Chapter 2 Solutions of Equations in One Variable

Baodong LIU
baodong@sdu.edu.cn

November 19, 2019

- A system of nonlinear equations in multi-variables has the form

$$\begin{aligned} f_1(x_1, x_2, \cdots, x_n) &= 0 \\ f_2(x_1, x_2, \cdots, x_n) &= 0 \\ \vdots \quad & \quad \vdots \\ f_m(x_1, x_2, \cdots, x_n) &= 0 \end{aligned} \tag{1}$$

- Each function f_i can be thought of as mapping a vector $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n$ into $\mathbb{R}$.

- A system of nonlinear equations in multi-variables has the form

$$\begin{aligned} f_1(x_1, x_2, \cdots, x_n) &= 0 \\ f_2(x_1, x_2, \cdots, x_n) &= 0 \\ \vdots \quad & \quad \vdots \\ f_m(x_1, x_2, \cdots, x_n) &= 0 \end{aligned} \tag{1}$$

- Each function f_i can be thought of as mapping a vector $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n$ into $\mathbb{R}$.

- This system of m nonlinear equations in n unknowns can alternatively be represented by defining a function $\mathbf{f}$, mapping $\mathbb{R}^n$ into $\mathbb{R}^m$ by

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T,$$

- If takes vector notation, then above nonlinear equation system assumes the form

$$\mathbf{f}(\mathbf{x}) = 0 \tag{2}$$

- The function $f_1, f_2, \cdots, f_n$ are the **coordinate functions** of $\mathbf{f}$.

- This system of m nonlinear equations in n unknowns can alternatively be represented by defining a function $\mathbf{f}$, mapping $\mathbb{R}^n$ into $\mathbb{R}^m$ by

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T,$$

- If takes vector notation, then above nonlinear equation system assumes the form

$$\mathbf{f}(\mathbf{x}) = 0 \qquad (2)$$

- The function $f_1, f_2, \cdots, f_n$ are the **coordinate functions** of $\mathbf{f}$.

- This system of m nonlinear equations in n unknowns can alternatively be represented by defining a function $\mathbf{f}$, mapping $\mathbb{R}^n$ into $\mathbb{R}^m$ by

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T,$$

- If takes vector notation, then above nonlinear equation system assumes the form

$$\mathbf{f}(\mathbf{x}) = 0 \tag{2}$$

- The function $f_1, f_2, \cdots, f_n$ are the **coordinate functions** of $\mathbf{f}$.

- The function $\mathbf{f}$ is continuous at $\mathbf{x}_0 \in D$ provided $\lim\limits_{\mathbf{x}\to\mathbf{x}_0} \mathbf{f}(\mathbf{x})$ exists and $\lim\limits_{\mathbf{x}\to\mathbf{x}_0} \mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_0)$.
- In addition, $\mathbf{f}$ is said to be continuous on the set D, if $\mathbf{f}$ is continuous at each $\mathbf{x}$ in D. This concept is expressed by writing

$$\mathbf{f} \in C(D).$$

- The function $\mathbf{f}$ is continuous at $\mathbf{x}_0 \in D$ provided $\lim_{\mathbf{x}\to\mathbf{x}_0} \mathbf{f}(\mathbf{x})$ exists and $\lim_{\mathbf{x}\to\mathbf{x}_0} \mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_0)$.
- In addition, $\mathbf{f}$ is said to be continuous on the set D, if $\mathbf{f}$ is continuous at each $\mathbf{x}$ in D. This concept is expressed by writing

$$\mathbf{f} \in C(D).$$

Theorem

Let f be a function from $D \subset \mathbb{R}^n$ into $\mathbb{R}$ and $\mathbf{x}_0 \in D$. If constants $\delta > 0$ and $K > 0$ exist with

$$\left|\frac{\partial f(\mathbf{x})}{\partial x_j}\right| \leq K, \texttt{ for each } j = 1, 2, \cdots, n$$

whenever $\|\mathbf{x} - \mathbf{x}_0\| < \delta$, and $\mathbf{x} \in D$, then f is continuous at $\mathbf{x}_0$.

