Slides on Numerical Analysis

Chapter 2

Solutions of Equations in One Variable

In this chapter, we consider one of the basic problems of numerical approximation:

the root-finding problem.

It involves finding a root x of an equation of the form $f(x) = 0$, for a given function f.

Or it can be described as follows:

Finding x, such that $f(x) = 0$.

2.1 Bisection Method

The first technique, based on the **Intermediate Value Theorem**, is called the **Bisection, or Binary-search, method (对分法或折半查找法).**

Suppose $f(x)$ is a continuous function defined on the interval $[a, b]$, and with $f(a)$ and $f(b)$ of opposite sign (That is, $f(a)f(b) < 0$). By the Intermediate Value Theorem, there exists one point $c \in (a, b)$, such that $f(c) = 0$.

NOTES: Although the procedure will work for the case when $f(a)$ and $f(b)$ have opposite signs and there is more than one root in the interval (a, b), we assume for simplicity that the root in this interval is unique.

BASIC IDEA:

1 Let $a_1 = a, b_1 = b$, and let $c_1 = (a_1 + b_1)/2$ be the midpoint of interval $[a, b]$. It is clear that if $f(c_1) = 0$, then $c = c_1$. If not, the $f(c_1)$ has the same sign as either $f(a_1)$ or $f(b_1)$. If $f(c_1)$ has the same sign as $f(a_1)$, then $c \in (c_1, b_1)$, and we set $a_2 = c_1, b_2 = b_1$; otherwise if $f(c_1)$ has the same sign as $f(b_1)$, then $c \in (a_1, c_1)$, and we set $a_2 = a_1, b_2 = c_1$. Thus we get the new subinterval $[a_2, b_2]$.

2 Re-apply the process to the new interval $[a_2, b_2]$, to get $[a_3, b_3]$.

3 Continue this procedure until the stopping criteria are met.

4 Suppose we have got the subinterval $[a_n, b_n]$, let $c_n = (a_n + b_n)/2 = a_n + (b_n - a_n)/2$;

If $|b_n - a_n| < \varepsilon$, where ε is a given sufficient small positive number, then stop this procedure and output the approximated solution as c_n;

else compute $f(c_n)$;

if $f(c_n) = 0$(or $|f(c_n)| < \epsilon$), then output the root $c = c_n$ and stop;

else if $f(a_n)f(c_n) > 0$, then let $a_{n+1} = c_n, b_{n+1} = b_n$;

otherwise let $a_{n+1} = a_n, b_{n+1} = c_n$;

continue this procedure.

ALGORITHM 2.1:

1 Input $a, b, \varepsilon_1, \varepsilon_2, FA = f(a), FB = f(b), N$**;**

2 Let $n = 1$**;**

3 If $n > N$**, goto 8**

4 Let $c = a + (b - a)/2$ **or** $c = (a + b)/2$**, compute** $FC = f(c)$**.**

5 If $FC = 0$ **or** $|FC| < \varepsilon_1$ **or** $|b-a| < \varepsilon_2$**, then output** c**, stop.**

6 If $FA \cdot FC < 0$**, then** $b = c, FB = FC$**, else** $a = c, FA = FC$**.**

7 Let $n = n + 1$**, goto 3.**

8 Output "Methods failed after N **iterations, N=",N, stop.**

Note that $\varepsilon_1, \varepsilon_2$ is the maximum tolerance.

Other stopping procedures(besides $|f(c_n)| < \epsilon$):

$$|c_n - c_{n-1}| < \epsilon,$$

$$\frac{|c_n - c_{n-1}|}{|c_n|} < \epsilon.$$

It is good practice to set an upper bound on the number of iterations.

It is advantageous to choose the initial interval $[a, b]$ as small as possible.

Example 1: Find the root in [1,2] for equation $f(x) = x^3 + 4x^2 - 10 = 0$.

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

The Bisection method is slow to converge. However, it always converges to a solution, so it is often used as a starter for the more efficient methods.

Convergence Analysis

THEOREM 2.1 Suppose that $f \in C[a,b]$, and $f(a)f(b) < 0$. The Bisection method as given in Algorithm 2.1 generates a sequence $\{c_n\}_1^\infty$ approximating a zero c of f with

$$|c_n - c| \leq \frac{b-a}{2^n}, n \geq 1$$

Proof: By the procedure, we know that

$$\begin{aligned}
|b_1 - a_1| &= |b - a|, c \in [a_1, b_1] \\
|b_2 - a_2| &= |b_1 - a_1|/2 = |b - a|/2, c \in [a_2, b_2] \\
\cdots & \quad \cdots \quad \cdots \\
|b_n - a_n| &= |b_{n-1} - a_{n-1}|/2 = |b - a|/2^{n-1}, \\
& \quad c \in [a_n, b_n].
\end{aligned}$$

Since $c_n = (a_n + b_n)/2$ and $c \in (a_n, c_n]$ or $c \in [c_n, b_n)$ for all $n \geq 1$, it follows that

$$|c_n - c| \leq \frac{|b_n - a_n|}{2} = \frac{|b - a|}{2^n}.$$

That is, c_n converges to c with rate of convergence $O(\frac{1}{2^n})$. That is,

$$c_n = c + O(\frac{1}{2^n}).$$

2.2 Fixed-Point Iteration

A **fixed point** for a given function g is a number p for which $g(p) = p$.

In this section, we consider the problem of finding solutions to fixed-point problems and the connection between these 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 a function g with a fixed point at p in a number of ways, for example, as $g(x) = x - f(x)$ or as $g(x) = x + 3f(x)$, $g(x) = x - \frac{f(x)}{f'(x)}$.

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.

Remarks: 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.

Our first task is 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 following theorem gives sufficient conditions (but not necessary) for the existence and uniqueness of a fixed point:

THEOREM 2.2:

a. If $g(x) \in C[a,b]$ and $g(x) \in [a,b]$ for all $x \in [a,b]$, then $g(x)$ 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 $|g'(x)| \leq k$, for all $x \in (a,b)$. Then the fixed point in $[a,b]$ is unique.

Proof:

a. If $g(a) = a$ or $g(b) = b$, then $g(x)$ has a fixed point at an endpoint. If not, then it must be true that $g(a) > a$ and $g(b) < b$. Thus the function $h(x) = g(x) - x$ is continuous on $[a, b]$, and we have

$$h(a) = g(a) - a > 0, h(b) = g(b) - b < 0.$$

The Intermediate Value Theorem implies that there exists $p \in (a, b)$ for which $h(p) = 0$. Thus $g(p) - p = 0$, and p is a fixed point of $g(x)$.

