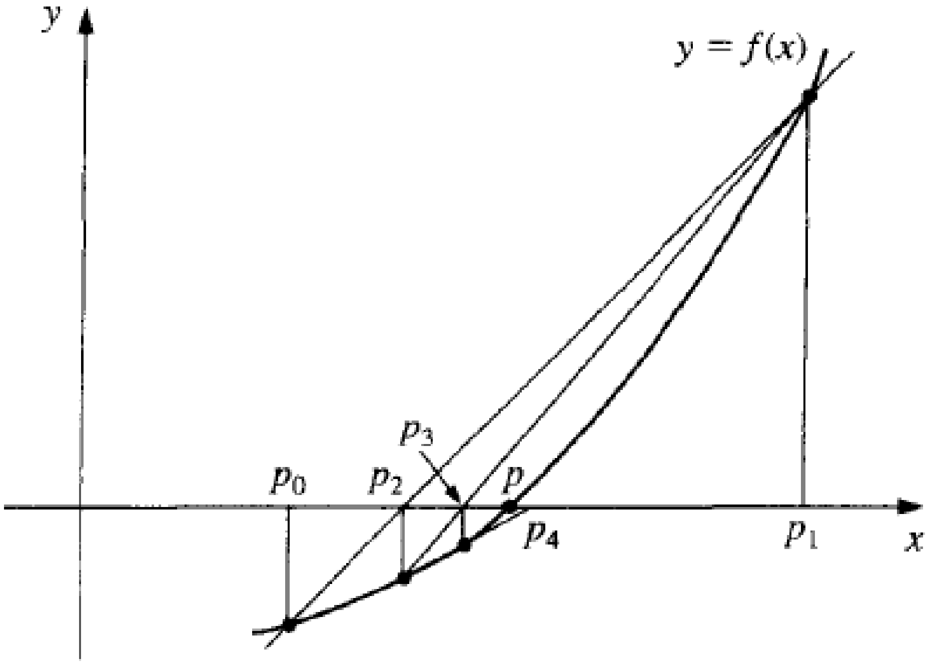
Letting $x = p_{n-2}$, we have

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

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

$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}. \tag{2.10}$$

This technique is called the **Secant method** and is presented in Algorithm 2.4. (See Figure 2.9.) 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.



Use the Secant method to find a solution to $x = \cos x$. In Example 1 we compared functional iteration and Newton's method with the initial approximation $p_0 = \pi/4$. Here we need two initial approximations. Table 2.5 lists the calculations with $p_0 = 0.5$, $p_1 = \pi/4$, and the formula

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

from Algorithm 2.4. ■

Table 2.5

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

By comparing the results here with those in Example 1, we see that p_5 is accurate to the tenth decimal place. The convergence of the Secant method is much faster than functional iteration but slightly slower than Newton's method, which obtained this degree of accuracy with p_3. This is generally true. (See Exercise 12 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 a good first approximation but generally give rapid convergence.