- **the root-finding problem:** Given a function $f(x)$ in one variable x, finding a root x of an equation of the form $f(x) = 0$.
- Solution x is called a **root of equation** $f(x) = 0$, or **zero of function** $f(x)$
- Such kind of problem is known as **root finding or zero finding problem.**

2.1 The Bisection Method for Root-finding Problem in one variable

- **the root-finding problem:** Given a function $f(x)$ in one variable x, finding a root x of an equation of the form $f(x) = 0$.
- Solution x is called a **root of equation** $f(x) = 0$, or **zero of function** $f(x)$
- Such kind of problem is known as **root finding or zero finding problem.**

- **the root-finding problem:** Given a function $f(x)$ in one variable x, finding a root x of an equation of the form $f(x) = 0$.
- Solution x is called a **root of equation** $f(x) = 0$, or **zero of function** $f(x)$
- Such kind of problem is known as **root finding or zero finding problem.**

Interval Bisection Method

- By the Intermediate Value Theorem, if

$$f \in C[a, b], \text{ and } f(a)f(b) < 0,$$

then there exists at least a point $x^* \in (a, b)$, such that $f(x^*) = 0$.

- Bisection(折半查找) or Binary-search(二分法) method begins with an initial bracket $[a, b]$, and successively reduce its length half with opposite endpoints, until the solution has been isolated as accurately as desired..
- Although the procedure will work for the case when $f(a)$ and $f(b)$ have opposite signs and maybe there is more than one root in the interval (a, b).

Interval Bisection Method

- By the Intermediate Value Theorem, if

$$f \in C[a, b], \text{ and } f(a)f(b) < 0,$$

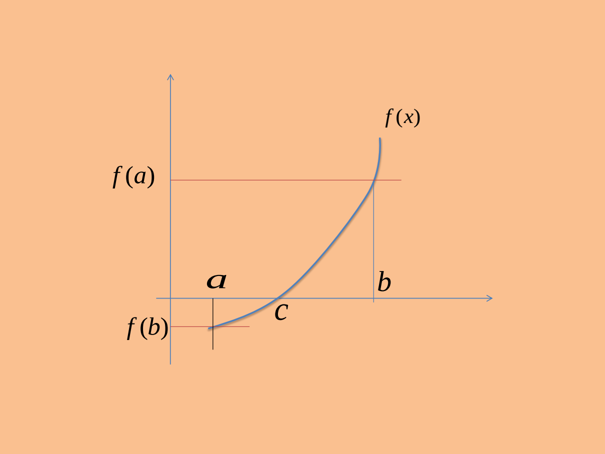
then there exists at least a point $x^* \in (a, b)$, such that $f(x^*) = 0$.

- Bisection(折半查找) or Binary-search(二分法) method begins with an initial bracket $[a, b]$, and successively reduce its length half with opposite endpoints, until the solution has been isolated as accurately as desired..
- Although the procedure will work for the case when $f(a)$ and $f(b)$ have opposite signs and maybe there is more than one root in the interval (a, b).

Interval Bisection Method

- By the Intermediate Value Theorem, if
$$f \in C[a, b], \text{ and } f(a)f(b) < 0,$$
then there exists at least a point $x^* \in (a, b)$, such that $f(x^*) = 0$.
- Bisection(折半查找) or Binary-search(二分法) method begins with an initial bracket $[a, b]$, and successively reduce its length half with opposite endpoints, until the solution has been isolated as accurately as desired..
- Although the procedure will work for the case when $f(a)$ and $f(b)$ have opposite signs and maybe there is more than one root in the interval (a, b).

Geometric means—-Interval Bisection Method



- Let $a_1 = a, b_1 = b$ and $c_1 = (a_1 + b_1)/2$ be the midpoint of interval $[a, b]$.
- Compute $f(c_1)$, It is clear that
  - If $f(c_1) = 0$, then $c = c_1$, and c is our solution.
  - Else, if the $f(c_1)$ has the same sign as $f(a_1)$, then set $a_2 = c_1, b_2 = b_1$;
  - Otherwise, set $a_2 = a_1, b_2 = c_1$.