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

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

Then

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

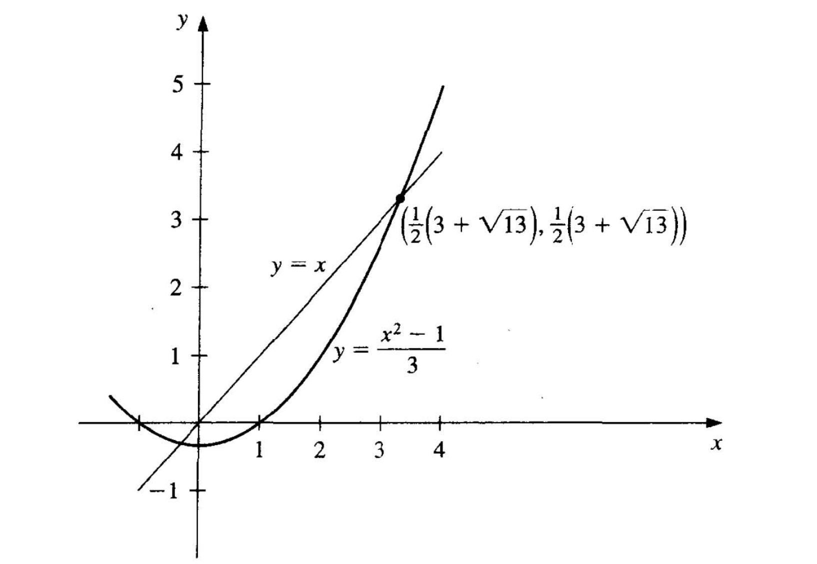
which is a contradiction. So $p = q$, and the fixed point in $[a, b]$ is unique. ■

Example 2: Let $g(x) = (x^2 - 1)/3$ on [-1,1]. Then $g(x) \in [-1/3, 0] \subset [-1, 1]$. Moreover, g is continuous and $|g'(x)| \leq 2/3$ for all $x \in (-1, 1)$, thus according to Theorem 2.2, $g(x)$ has a unique fixed point in [-1,1]: $p = \frac{1}{2}(3 - \sqrt{13})$.

Note that g does not satisfy the hypotheses of Theorem 2.2 on [3,4] since $g(4) = 5$ and $g'(4) = \frac{8}{3} > 1$, however, g also has a unique fixed point $p = \frac{1}{2}(3 + \sqrt{13})$ in [3,4], see Figure 2.4.

Hence, the hypotheses of Theorem 2.2 are sufficient to guarantee a unique fixed point but are not necessary.

Figure 2.4



y
5
4
3
2
1
−1
1
2
3
4
x
$y = x$
$y = \frac{x^2 - 1}{3}$
$\left(\frac{1}{2}\left(3 + \sqrt{13}\right), \frac{1}{2}\left(3 + \sqrt{13}\right)\right)$

♦ How to Find the Fixed-Point of a Function

To approximate the fixed point of a function $g(x)$, 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 $n \geq 1$.

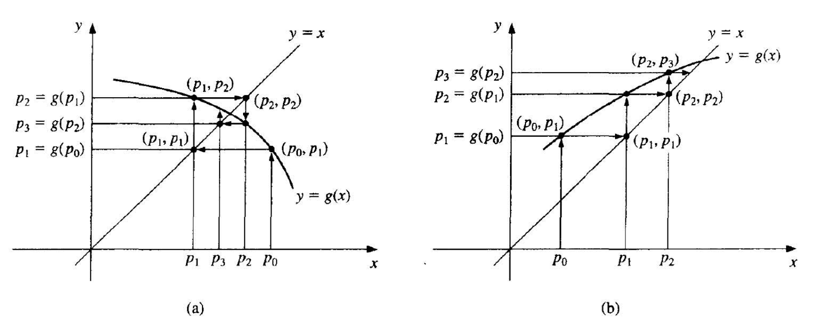
If the sequence $\{p_n\}_{n=0}^{\infty}$ converges to p and $g(x)$ is continuous, then we have

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

and a solution to $x = g(x)$ is obtained, see Figure 2.6.

This technique is called **fixed point iteration (or functional iteration)**.

Figure 2.6



y
y = x
(p1, p2)
p2 = g(p1)
(p2, p2)
p3 = g(p2)
(p1, p1)
p1 = g(p0)
(p0, p1)
y = g(x)
p1
p3
p2
p0
x
(a)
y
y = x
(p2, p3)
y = g(x)
p3 = g(p2)
p2 = g(p1)
(p2, p2)
(p0, p1)
p1 = g(p0)
(p1, p1)
p0
p1
p2
x
(b)

ALGORITHM 2.2 Fixed-Point Iteration Method

Step 1 Input initial approximation p_0; tolerance ε; maximum number of iterations N.

Step 2 Set $n = 1$.

Step 3 If $n > N$, goto Step 7.

Step 4 Set $p = g(p_0)$.

Step 5 If $|p - p_0| < \varepsilon$ then Output p; (Procedure completed successfully), STOP.

Step 6 Set $n = n + 1, p_0 = p$.

Step 7 Output('Method failed after N iterations, $N =$',N); (Procedure completed unsuccessfully), STOP.

Example 3: The equation $x^3 + 4x^2 - 10 = 0$ has a unique root 1.365230013 in [1,2]. There are many ways to change the equation to the fixed-point form $x = g(x)$:

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

Table 2.2 shows that excellent results have been obtained for (c), (d), (e), since the Bisection method requires 27 iterations for this accuracy. (a) was divergent, (b) became undefined.

Notice that (d) converges faster than (c), and (e) is the most efficient choice.

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		

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?

♦ Convergence Analysis for Fixed-Point Iteration

THEOREM 2.3 (Fixed-Point Theorem) Let $g \in C[a, b]$ and $g(x) \in [a, b]$ for all x in $[a, b]$. Suppose, in addition, that $g'(x)$ exists on (a, b) and a positive constant $k < 1$ exists with $|g'(x)| \leq k$, for all $x \in (a, b)$. Then for any number $p_0 \in [a, b]$, the sequence $\{p_n\}_0^\infty$ defined by $p_n = g(p_{n-1}), n \geq 1$, converges to the unique fixed point p in $[a, b]$.

Proof: Since the function $g(x)$ satisfies the all basic conditions that a unique fixed point existed, so by the theorem 2.2, we know that a unique fixed point p exists in $[a, b]$.

Since $g(x)$ maps $[a, b]$ into itself, the sequence $\{p_n\}_0^\infty$ 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, we have

$$\begin{aligned} |p_n - p| &= |g(p_{n-1}) - g(p)| = |g'(\xi)||p_{n-1} - p| \\ &\leq k|p_{n-1} - p|, \end{aligned}$$

where $\xi \in (a, b)$.

Applying this inequality inductively gives

$$\begin{aligned} |p_n - p| &\leq k|p_{n-1} - p| \leq k^2|p_{n-2} - p| \leq \cdots \\ &\leq k^n|p_0 - p|. \end{aligned}$$

Since $k < 1$,

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

and $\{p_n\}_0^\infty$ converges to p.