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

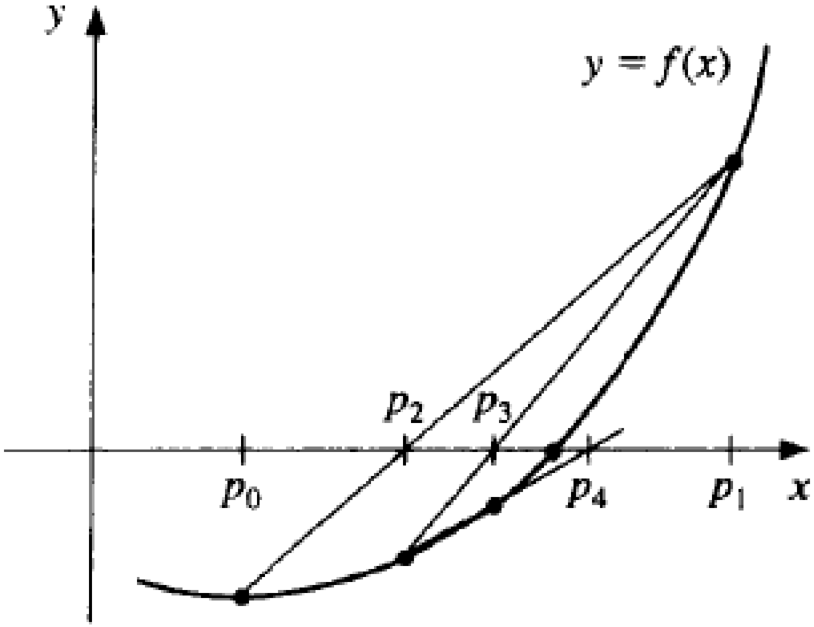
$$|p_n - p| < \frac{1}{2}|a_n - b_n|,$$

which provides an easily calculated error bound for the approximations. Root bracketing is not guaranteed for either Newton's method or the Secant method. Table 2.4 contains results from Newton's method applied to $f(x) = \cos x - x$, where an approximate root was found to be 0.7390851332. Notice that this root is not bracketed by either p_0, p_1 or p_1, p_2. The Secant method approximations for this problem are given in Table 2.5. 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.

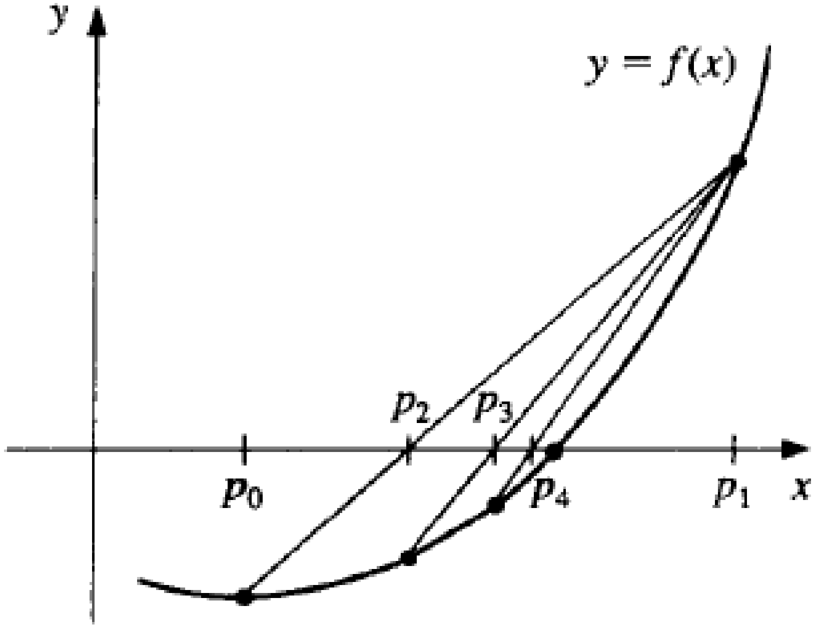
The **method of False Position** (also called *Regula Falsi*) generates approximations in the same manner as the Secant method, but it includes a test to ensure that the root is bracketed between successive iterations. Although it is not a method we generally recommend, it illustrates how bracketing 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, we check $f(p_2) \cdot f(p_1)$. If this value is negative, then p_1 and p_2 bracket a root, and we choose p_3 as the x-intercept of the line joining $(p_1, f(p_1))$ and $(p_2, f(p_2))$. If not, we 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 2.10 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.

Secant method



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 procedure for measuring how rapidly a sequence converges.

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 is not as important as the order. Two cases of order are given special attention.

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

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

The next example 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.

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, suppose that

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

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 when $|p_0| = |\tilde{p}_0| = 1$.

	Linear Convergence	Quadratic Convergence
	Sequence $\{p_n\}_{n=0}^{\infty}$	Sequence $\{\tilde{p}_n\}_{n=0}^{\infty}$
n	$(0.5)^n$	$(0.5)^{2^n-1}$
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}

Theorem 2.7

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

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

If $g'(p) \neq 0$, then for any number p_0 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 2.3 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 2.7 implies that higher-order convergence for fixed-point methods can occur only when $g'(p) = 0$. The next result describes additional conditions that ensure the quadratic convergence we seek.

Theorem 2.8

Let p be a solution of the equation $x = g(x)$. Suppose that $g'(p) = 0$ and g'' is continuous and strictly bounded by 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,

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

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 2.5 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 $g''(p) = 0$.