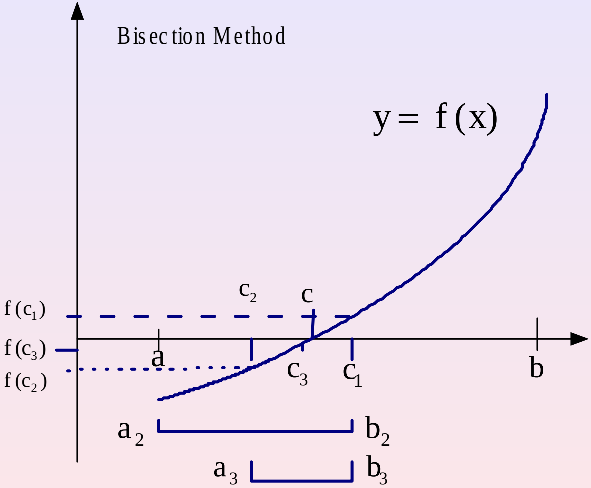
- Let $a_1 = a, b_1 = b$ and $c_1 = (a_1 + b_1)/2$ be the midpoint of interval $[a, b]$.
- Compute $f(c_1)$, It is clear that
  - If $f(c_1) = 0$, then $c = c_1$, and c is our solution.
  - Else, if the $f(c_1)$ has the same sign as $f(a_1)$, then set $a_2 = c_1, b_2 = b_1$;
  - Otherwise, set $a_2 = a_1, b_2 = c_1$.

- Continue this procedure. Suppose we have got the subinterval $[a_n, b_n]$, let $c_n = (a_n + b_n)/2 = a_n + (b_n - a_n)/2,.$
- Compute $f(c_n)$, and determine that
  - If $f(c_n) = 0$ or $|b_n - a_n| < \varepsilon$, where $\varepsilon > 0$ is small enough, then stop and output the solution as $c = c_n$.
  - Otherwise, if $f(c_n)f(a_n) < 0$, then set $a_{n+1} = a_n, b_{n+1} = c_n$, else set $a_{n+1} = c_n, b_{n+1} = b_n$
- Continue this procedure.

- Continue this procedure. Suppose we have got the subinterval $[a_n, b_n]$, let $c_n = (a_n + b_n)/2 = a_n + (b_n - a_n)/2,$.
- Compute $f(c_n)$, and determine that
  - If $f(c_n) = 0$ or $|b_n - a_n| < \varepsilon$, where $\varepsilon > 0$ is small enough, then stop and output the solution as $c = c_n$.
  - Otherwise, if $f(c_n)f(a_n) < 0$, then set $a_{n+1} = a_n, b_{n+1} = c_n$, else set $a_{n+1} = c_n, b_{n+1} = b_n$
- Continue this procedure.

- Continue this procedure. Suppose we have got the subinterval $[a_n, b_n]$, let $c_n = (a_n + b_n)/2 = a_n + (b_n - a_n)/2,.$
- Compute $f(c_n)$, and determine that
  - If $f(c_n) = 0$ or $|b_n - a_n| < \varepsilon$, where $\varepsilon > 0$ is small enough, then stop and output the solution as $c = c_n$.
  - Otherwise, if $f(c_n)f(a_n) < 0$, then set $a_{n+1} = a_n$, $b_{n+1} = c_n$, else set $a_{n+1} = c_n$, $b_{n+1} = b_n$
- Continue this procedure.

Geometric Means



Algorithm 2.1: Bisection Algorithm

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

OUTPUT approximate solution c or message of failure.

Step 1 Set $k = 1, FA = f(a)$;

Step 2 While $k \leq N$, do Steps 3-6

- Step 3 Set $c = a + (b - a)/2$; and compute $FC = f(c)$.
- Step 4 If $FC = 0$ or $|b - a|/2 < TOL$, then output c, (Procedure complete successfully.) Stop!
- Step 5 If $FA \cdot FC < 0$, then set $b = c$; else set $a = c$
- Step 6 Set $k = k + 1$.