Corollary 2.4

If $g(x)$ satisfies the hypotheses of Theorem 2.3, 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

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

Proof:

The first bound can be derived as follows:

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

Since $p \in [a, b]$, the next inequality can be given as

$$\begin{aligned} |p_n - p_{n-1}| &\leq |g(p_{n-1}) - g(p_{n-2})| \\ &\leq k|p_{n-1} - p_{n-2}| \\ &\leq \cdots\cdots \\ &\leq k^{n-1}|p_1 - p_0|. \end{aligned}$$

Let $m > n$, then we have

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

Let $m \to \infty$, and since the sequence $\{p_m\}_0^\infty$ converges to the fixed point p, we have

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

Both inequalities in the corollary show that the rate of convergence depends on the factor k^n. The small the value of k, the faster the convergence, which may be very slow if k is close to 1.

Example 4: Reconsider the methods in Example 3.

a. For $g_1(x) = x - x^3 - 4x^2 + 10$, it can be easily checked that g_1 does not map [1,2] into itself and $|g_1'(x)| > 1$ for all x in [1,2]. There is no reason to expect convergence.

b. For $g_2(x) = \left(\frac{10}{x} - 4x\right)^{1/2}$, we see that g_2 does not map [1,2] to [1,2], and $|g_2'(1.365)| \approx 3.4$. There is no reason to expect convergence.

c. For $g_3(x) = \frac{1}{2}(10 - x^3)^{1/2}$, we can show that g_3 maps [1,1.5] into itself. Moreover, $|g_3'(x)| \leq |g_3'(1.5)| \approx 0.66$. Therefore, Theorem 2.3 confirms the convergence.

d. For $g_4(x) = \left(\frac{10}{4+x}\right)^{1/2}$, we have $|g_4'(x)| < 0.15$, much smaller than the bound on the magnitude of $g_3'(x)$, which explains the more rapid convergence using g_4.

e. $g_5(x) = x - \frac{x^3+4x^2-10}{3x^2+8x}$ is the most efficient method (the Newton's Method). In the next sections we will see where this choice came from and why it is so effective.

2.3.1 The Newton-Raphson Method

The Newton-Raphson (or simply Newton's) method is one of the most powerful and well-known numerical methods for solving a root-finding problem $f(x) = 0$.

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

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

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

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

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

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

Solving for p in this equation gives

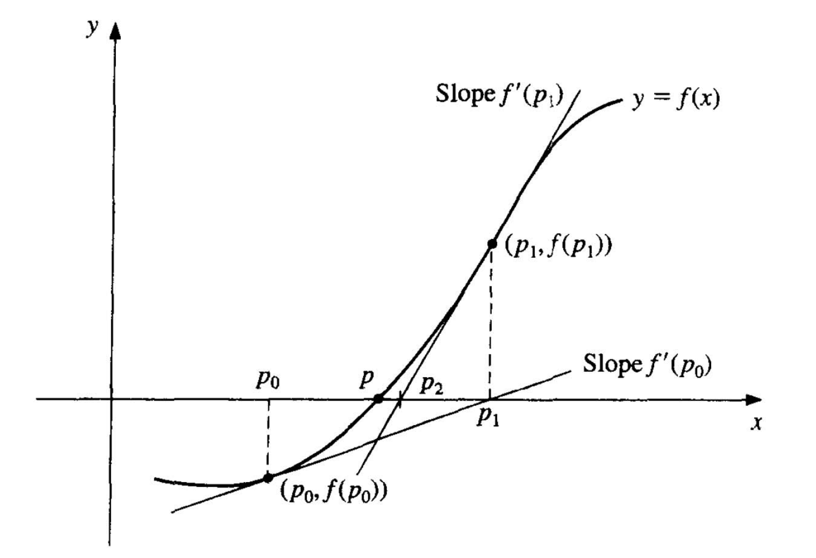
$$p \approx \bar{x} - \frac{f(\bar{x})}{f'(\bar{x})}.$$

This sets the stage for the Newton-Raphson method, which starts with an initial approximation p_0 and generates the sequence $\{p_n\}_0^\infty$ defined by

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

Figure 2.7 illustrates how the approximations are obtained using successive tangents.

Figure 2.7



ALGORITHM 2.3 Newton's Algorithm

To find a solution to $f(x) = 0$ given the differentiable function f and 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 Let $i = 1$.

Step 2 While $i \leq N_0$, do step 3-5.

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); (Procedure completed successfully.) STOP.

Step 5 Set $i = i + 1, p_0 = p$, goto Step 2.

Step 6 OUTPUT ('Method failed after N_0 iterations, $'N_0 =' , N_0$); (Procedure completed unsuccessfully.) STOP.

The stopping-technique inequalities given with Bisection method are applicable to Newton's method:

$$|p_N - p_{N-1}| < \varepsilon, \quad \frac{|p_N - p_{N-1}|}{|p_N|} < \varepsilon, \quad |f(p_N)| < \varepsilon.$$

Newton's method is a fixed-point iteration technique of the form $p_n = g(p_{n-1})$ with

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

which was used to give the rapid convergence we saw in part (e) of Example 3 in Section 2.2.

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

Example 1: Approximate the fixed point of $g(x) = \cos x$ in $[0, \pi/2]$. The real solution $p \approx 0.739085133215$.

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 $p \approx 0.74$.

Define $f(x) = \cos x - x$ and apply Newton's method gives

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

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 points out the importance of an accurate initial approximation. The crucial assumption is that $(p - \bar{x})^2$ is so small that it can be deleted by comparison of $|p - \bar{x}|$. 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.

Sometimes, the Newton's method will even fail for some initial approximation p_0.

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 $\{p_n\}_1^\infty$ 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

$$g(x) = x - f(x)/f'(x).$$

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

Since $f'(p) \neq 0$ and f' is continuous, there exists $\delta_1 > 0$ 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

$$g'(p) = f(p)f''(p)/[f'(p)]^2 = 0.$$

Since g' is continuous and $0 < k < 1$, there exists a δ, with $0 < \delta < \delta_1$ and

$$|g'(x)| \leq k, \qquad \forall x \in [p - \delta, p + \delta].$$

It remains to show that

$$g(x) \;:\; [p-\delta, p+\delta] \mapsto [p-\delta, p+\delta].$$

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

$$\begin{aligned} |g(x) - p| &= |g(x) - g(p)| = |g'(\xi)||x-p| \\ &\leq k|x-p| < |x-p|. \end{aligned}$$

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

This result implies that g maps $[p-\delta, p+\delta]$ into $[p-\delta, p+\delta]$.