Since 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 2.7 and 2.8 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.

The easiest way to construct a fixed-point problem associated with a root-finding problem $f(x) = 0$ is to subtract a multiple of $f(x)$ from x. So let us consider

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

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 $f(p) = 0$. Since

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

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.

In the preceding discussion, the restriction was made that $f'(p) \neq 0$, where p is the solution to $f(x) = 0$. From the definition of Newton's method, it is clear that difficulties might occur if $f'(p_n)$ goes to zero simultaneously with $f(p_n)$. 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 2.9

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 $\lim_{x \to p} q(x) \neq 0$. ■

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 2.10

$f \in C^1[a, b]$ 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. ■ ■ ■

Theorem 2.11

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

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

The result in Theorem 2.10 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 may not occur if the zero is not simple.

Consider $f(x) = e^x - x - 1$. Since $f(0) = e^0 - 0 - 1 = 0$ and $f'(0) = e^0 - 1 = 0$, but $f''(0) = e^0 = 1$, f has a zero of multiplicity two at $p = 0$. In fact, $f(x)$ can be expressed in the form

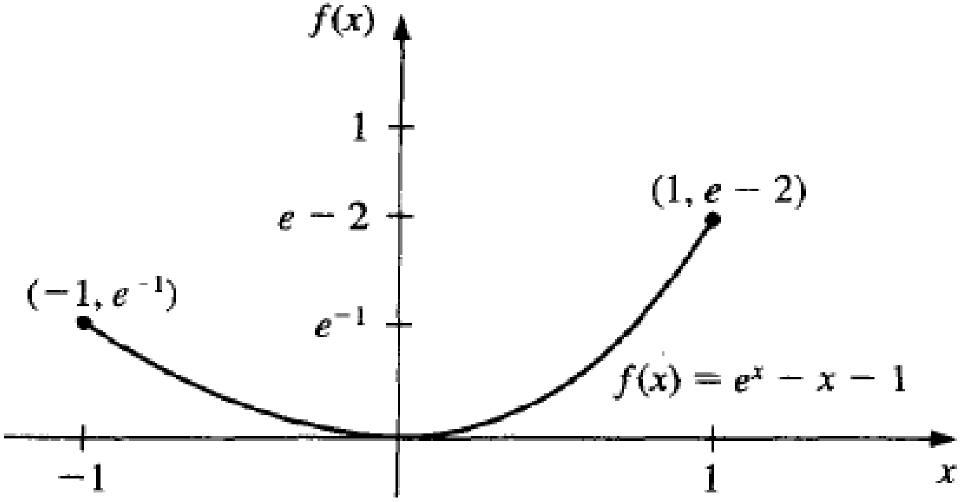
$$f(x) = (x - 0)^2 \frac{e^x - x - 1}{x^2},$$

where, by L’Hôpital’s rule,

$$\lim_{x\to 0} \frac{e^x - x - 1}{x^2} = \lim_{x\to 0} \frac{e^x - 1}{2x} = \lim_{x\to 0} \frac{e^x}{2} = \frac{1}{2} \neq 0.$$

The terms generated by Newton’s method applied to f with $p_0 = 1$ are shown in Table 2.8. The sequence is clearly converging to 0, but not quadratically. The graph of f is shown in Figure 2.11. ■

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


$f(x)$
1
$e-2$
e^{-1}
$(-1, e^{-1})$
$(1, e-2)$
$f(x) = e^x - x - 1$
-1
1
x

One method of handling the problem of multiple roots is to define

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

If p is a zero of f of multiplicity m and $f(x) = (x-p)^m q(x)$, 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 μ. Newton's method can then be applied to μ to give

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

or

$$g(x) = x - \frac{f(x)f'(x)}{[f'(x)]^2 - f(x)f''(x)}. \tag{2.11}$$

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 roundoff problems since the denominator of (2.11) consists of the difference of two numbers that are both close to 0.