Step 7 OUTPUT "Method failed after N iterations." STOP.

Algorithm 2.1: Bisection Algorithm

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

OUTPUT approximate solution c or message of failure.

Step 1 Set $k = 1, FA = f(a)$;

Step 2 While $k \leq N$, do Steps 3-6

- Step 3 Set $c = a + (b - a)/2$; and compute $FC = f(c)$.
- Step 4 If $FC = 0$ or $|b - a|/2 < TOL$, then output c, (Procedure complete successfully.) Stop!
- Step 5 If $FA \cdot FC < 0$, then set $b = c$; else set $a = c$
- Step 6 Set $k = k + 1$.

Step 7 OUTPUT "Method failed after N iterations." STOP.

Algorithm 2.1: Bisection Algorithm

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

OUTPUT approximate solution c or message of failure.

Step 1 Set $k = 1, FA = f(a)$;

Step 2 While $k \leq N$, do Steps 3-6

- Step 3 Set $c = a + (b - a)/2$; and compute $FC = f(c)$.
- Step 4 If $FC = 0$ or $|b - a|/2 < TOL$, then output c, (Procedure complete successfully.) Stop!
- Step 5 If $FA \cdot FC < 0$, then set $b = c$; else set $a = c$
- Step 6 Set $k = k + 1$.

Step 7 OUTPUT "Method failed after N iterations." STOP.

Theorem

Suppose that $f \in C[a, b]$, and $f(a)f(b) < 0$. The Bisection method generates a sequence $\{p_n\}_1^\infty$ approximating a zero point p of f with

$$|p_n - p| \leq \frac{b-a}{2^n}, n \geq 1. \blacksquare$$

Proof:

- By the procedure, we know that

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

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

$$|p_n - p| \leq \frac{|b_n - a_n|}{2} = \frac{|b - a|}{2^n}. \blacksquare$$

Proof:

- By the procedure, we know that

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

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

$$|p_n - p| \leq \frac{|b_n - a_n|}{2} = \frac{|b - a|}{2^n}. \blacksquare$$

Remarks on Bisection method

- Other Stopping Criteria for Iteration procedures with a given tolerance $\varepsilon > 0$:

$$
\begin{aligned}
|p_n - p_{n-1}| &< \varepsilon \\
\frac{|p_n - p_{n-1}|}{|p_n|} &< \varepsilon \\
|f(p_n)| &< \varepsilon
\end{aligned}
$$

Remarks on Bisection method

- Since

$$|p_n - p| \leq \frac{|b_n - a_n|}{2} = \frac{|b - a|}{2^n}$$

- The Sequence $\{p_n\}_{n=1}^{\infty}$ converges to p with rate of convergence $O(\frac{1}{2^n})$, that is

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

- Bisection is certain to converge, but does so slowly
- Given starting interval $[a, b]$, length of interval after k iterations is $(b-a)/2^k$, so achieving error tolerance of ε $(\frac{(b-a)}{2^k} < \varepsilon)$ requires $k \approx [\log_2^{\frac{b-a}{\varepsilon}}]$ iterations, regardless of function f involved.

- Since

$$|p_n - p| \le \frac{|b_n - a_n|}{2} = \frac{|b - a|}{2^n}$$

- The Sequence $\{p_n\}_{n=1}^{\infty}$ converges to p with rate of convergence $O(\frac{1}{2^n})$, that is

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

- Bisection is certain to converge, but does so slowly
- Given starting interval $[a, b]$, length of interval after k iterations is $(b-a)/2^k$, so achieving error tolerance of ε ($\frac{(b-a)}{2^k} < \varepsilon$) requires $k \approx [\log_2^{\frac{b-a}{\varepsilon}}]$ iterations, regardless of function f involved.