All the hypotheses of the Fixed-Point Theorem are now satisfied for $g(x) = x - f(x)/f'(x)$, so the sequence $\{p_n\}_{n=1}^{\infty}$ defined by

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

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

Theorem 2.5 is important for the theory of Newton's method, but it is seldom applied in practice since 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.

2.3.2 Secant method (割线法)

The Newton's method is an extremely powerful technique to solve root finding problems, but it has a major weakness: the need to know the value of the derivative of $f(x)$ at each approximation. Frequently, $f'(x)$ is far more difficult to determine and needs more arithmetic operations to calculate than $f(x)$. We try to find another method to approximate the derivative of $f(x)$.

By the definition of derivative of a function, we have

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

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

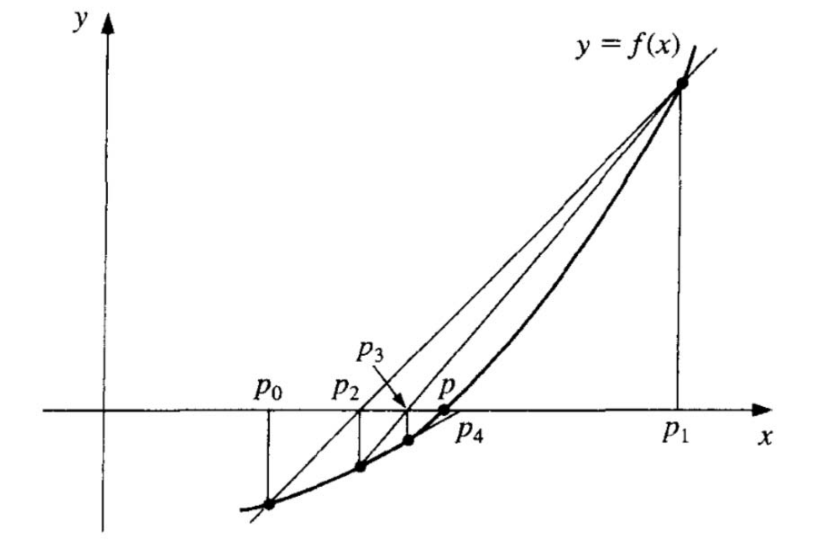
$$f'(p_{n-1}) \approx \frac{f(p_{n-2}) - f(p_{n-1})}{p_{n-2} - p_{n-1}} = \frac{f(p_{n-1}) - f(p_{n-2})}{p_{n-1} - p_{n-2}}.$$

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

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

This technique is called the Secant method, see Figure 2.9.

Figure 2.9



ALGORITHM 2.4 Secant method

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

Step 2 While $i \leq N_0$, do step 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); (Procedure completed successfully.) STOP.

Step 5 Set $i = i + 1$.

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

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

Example 2: Use the Secant method to find a solution to $x = \cos x$. Two initial approximations $p_0 = 0.5$ and $p_1 = \pi/4$ is chosen and the calculations are listed in Table 2.5.

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

It can be seen that the convergence of the Secant method is much faster than functional iteration but slighter slower than Newton's method.

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.

Root bracketing: Each successive pair of approximations in the Bisection method brackets a root p of the equation, i.e.,

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

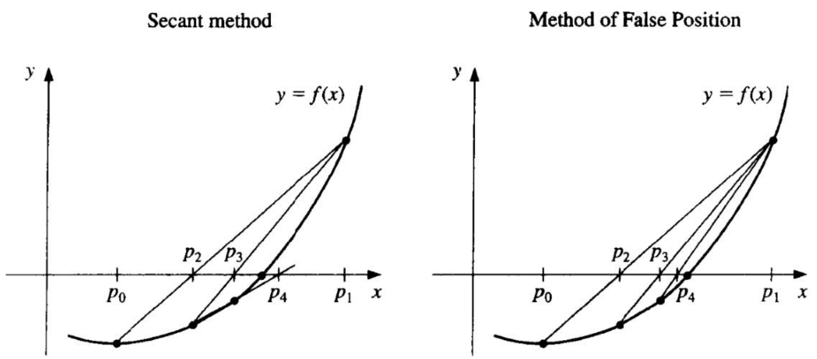
which provides an easily calculated error bound for the approximations. Root bracketing is NOT guaranteed for either Newton's method or the Secant method.

The method of False Positions 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.

2.3.3 Method of False Position (试位法)

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 $f(p_0)f(p_1) < 0$. The approximation p_2 is chosen in same manner as in 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 computer p_3, we need to check $f(p_2) \cdot f(p_1)$ or $f(p_2) \cdot f(p_0)$. If the value of $f(p_2) \cdot f(p_1)$ is negative, then p_1, p_2 bracket a root, and we choose p_3 as the x-intercept of the line joining $(p_1, f(p_1))$ and $(p_2, f(p_2))$. Otherwise p_0, p_2 bracket a root, we choose p_3 as the x-intercept of the line joining $(p_0, f(p_0))$ and $(p_2, f(p_2))$. In a similar manner, we can get a sequence $\{p_n\}_2^\infty$ which approximates to the root, see Figure 2.10.

Figure 2.10



ALGORITHM 2.5: Method of False Position

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 $i \leq N_0$, do step 3-6.

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

Step 4 If $|p - p_1| < TOL$ or $|f(p)| < TOL$ then OUTPUT (p); (Procedure completed successfully.) STOP.

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

Step 6 If $q.q_1 < 0$ then set $p_0 = p, q_0 = q$; else $p_1 = p, q_1 = q$.

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

Example 3: Table 2.6 shows the results of the method of False Position applied to $f(x) = \cos x - x$ with the same initial approximations for the Secant method in Example 2.

It can be seen that the added insurance of the method of False Position requires more calculation than the Secant method.

n	p_n
0	0.5
1	0.7853981635
2	0.7363841388
3	0.7390581392
4	0.7390848638
5	0.7390851305
6	0.7390851332

2.4 Error Analysis for Iteration Methods

In this section, we will 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.

♦Order of Convergence

Definition 2.6 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 (渐近误差) 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.

1. If $\alpha = 1$, the sequence is **linearly convergent**.
2. If $\alpha = 2$, the sequence is **quadratically convergent**.

Example 1 Suppose two sequences $\{p_n\}$ and $\{q_n\}$ both converge to 0, that $\{p_n\}$ is linear with

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

and $\{q_n\}$ is quadratical with the same asymptotic error constant

$$\lim_{n\to\infty} \frac{|q_{n+1}|}{|q_n|^2} = 0.5.$$

Suppose also, for simplicity, that

$$\frac{|p_{n+1}|}{|p_n|} \approx 0.5, \text{ and } \frac{|q_{n+1}|}{|q_n|^2} \approx 0.5.$$

These mean that

$$\begin{aligned}
|p_n - 0| &= |p_n| \approx 0.5|p_{n-1}| \approx 0.5^2|p_{n-2}| \\
&\approx \cdots \approx 0.5^n|p_0|; \\
|q_n - 0| &= |q_n| \approx 0.5|q_{n-1}|^2 \approx 0.5 \times (0.5|q_{n-2}|^2)^2 \\
&= 0.5^3|q_{n-2}|^4 \approx \cdots \approx 0.5^{2^n-1}|q_0|^{2^n}.
\end{aligned}$$

Table 2.7 illustrates the relative speed of convergence of the sequences to 0 when $|p_0| = |q_0| = 1$.