Table 2.9 lists the approximations to the double zero at $x = 0$ of $f(x) = e^x - x - 1$ using $p_n = g(p_{n-1})$, for $n \geq 1$, where g is given by (2.11). The results were recorded using a calculator with ten digits of precision. The initial approximation of $p_0 = 1$ was chosen so that the entries can be compared with those in Table 2.8. What Table 2.9 does not show is that no improvement to the zero approximation $-2.8085217 \times 10^{-7}$ occurs in subsequent computations using this calculator since both the numerator and the denominator 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	$-2.8085217 \times 10^{-7}$

Table 2.10

	(i)	(ii)
p_1	1.37333333	1.35689898
p_2	1.36526201	1.36519585
p_3	1.36523001	1.36523001

In Example 3 of Section 2.2 we solved $f(x) = x^3 + 4x^2 - 10 = 0$ for the zero $p = 1.36523001$. To compare convergence for a zero of multiplicity one by Newton's method and the modified Newton's method listed in Eq. (2.11), let

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

and, from $p_n = g(p_{n-1})$, where g is given by Eq. (2.11),

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

With $p_0 = 1.5$, the first three iterates for (i) and (ii) are shown in Table 2.10. The results illustrate the rapid convergence of both methods in the case of a simple zero. ■

2.5 Accelerating Convergence

It is rare to have the luxury of quadratic convergence. We now consider a technique called **Aitken's Δ^2 method** that can be used to accelerate the convergence of a sequence that is linearly convergent, regardless of its origin or application.

Suppose $\{p_n\}_{n=0}^{\infty}$ is a linearly convergent sequence with limit p. To motivate the construction of a sequence $\{\hat{p}_n\}_{n=0}^{\infty}$ that converges more rapidly to p than does $\{p_n\}_{n=0}^{\infty}$, let us first assume that the signs of $p_n - p$, $p_{n+1} - p$, and $p_{n+2} - p$ agree and that n is sufficiently large that

$$\frac{p_{n+1} - p}{p_n - p} \approx \frac{p_{n+2} - p}{p_{n+1} - p}.$$

Then

$$(p_{n+1} - p)^2 \approx (p_{n+2} - p)(p_n - p),$$

$$p_{n+1}^2 - 2p_{n+1}p + p^2 \approx p_{n+2}p_n - (p_n + p_{n+2})p + p^2$$

and

$$(p_{n+2} + p_n - 2p_{n+1})p \approx p_{n+2}p_n - p_{n+1}^2.$$

Solving for p gives

$$p \approx \frac{p_{n+2}p_n - p_{n+1}^2}{p_{n+2} - 2p_{n+1} + p_n}.$$

Adding and subtracting the terms p_n^2 and $2p_n p_{n+1}$ in the numerator and grouping terms appropriately gives

gives

$\boxed{+}$

$$p \approx \frac{p_n^2 - p_n p_{n+2} - 2p_n p_{n+1} - 2p_n p_{n+1} - p_n^2 - p_{n+1}^2}{p_{n+2} - 2p_{n+1} + p_n}$$

$$= \frac{(p_n^2 - p_n p_{n+2} + 2p_n p_{n+1}) - (p_n^2 - 2p_n p_{n+1} + p_{n+1}^2)}{p_{n+2} - 2p_{n+1} + p_n}$$

$\boxed{-}$

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

Aitken's Δ^2 method is based on the assumption that the sequence $\{\hat{p}_n\}_{n=0}^{\infty}$, defined by

$$\hat{p}_n = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}, \tag{2.12}$$

converges more rapidly to p than does the original sequence $\{p_n\}_{n=0}^{\infty}$.

EXAMPLE 1

The sequence $\{p_n\}_{n=1}^{\infty}$, where $p_n = \cos(1/n)$, converges linearly to $p = 1$. The first few terms of the sequences $\{p_n\}_{n=1}^{\infty}$ and $\{\hat{p}_n\}_{n=1}^{\infty}$ are given in Table 2.11. It certainly appears that $\{\hat{p}_n\}_{n=1}^{\infty}$ converges more rapidly to $p = 1$ than does $\{p_n\}_{n=1}^{\infty}$. ■

Table 2.11

n	p_n	$\hat{p}_n$
1	0.54030	0.96178
2	0.87758	0.98213
3	0.94496	0.98979
4	0.96891	0.99342
5	0.98007	0.99541
6	0.98614	
7	0.98981	

The Δ notation associated with this technique has its origin in the following definition.