- **Fixed point** of given function $g : \mathbb{R} \to \mathbb{R}$ is value x^* such that $x^* = g(x^*)$
- Many iterative methods for solving nonlinear equations use fixed-point iteration scheme of form

$$x_{k+1} = g(x_k)$$

where fixed points for g are solutions for $f(x) = 0$.

Fixed Point Method

- **Fixed point** of given function $g : \mathbb{R} \to \mathbb{R}$ is value x^* such that $x^* = g(x^*)$
- Many iterative methods for solving nonlinear equations use fixed-point iteration scheme of form

$$x_{k+1} = g(x_k)$$

where fixed points for g are solutions for $f(x) = 0$.

Fixed Point Method

- This kind of method is also called **functional iteration**, since function g is applied repeatedly to initial starting value x_0
- For given equation $f(x) = 0$, there may be many equivalent fixed-point problems $x = g(x)$ with different choices for g. 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.

Fixed Point Method

- This kind of method is also called **functional iteration**, since function g is applied repeatedly to initial starting value x_0
- For given equation $f(x) = 0$, there may be many equivalent fixed-point problems $x = g(x)$ with different choices for g. 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.

Fixed Point Method

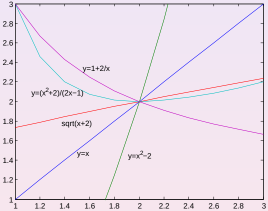
- This kind of method is also called **functional iteration**, since function g is applied repeatedly to initial starting value x_0
- For given equation $f(x) = 0$, there may be many equivalent fixed-point problems $x = g(x)$ with different choices for g. For example, as $g(x) = x - f(x)$ or as $g(x) = x + 3f(x)$.
- 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.

If $f(x) = x^2 - x - 2$, it has two roots $x^* = 2$ and $x^* = -1$. Then fixed points of each of functions

1. $g(x) = x^2 - 2$
2. $g(x) = \sqrt{x+2}$
3. $g(x) = 1 + \frac{2}{x}$
4. $g(x) = \frac{x^2+2}{2x-1}$

are solutions to equation $f(x) = 0$.

Examples for Fixed Point Problems



- 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

$$\begin{cases} \texttt{Given} \quad p_0 \\ p_n = g(p_{n-1}), n = 0, 1, \cdots , \end{cases}$$

  for each $n \geq 1$.
- If the sequence $\{p_n\}_{n=0}^{\infty}$converges to p and $g(x)$ is continuous, then we have

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

  and a solution to $x = g(x)$ is obtained.
- This technique is called **fixed point iteration(or functional iteration)**.

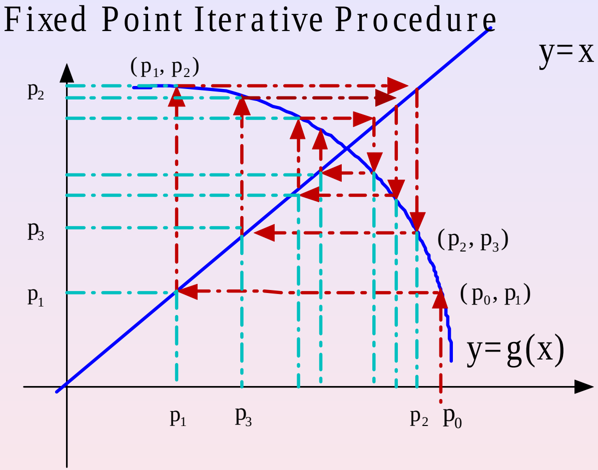


Fig.2-3. Fixed point iteration procedure.(a)

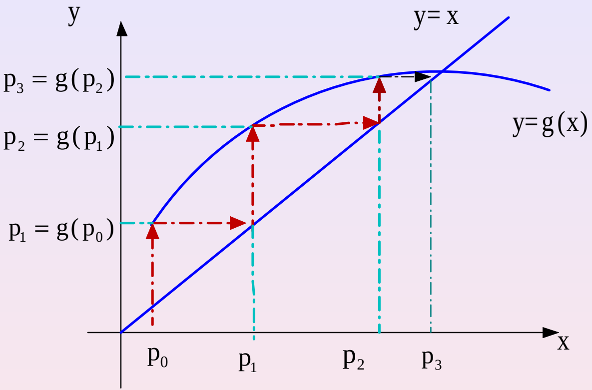


Fig.2-3. Fixed point iteration procedure.(b)

ALGORITHM 2.2 Fixed-Point Iteration Method

INPUT Initial approximation p_0, tolerance TOL,
Maximum number of iteration N.