From above example, we can see that: Quadratical convergent sequence generally converges more rapidly than those that converge only linearly, but many techniques that generate convergent sequences do so only linearly.

Think: What will happen when $p_0 = 1$ and $q_0 = 2$?

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}

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 $g'(x)$ is continuous on (a,b) and a positive constant $0 < k < 1$ exists with

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

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

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

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, $\{\xi_n\}_{n=0}^{\infty}$ also converges to p. Since g' is continuous on $[a, b]$, we have

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

Thus,

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

and

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

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

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,

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

Proof:

Choose k in $(0,1)$ and $\delta > 0$ such that on the interval $[p-\delta, p+\delta]$, contained in I, we have $|g'(x)| \le k$ and g'' continuous. Since $|g'(x)| \le k < 1$, 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 $|g'(x)| \le k < 1$ 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, since g'' is continuous,

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

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

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

Problem: **How to construct a fixed point problem $x = g(x)$ to be quadratically convergent associated with a root finding problem $f(x) = 0$?**

Let $g(x)$ be in the form

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

For the iteration procedure derived from $g(x)$ to be quadratically convergent, we need to have $g'(p) = 0$. Since

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

Let $x = p$, we have $g'(p) = 1 - \phi(p)f'(p)$, and $g'(p) = 0$ if and only if $\phi(p) = 1/f'(p)$.

A reasonable approach is to let $\phi(x) = 1/f'(x)$, which is the Newton's method.

It is clear that difficulties might occur if $f'(p_n)$ goes to zero simultaneously with $f(p_n)$. 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 $f(x)$ if for $x \neq p$, we can write $f(x) = (x-p)^m q(x)$, where $\lim_{x \to p} q(x) \neq 0$.

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 $f \in C^1[a,b]$,

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

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

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

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

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

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 if and only if

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

Theorem 2.10 implies that Newton's method converges quadratically to p, provided that p is a simple zero.

However, quadratic convergence may not occur if the zero is not simple.

Example: Use Newton's Method to find a root of $f(x) = x^2$.

This may seem like a trivial problem, since we know there is one root: $p = 0$. But often it is instructive to apply a new method to an example we understand thoroughly. The Newton's Method formula is

$$p_{n+1} = p_n - f(p_n)/f'(p_n) = p_n/2.$$

The surprising result is that Newton's Method simplifies to dividing by two. Since the root is $p = 0$, we have the following table of Newton iterates for initial guess $p_0 = 1$:

n	p_n	$e_n = \lvert p_n - p \rvert$	p_n/p_{n-1}
0	1.000	1.000	
1	0.500	0.500	0.500
2	0.250	0.250	0.500
3	0.125	0.125	0.500
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Newton's Method does converge to the root $p = 0$. The error formula is $e_{n+1} = e_n/2$, so the convergence is linear with convergence proportionality constant $\lambda = 1/2$.

How to deal with this problem?

Let $g(x) = x - \frac{f(x)}{f'(x)}$, it can be easily derived that

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

If p is the simple zero of $f(x)$, then $f(p) = 0,\ f'(p) \neq 0$. In this case, $g'(p) = 0$, Newton's method converges quadratically.

If p is the multiple zero of $f(x)$, for example, zero of multiplicity 2, then $f(x) = (x-p)^2 q(x)$, with $\lim\limits_{x \to p} q(x) \neq 0$. In this case,

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

clearly we have $g'(p) \neq 0$, according to Theorem 2.7, Newton's method converges only linearly.

Example 2: Consider $f(x) = e^x - x - 1$. Since $f(0) = 0, f'(0) = 0$ and $f''(0) = 1$, f has a zero of multiplicity 2 at $p = 0$. 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.

Question: How to handle the problem of multiple roots?

Table 2.8

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		

Define a function μ by

$$\mu(x) = f(x)/f'(x).$$

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

$$\begin{aligned}
\mu(x) &= \frac{(x-p)^m q(x)}{m(x-p)^{m-1}q(x) + (x-p)^m q'(x)} \\
&= (x-p)\frac{q(x)}{mq(p) + (x-p)q'(x)},
\end{aligned}$$

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

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

so p is a simple zero of $\mu(x)$.

Newton's method can be applied to the function 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

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

Theoretically, this method will convergent quadratically, and the only **drawback** 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 (分母) consists of the difference of two numbers that are both close to 0, see Example 3.

2.5 Accelerating Convergence

Theorem 2.7 indicates that it is rare to have the luxury of quadratic convergence. In this section, we consider a technique call 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. That means

$$\lim_{n\to\infty} \frac{|p_{n+1}-p|}{|p_n-p|} = \lambda, \quad (\lambda \neq 0).$$

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

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

so

$$\begin{aligned} & p_{n+1}^2 - 2p_{n+1}p + p^2 \\ \approx \quad & p_{n+2}p_n - (p_n + p_{n+2})p + p^2 \end{aligned}$$

and

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

Solving for p gives

$$\begin{aligned}
p &\approx \frac{p_np_{n+2} - 2p_np_{n+1} + p_n^2 - p_{n+1}^2 + 2p_np_{n+1} - p_n^2}{p_{n+2} - 2p_{n+1} + p_n} \\
&= \frac{p_n(p_{n+2} - 2p_{n+1} + p_n) - (p_{n+1}^2 - 2p_np_{n+1} + p_n^2)}{p_{n+2} - 2p_{n+1} + p_n} \\
&= p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}.
\end{aligned}$$

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

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

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 $p = 1$. Aitken's Δ^2 method can be used to derive the sequence $\{\hat{p}_n\}_{n=1}^{\infty}$, see Table 2.11.

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	

It certainly appears that $\{\hat{p}_n\}_{n=1}^{\infty}$ converges more rapidly to $p = 1$ than does $\{p_n\}_{n=1}^{\infty}$.