Definition 2.12 For a given sequence $\{p_n\}_{n=0}^{\infty}$, the **forward difference** Δp_n is defined by

$$\Delta p_n = p_{n+1} - p_n, \quad \text{for } n \geq 0.$$

Higher powers, $\Delta^k p_n$, are defined recursively by

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

The definition implies that

$$\Delta^2 p_n = \Delta(p_{n+1} - p_n) = \Delta p_{n+1} - \Delta p_n = (p_{n+2} - p_{n+1}) - (p_{n+1} - p_n).$$

So

$$\Delta^2 p_n = p_{n+2} - 2p_{n+1} + p_n,$$

and the formula for $\hat{p}_n$ given in Eq. (2.12) can be written as

$$\hat{p} = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n}, \quad \text{for } n \geq 0. \tag{2.13}$$

To this point in our discussion of Aitken's Δ^2 method, we have stated that the sequence $\{\hat{p}_n\}_{n=0}^{\infty}$, converges to p more rapidly than does the original sequence $\{p_n\}_{n=0}^{\infty}$, but we have not said what is meant by the term "more rapid" convergence. Theorem 2.13 explains and justifies this terminology. The proof of this theorem is considered in Exercise 14.

Theorem 2.13

Suppose that $\{p_n\}_{n=0}^{\infty}$ is a sequence that converges linearly to the limit p and that for all sufficiently large values of n we have $(p_n - p)(p_{n+1} - p) > 0$. Then the sequence $\{\hat{p}_n\}_{n=0}^{\infty}$ converges to p faster than $\{p_n\}_{n=0}^{\infty}$ in the sense that

$$\lim_{n\to\infty} \frac{\hat{p}_n - p}{p_n - p} = 0. \qquad \blacksquare$$

By applying a modification of Aitken's Δ^2 method to a linearly convergent sequence obtained from fixed-point iteration, we can accelerate the convergence to quadratic. This procedure is known as Steffensen's method and differs slightly from applying Aitken's Δ^2 method directly to the linearly convergent fixed-point iteration sequence. Aitken's Δ^2 method constructs the terms in order:

$$\begin{gathered} p_0, \quad p_1 = g(p_0), \quad p_2 = g(p_1), \quad \hat{p}_0 = \{\Delta^2\}(p_0), \\ p_3 = g(p_2), \quad \hat{p}_1 = \{\Delta^2\}(p_1), \ldots, \end{gathered}$$

where $\{\Delta^2\}$ indicates that Eq. (2.13) is used. Steffensen's method constructs the same first four terms, p_0, p_1, p_2, and $\hat{p}_0$. However, at this step it assumes that $\hat{p}_0$ is a better approximation to p than is p_2 and applies fixed-point iteration to $\hat{p}_0$ instead of p_2. Using this notation the sequence generated is

$$p_0^{(0)}, \quad p_1^{(0)} = g(p_0^{(0)}), \quad p_2^{(0)} = g(p_1^{(0)}), \quad p_0^{(1)} = \{\Delta^2\}(p_0^{(0)}), \quad p_1^{(1)} = g(p_0^{(1)}), \ldots .$$

Every third term is generated by Eq. (2.13); the others use fixed-point iteration on the previous term. The process is described in Algorithm 2.6.

Steffensen's

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 $p_1 = g(p_0)$;   (Compute $p_1^{(i-1)}$.)
                $p_2 = g(p_1)$;   (Compute $p_2^{(i-1)}$.)
                $p = p_0 - (p_1 - p_0)^2/(p_2 - 2p_1 + p_0)$.   (Compute $p_0^{(i)}$.)

    Step 4  If $|p - p_0| < TOL$ then
              OUTPUT ($p$);   (Procedure completed successfully.)
              STOP.

    Step 5  Set $i = i + 1$.

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

Step 7  OUTPUT ('Method failed after $N_0$ iterations, $N_0$ =', $N_0$);
        (Procedure completed unsuccessfully.)
        STOP.                                                        ■
```