OUTPUT approximation solution p or message of failure.

Step 1 Set $n = 1$.

Step 2 While $n \leq N$, do Step3-6.

Step 3 Set $p = g(p_0)$.

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

Step 5 Set $n = n + 1, p_0 = p$.

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

Sufficient Conditions 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 of Theorem: Existence

- If $g(a) = a$ or $g(b) = b$, then $g(x)$ has a fixed point at an endpoint.
- Suppose 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 $h(x) = g(x) - x$ which $h(p) = 0$.
- Thus $g(p) - p = 0$, and p is a fixed point of $g(x)$.

Uniqueness

- 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. ■■

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 of Theorem 2.3:

- 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.$\blacksquare$.

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

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

and

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

Proof:

- The first bound can be derived as follows:

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

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

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

- 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

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

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 $f \in C^2[a, b]$, and x^* is a solution of $f(x) = 0$.
- Let $\bar{x} \in [a, b]$ be an approximation to x^* such that $f'(\bar{x}) \neq 0$ and $|\bar{x} - x^*|$ 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 $\bar{x}$.

Newton's Method,Continued

- Since $f(x^*) = 0$, let $x = x^*$ in this equation, and gives

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

- Newton's method is derived by assuming that since $|x^* - \bar{x}|$ is small, thus the term involving $(x^* - \bar{x})^2$ is much smaller.
- Omit the last term, and gives

$$0 = f(x^*) \approx f(\bar{x}) + (x^* - \bar{x})f'(\bar{x}),$$

- Solving for x^* in this equation gives

$$x^* \approx \bar{x} - \frac{f(\bar{x})}{f'(\bar{x})}$$

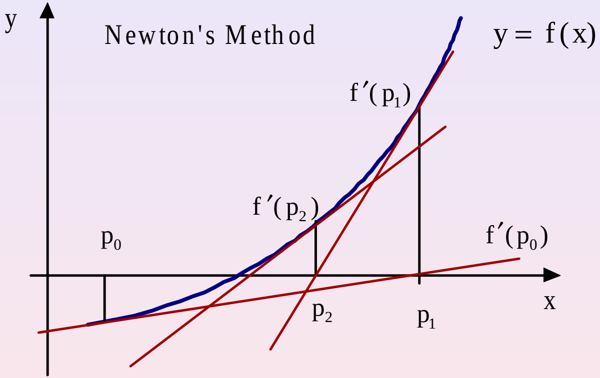
I. The Newton-Raphson Method–牛顿法或切线法

- Starts with an initial approximation x_0
- Defined iteration scheme by

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, \forall n \geq 1$$

- This scheme generates the sequence $\{x_n\}_0^\infty$

Geometric Explanation for Newton's Method



ALGORITHM 2.3 Newton-Raphson 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.

OUTPUT approximate solution p or message of failure.

Step 1 Let $i = 1$.

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

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

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

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

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

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 of Theorem 2.5

- 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 $x \in [p-\delta_1, p+\delta_1] \subset C[a.b]$.
- Thus, g is defined and continuous on $[p-\delta_1, p+\delta_1]$.