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

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

Higher powers of the operator Δ are defined recursively by

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

This implies that

$$\begin{aligned}\Delta^2 p_n &= \Delta(\Delta p_n) = \Delta(p_{n+1} - p_n) \\ &= \Delta p_{n+1} - \Delta p_n = p_{n+2} - 2p_{n+1} + p_n\end{aligned}$$

By this definition, we rewrite the formula

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

as a more simple form

$$\hat{p}_n = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n}.$$

THEOREM 2.13 Suppose that $\{p_n\}_{n=0}^{\infty}$ is a sequence that converges linearly to limit p, and 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

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

Steffensen's Method:

For a fixed iteration problem $p = g(p)$, given initial approximation p_0, let $p_0, p_1 = g(p_0), p_2 = g(p_1)$, and then $\hat{p}_0 = p_0 - (p_1 - p_0)^2/(p_2 - 2p_1 + p_0)$. Assume that $\hat{p}_0$ is a better approximation than p_2, and applies fixed point iteration to $\hat{p}_0$ instead of p_2............

$$p_0,\ p_1 = g(p_0),\ p_2 = g(p_1),$$

$$\hat{p}_0 = p_0 - (p_1 - p_0)^2/(p_2 - 2p_1 + p_0)$$

$\Updownarrow$

$$p_0 = \hat{p}_0,\ p_1 = g(p_0),\ p_2 = g(p_1),$$

$$\hat{p}_0 = p_0 - (p_1 - p_0)^2/(p_2 - 2p_1 + p_0)$$

ALGORITHM 2.6 Steffensen's Algorithm

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 Step 3-6.

Step 3 Set $p_1 = g(p_0), p_2 = g(p_1)$,
$p = p_0 - (p_1 - p_0)^2/(p_2 - 2p_1 + p_0)$.

Step 4 If $|p - p_0| < TOL$, then OUTPUT p, STOP.

Step 5 Set $i = i + 1$.

Step 6 Set $p_0 = p$.

Step 7 OUTPUT (Method failed after N_0 iterations, "$N_0 =$", N_0), STOP.

Example 2: Solve $x^3 + 4x^2 - 10 = 0$ using Steffensen's method. Let $g(x) = (\frac{10}{x+4})^{1/2}$, the fixed-point method used in Example 3(d) of Section 2.2.

Steffensen's procedure with $p_0 = 1.5$ gives the values in Table 2.12.

k	$p_0^{(k)}$	$p_1^{(k)}$	$p_2^{(k)}$
0	1.5	1.348399725	1.367376372
1	1.365265224	1.365225534	1.365230583
2	1.365230013		

The iterate $p_0^{(2)}$ is accurate to the 9th decimal place. In this example, Steffensen's method gives about the same accuracy as Newton's method.

Note that $\Delta^2 p_n$ might be 0, which would introduce a 0 in the denominator of the next iterate. If this occurs, we terminate the sequence and select $p_2^{(n-1)}$ as the best approximation.

The advantage of Steffensen's method is that it gives quadratic convergence without evaluating a derivative.

Theorem 2.14 Suppose that $x = g(x)$ has the solution p with $g'(p) \neq 1$. If there exists a $\delta > 0$ such that $g \in C^3[p-\delta, p+\delta]$, then Steffensen's method gives quadratic convergence for any $p_0 \in [p-\delta, p+\delta]$.

2.6 Zeros of Polynomials and Müller's Method

In this section, we will discuss the root finding methods for a polynomial of order n.

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 $P(x)$, are constants and $a_n \neq 0$.

THEOREM 2.15 (Fundamental Theorem of Algebra) If $P(x)$ is a polynomial of degree $n \geq 1$, then $P(x)$ has at least one (possibly complex) root.

Corollary 2.16 If $P(x)$ is a polynomial of degree $n \geq 1$, then there exist unique constants $x_1, x_2, \cdots, x_k$ (possibly complex), and positive integer $m_1, m_2, \cdots, m_k$, such that $\sum_{i=1}^{n} m_i = n$, and

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

Corollary 2.17 Let $P(x)$ and $Q(x)$ are polynomials of degree at most n, if $x_1, x_2, \cdots, x_k$ with $k > n$ are distinct numbers with $P(x_i) = Q(x_i), i = 1, 2, \cdots, k$, then $P(x) = Q(x)$ for all values of x.

Proof:

Since $P(x)$ and $Q(x)$ are polynomials of degree at most n. Let

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

and

$$Q(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^n$$

are different polynomials of degree at most n.

We set

$$
\begin{aligned}
R(x) &= P(x) - Q(x) \\
&= (a_0 - b_0) + (a_1 - b_1)x + (a_2 - b_2)x^2 \\
&\quad + \cdots + (a_n - b_n)x^n,
\end{aligned}
$$

then $R(x)$ is also a polynomial of degree at most n.

As known condition, there exists $k > n$ distinct points or numbers $x_1, x_2, \cdots, x_k$, such that $R(x_i) = P(x_i) - Q(x_i) = 0$, this implies $R(x) \equiv 0$ for all values of x, or $P(x) = Q(x)$. ■

Horner's Method

To find the roots for a polynomial $P(x)$ using the methods such as Newton's method in previous sections, we need to evaluate $P(x)$ and $P'(x)$ at specified points. Since both $P(x)$ and $P'(x)$ are polynomials, computational efficiency is required for evaluation of these functions. Horner gave a more efficient method to do this. Horner’ s method incorporates the nesting technique, and, as a consequence, requires only n multiplications and n additions to evaluate an arbitrary nth-degree polynomial.

THEOREM 2.18 Let

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

If $b_n = a_n$ and

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

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

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

Proof

By the Definition of $Q(x)$, we have

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

By the hypothesis,

$$b_n = a_n, b_{n-1} - b_n x_0 = a_{n-1}, \cdots, b_1 - b_2 x_0 = a_1, b_0 - b_1 x_0 = a_0,$$

so

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

and $P(x_0) = b_0$.$\blacksquare$

Application of Horner's Method

Using Horner's Method to evaluate the value $P(x_0)$ of a polynomial $P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ at a specified point x_0 equals to find b_0, we can do as follows (the synthetic division).

x_0	a_n	a_{n-1}	a_{n-2}	$\cdots$	a_1	a_0
		$b_n x_0$	$b_{n-1}x_0$	$\cdots$	$b_2 x_0$	$b_1 x_0$
	$b_n = a_n$	b_{n-1}	b_{n-2}	$\cdots$	b_1	b_0

Example 1 Use Horner's method to evaluate $P(x) = 2x^4 - 3x^2 + 3x - 4$ at $x_0 = -2$.

$x_0 = -2$	$a_4 = 2$	$a_3 = 0$	$a_2 = -3$	$a_1 = 3$	$a_0 = -4$
		$b_4 x_0 = -4$	$b_3 x_0 = 8$	$b_2 x_0 = -10$	$b_1 x_0 = 14$
	$b_4 = 2$	$b_3 = -4$	$b_2 = 5$	$b_1 = -7$	$b_0 = 10 \quad = P(-2)$

So,

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

Computing the derivative $P'(x_0)$ of $P(x)$ at a given point x_0:

Since $P(x) = (x - x_0)Q(x) + b_0$, thus differentiating with respect to x, gives

$$P'(x) = Q(x) + (x - x_0)Q'(x) \Rightarrow P'(x_0) = Q(x_0).$$

Due to $Q(x)$ is also a polynomial of degree at most $n - 1$, so Horner's Method can be used to get $Q(x_0)$, which equals to $P'(x_0)$.