Proof: Continued

- 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

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

- It remains to show that

$$g \in [p - \delta, p + \delta] \mapsto [p - \delta, p + \delta].$$

- 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 $g \in [p-\delta, p+\delta] \mapsto [p-\delta, p+\delta]$.
- 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

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

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

- Use Newton's method to find root of equation

$$f(x) = x^2 - 4\sin(x) = 0$$

- Derivative is

$$f'(x) = 2x - 4\cos(x).$$

- So iteration scheme is

$$x_{k+1} = x_k - \frac{x_k^2 - 4\sin(x_k)}{2x_k - 4\cos(x_k)}$$

Taking $x_0 = 3$ as starting value, we obtain

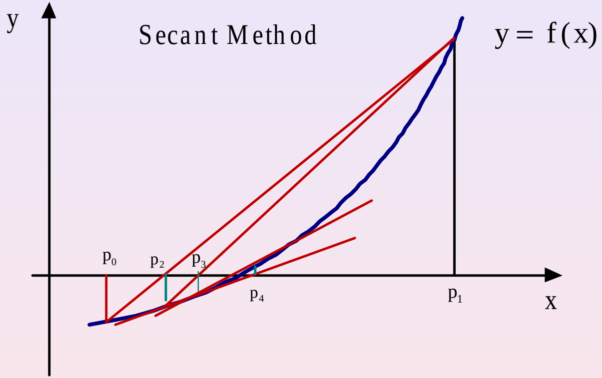
k	x	$f(x)$	$f'(x)$
0	3.000000	8.435520	9.959970
1	2.153058	1.294772	6.505771
2	1.954039	0.108438	5.403795
3	1.933972	0.001152	5.288919
4	1.933754	0.000000	5.287670

- **Remark:** For Newton's method, each iteration requires evaluation of both **function** $(f(x_k))$ and its **derivative**$(f'(x_k))$, which may be inconvenient or expensive.
- **Improvement:** Derivative is approximated by finite difference using two successive iterates, so iteration becomes

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

- This method is known as **secant method**.

Secant Method, continued



Example: Secant Method

Using Secant's method to find a root of equation

$$f(x) = x^2 - 4\sin(x) = 0$$

Taking $x = 1, 3$ as starting values, we obtain

k	x_k	$f(x_k)$
0	1.0000	−2.3659
1	3.0000	8.4355
2	1.4381	−1.8968
3	1.7248	−0.9777
4	2.0298	0.5343
5	1.9220	−0.0615
6	1.9332	−0.0031
7	1.9338	0.0000

Secant Algorithm 2.4

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.)

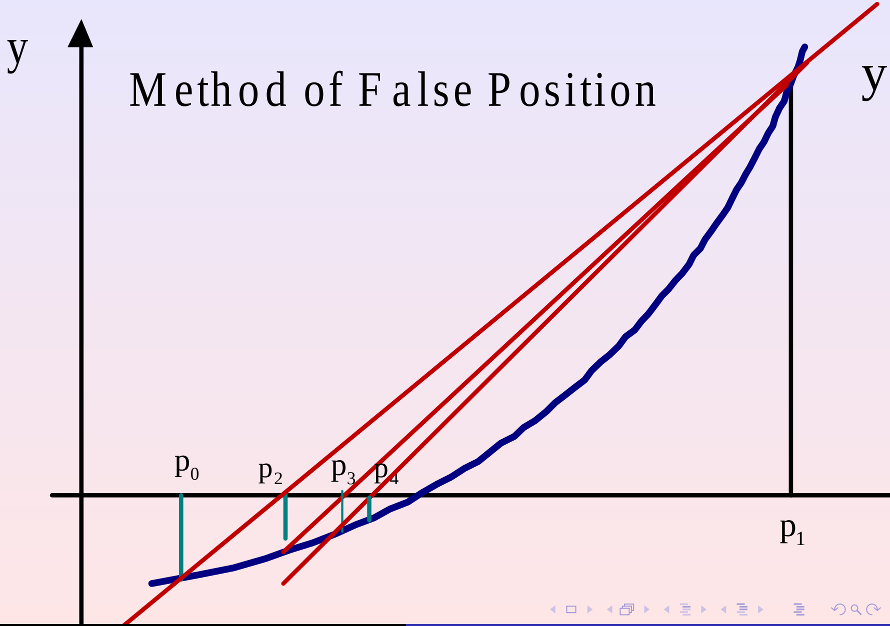
Step 7 OUTPUT ('Method failed after N_0 iterations, $N_0 =$', N_0); (Procedure completed unsuccessfully.) STOP.