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.

By Horner's method, since

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

Let $Q(x) = (x - x_0)R(x) + c_1$, where

$$R(x) = c_n x^{n-2} + c_{n-1}x^{n-3} + \cdots + c_3 x + c_2.$$

Thus

$$\begin{aligned}
Q(x) &= (x - x_0)R(x) + c_1 \\
&= (x - x_0)(c_n x^{n-2} + c_{n-1}x^{n-3} + \cdots + c_3 x + c_2) + c_1 \\
&= c_n x^{n-1} + (c_{n-1} - c_n x_0)x^{n-2} + (c_{n-2} - c_{n-1}x_0)x^{n-3} \\
&\quad + \cdots + (c_2 - c_3 x_0)x + (c_1 - c_2 x_0) \\
&= b_n x^{n-1} + b_{n-1}x^{n-2} + \cdots + b_2 x + b_1.
\end{aligned}$$

$\Rightarrow c_n = b_n, c_k = b_k + c_{k+1}x_0, k = n-1, n-2, \cdots, 2, 1$

and $Q(x_0) = c_1 = P'(x_0)$.

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

$x_0 = -2$	2	0	-3	3	-4
		-4	8	-10	14
	2	-4	5	-7	10$= P(-2)$
		-4	16	-42	
	2	-8	21	-49	$= Q(-2) = P'(-2)$

Therefore, $P(-2) = 10$ and $P'(-2) = -49$, which can be used in Newton's method to find the zero of $P(x)$.

ALGORITHM 2.7 Horner's Method

To compute the value $P(x_0)$ of a polynomial

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

and its derivative $P'(x_0)$.

INPUT INPUT degree n; Coefficients $a_0, a_1, a_2, \cdots, a_n$ of polynomial $P(x)$; Point x_0.

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

Step 2 Set $y = a_n$ (compute b_n for P);
$z = a_n$ (compute b_{n-1} for Q).

Step 3 For $j = n-1, n-2, \cdots, 1$, set
$y = x_0 y + a_j$; (compute b_j for P)
$z = x_0 z + y$. (compute b_{j-1} for Q)

Step 4 Set $y = x_0 y + a_0$. (compute b_0 for P)

Step 5 OUTPUT: (y, z); STOP.

★ Using the Newton's method to solve a root of a polynomial with the Horner's method to compute the values $P(x_0)$ and $P'(x_0)$, the algorithm can be described as follows:

Step 1 INPUT degree n; Coefficients $a_0, a_1, a_2, \cdots, a_n$ of polynomial $P(x)$; initial approximation x_0, tolerance TOL, Maximum iteration number N_0.

Step 2 Set $n = 1$ and $p_0 = x_0$.

Step 3 Set $y = a_n$ (compute b_n for P);
$z = a_n$ (compute b_{n-1} for Q);

Step 4 For $j = n-1, n-2, \cdots, 1$,
set $y = p_0 y + a_j$; (compute b_j for P)
$z = p_0 z + y$; (compute b_{j-1} for Q)

Step 5 Set $y = p_0 y + a_0$, (compute b_0 for P)

Step 6 Compute Newton's approximation $p = p_0 - y/z$;

Step 7 If $|p - p_0| < TOL$, output p, STOP.

Step 8 Set $n = n + 1, p_0 = p$.

Step 9 If $n \leq N_0$, goto Step 3.

Step 10 OUTPUT: (Method failed), STOP.

Remarks:

1. Using Newton's method with the help of Horner's method each time, we can get an approximation zero of a polynomial $P(x)$. Suppose that if the Nth iteration, x_N, in the Newton-Raphson procedure, is an approximation zero of $P(x)$, then

$$\begin{aligned} P(x) &= (x - x_N)Q(x) + b_0 = (x - x_N)Q(x) + P(x_N) \\ &\approx (x - x_N)Q(x); \end{aligned}$$

Let $\hat{x}_1 = x_N$ be the approximate zero of P, and $Q_1(x) \equiv Q(x)$ be the approximate factor, then we have

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

To find the second approximate zero of $P(x)$, we can use the same procedure to $Q_1(x)$, give

$$Q_1(x) \approx (x - \hat{x}_2)Q_2(x).$$

where $Q_2(x)$ is a polynomial of degree $n-2$.

Thus

$$P(x) \approx (x - \hat{x}_1)Q_1(x) \approx (x - \hat{x}_1)(x - \hat{x}_2)Q_2(x).$$

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.

2. This method is called **deflation method**（压缩技术）.

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 $P(x)$.

3. If a polynomial has complex roots, how can we get them by Newton's method? If the initial approximation using Newton's method is a real number, all subsequent approximations will also be real numbers.

One way to solve complex root finding problem during the use of Newton's method is to begin with a complex initial approximation and do all computations using complex arithmetic.

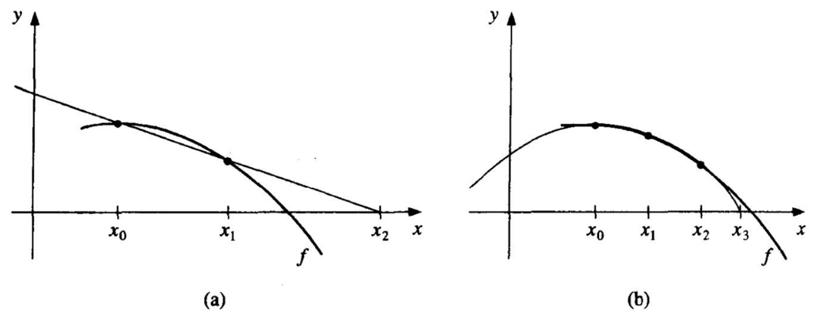
THEOREM 2.19 If $z = a + bi$ is a complex zero of multiplicity m of the polynomial $P(x)$ with real coefficients, then $\bar{z} = a - bi$ is also a zero of multiplicity m of the polynomial $P(x)$, and $(x^2 - 2ax + a^2 + b^2)^m$ is a factor of $P(x)$.

Complex Zeros: Müller's Method

In this part, we consider another method to solve root finding problems especially for approximating the zeros of polynomials.

Müller's method is first presented by D.E.Müller in 1956, and can be thought as an extension of the Secant method. It uses three initial approximations, x_0, x_1 and x_2, and determines the next approximation x_3 by considering the intersection of the x-axis with the parabola (抛物线) through $(x_0, f(x_0)), (x_1, f(x_1))$ and $(x_2, f(x_2))$, see Figure 2.12(b).

Figure 2.12



It is clear that three point can only determine a quadratic polynomial $P(x)$. Suppose that $P(x)$ has the form

$$P(x) = a(x - x_2)^2 + b(x - x_2) + c$$

that passes through $(x_0, f(x_0)), (x_1, f(x_1))$ and $(x_2, f(x_2))$. Then we have

$$\begin{cases} f(x_0) = a(x_0 - x_2)^2 + b(x_0 - x_2) + c, \\ f(x_1) = a(x_1 - x_2)^2 + b(x_1 - x_2) + c, \\ f(x_2) = a \times 0 + b \times 0 + c = c, \end{cases}$$

Solve this equations, we can get the coefficients a, b, c of $P(x)$.

$$
\begin{aligned}
a &= \frac{\frac{f(x_0)-f(x_2)}{x_0-x_2} - \frac{f(x_1)-f(x_2)}{x_1-x_2}}{x_0 - x_1}, \\
&= \frac{\frac{f(x_0)-f(x_1)+f(x_1)-f(x_2)}{x_0-x_2} - \frac{f(x_1)-f(x_2)}{x_1-x_2}}{x_0 - x_1} \\
&= \frac{\frac{f(x_0)-f(x_1)}{x_0-x_2} + (\frac{1}{x_0-x_2} - \frac{1}{x_1-x_2})(f(x_1) - f(x_2))}{x_0 - x_1} \\
&= \frac{\frac{x_0-x_1}{x_0-x_2}\frac{f(x_1)-f(x_0)}{x_1-x_0} + \frac{x_1-x_0}{x_0-x_2}\frac{f(x_2)-f(x_1)}{x_2-x_1}}{x_0 - x_1} \\
&= \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2 - x_0} \\
b &= \frac{f(x_2) - f(x_1)}{x_2 - x_1} + (x_2 - x_1)a, \\
c &= f(x_2).
\end{aligned}
$$

To determine the intersection x_3 , or a zero of quadratic polynomial $P(x)$, we apply the quadratic formula to $P(x) = 0$, and get

$$\begin{aligned} x - x_2 &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \\ &= \frac{(-b \pm \sqrt{b^2 - 4ac})(-b \mp \sqrt{b^2 - 4ac})}{2a(-b \mp \sqrt{b^2 - 4ac})}. \end{aligned}$$

so

$$x - x_2 = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

Let $x = x_3$, thus above formula gives two solutions or possibilities for the approximation x_3. In Müller's method, the sign is chosen to agree with the sign of b.

$$x_3 = x_2 - \frac{2c}{b + \operatorname{sign}(b)\sqrt{b^2 - 4ac}}$$

Chosen in this manner, the denominator will be the largest in magnitude and will result in x_3 being selected as the closest zero of P to x_2.

Once x_3 is determined, the procedure is reinitialized using x_1, x_2, x_3 in place of x_0, x_1 and x_2 to determine next approximation x_4. The method continues until 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 Müller's Algorithm

To find a solution to $f(x) = 0$ given three approximations x_0, x_1 and x_2.

INPUT x_0, x_1, x_2; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set

$$\begin{aligned}
h_1 &= x_1 - x_0, h_2 = x_2 - x_1; \\
\delta_1 &= (f(x_1) - f(x_0))/h_1, \delta_2 = (f(x_2) - f(x_1))/h_2; \\
a &= (\delta_2 - \delta_1)/(h_2 + h_1), \\
i &= 3.
\end{aligned}$$

Step 2 While $i \leq N_0$, do Step 3-7.

Step 3 $b = \delta_2 + h_2 a, d = (b^2 - 4 * a * f(x_2))^{1/2}$. (Note: maybe complex arithmetic.)

Step 4 If $|b-d| < |b+d|$, then $e = b+d$, else $e = b-d$.

Step 5 Let $h = -2f(x_2)/e; p = x_2 + h$.

Step 6 If $|h| < TOL$, then OUTPUT p (Procedure completed successfully),STOP.

Step 7 Set (To prepare next iteration)

$$\begin{aligned}
x_0 &= x_1, x_1 = x_2, x_2 = p; \\
h_1 &= x_1 - x_0, h_2 = x_2 - x_1; \\
\delta_1 &= (f(x_1) - f(x_0))/h_1, \delta_2 = (f(x_2) - f(x_1))/h_2; \\
a &= (\delta_2 - \delta_1)/(h_2 + h_1), \\
i &= i + 1.
\end{aligned}$$

Step 8 OUTPUT ('Method failed after N_0 iteration', '$N_0 =$', N_0), STOP.

Example 3 Consider the polynomial $f(x) = 16x^4 - 40x^3 + 5x^2 + 20x + 6$. Using Algorithm 2.8 with $TOL = 10^{-5}$ and different values of x_0, x_1 and x_2 produces the results in Table 2.13.

a.

	$x_0 = 0.5,\quad x_1 = -0.5,\quad x_2 = 0$	
i	x_i	$f(x_i)$
3	$-0.555556 + 0.598352i$	$-29.4007 - 3.89872i$
4	$-0.435450 + 0.102101i$	$1.33223 - 1.19309i$
5	$-0.390631 + 0.141852i$	$0.375057 - 0.670164i$
6	$-0.357699 + 0.169926i$	$-0.146746 - 0.00744629i$
7	$-0.356051 + 0.162856i$	$-0.183868 \times 10^{-2} + 0.539780 \times 10^{-3}i$
8	$-0.356062 + 0.162758i$	$0.286102 \times 10^{-5} + 0.953674 \times 10^{-6}i$

b.

i	$x_0 = 0.5,\quad x_1 = 1.0,\quad x_2 = 1.5$ x_i	$f(x_i)$
3	1.28785	-1.37624
4	1.23746	0.126941
5	1.24160	0.219440×10^{-2}
6	1.24168	0.257492×10^{-4}
7	1.24168	0.257492×10^{-4}

c.

i	$x_0 = 2.5,\quad x_1 = 2.0,\quad x_2 = 2.25$ x_i	$f(x_i)$
3	1.96059	-0.611255
4	1.97056	0.748825×10^{-2}
5	1.97044	-0.295639×10^{-4}
6	1.97044	-0.259639×10^{-4}

Example 3 illustrates 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.

2.7 Survey of Methods

- The Bisection or the False Position method can be used as starter methods for the Secant or Newton's method.
- Müller's method give rapid convergence (α=1.84) without a particularly good initial approximation. It is not as efficient as Newton's method (α=2), but better than the Secant method (α=1.62), and it has the added advantage of being able to approximate complex roots.
- Müller's method are recommended for finding all the zeros of polynomials, real or complex. Müller's method can also be used for an arbitrary continuous function.