III. Method of False Position–错位法

- To find a solution to $f(x) = 0$ for a 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 this value 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 $(p_2, f(p_2))$.
- In a similar manner, we can get a sequence $\{p_n\}_2^\infty$ which approximates to the root.


Method of False Position
y
y
p0
p2
p3
p4
p1

False Position Algorithm 2.5

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 $|p - p_1| < TOL$ 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.

In this section , we will investigate

- **The rate of convergence** of a sequence;
- **The order of convergence** of functional iteration schemes;
- **Ways of accelerating the convergence of Newton's method.**

Definition for measuring the rate of a sequence 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 constant** λ.$\blacksquare$

- An iterative technique of the form

$$p_n = g(p_{n-1})$$

is said to be **of order** α if the sequence $\{p_n\}_{n=0}^{\infty}$ (generated by $p_n = g(p_{n-1}), n = 1, 2, \cdots$) 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**.

Example 3

- Suppose two sequences $\{p_n\} \mapsto 0$ and $\{q_n\} \mapsto 0$
- Further, we also suppose that

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

- 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

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

- Quadratical convergent sequence generally converges more rapidly than those that converge only linearly.

Example 3

- Suppose two sequences $\{p_n\} \mapsto 0$ and $\{q_n\} \mapsto 0$
- Further, we also suppose that

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

- 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

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

- Quadratical convergent sequence generally converges more rapidly than those that converge only linearly.

Example 3

- Suppose two sequences $\{p_n\} \mapsto 0$ and $\{q_n\} \mapsto 0$
- Further, we also suppose that

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

- 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

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

- Quadratical convergent sequence generally converges more rapidly than those that converge only linearly.

Example 3

- Suppose two sequences $\{p_n\} \mapsto 0$ and $\{q_n\} \mapsto 0$
- Further, we also suppose that

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

- 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

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

- Quadratical convergent sequence generally converges more rapidly than those that converge only linearly.

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

$$|g'(x)| \leq k,$$

for all $x \in (a,b)$.
- If $g'(p) \neq 0$, then for any number p_0 in $[a,b]$ the sequence $p_n = g(p_{n-1})$, for $n \geq 1$, converges **only linearly to the unique fixed point** p in $[a,b]$. ■

Proof of Theorem 2.7:

- 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, and ξ_n is between p_n and p, thus $\{\xi_n\}_{n=0}^{\infty}$ also converges to p.
- By the known condition, g' is continuous on [a, b], so we have

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

Continued

- 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,

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

Proof of Theorem 2.8:

- Since $g'(p) = 0$ and $g''(x)$ is continuous on the open interval I, so we can choose a positive k $(0 < k < 1)$ 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, since g'' is continuous,

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

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

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

$$\lim_{x \to p} q(x) \neq 0. \blacksquare$$

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 $f'(p) \neq 0$. $\blacksquare$

Proof of Theorem 2.10

- 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

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

but

$$f^{(m)}(p) \neq 0. \blacksquare$$

Method to handle multiple root finding problems:

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

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

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 zero of multiplicity 1 of $\mu(x)$.

- Newton's method can be applied to the function μ to give

$$\begin{aligned} g(x) &= x - \frac{\mu(x)}{\mu'(x)} \\ &= x - \frac{f(x)/f'(x)}{[f'(x)^2 - f(x)f''(x)]/f'(x)^2}, \end{aligned}$$

or

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

1. Newton's method transforms nonlinear equation $f(x) = 0$ into fixed-point problem $x = g(x)$, where $g(x) = x - f(x)/f'(x)$ and hence
$$g'(x) = f(x)f''(x)/(f'(x))^2$$
2. If p is simple root (
$$\texttt{i.e.,} f(p) = 0 \text{ and } f'(p) \neq 0,$$
then $g'(p) = 0$, thus Convergence rate of **Newton's method** for simple root is therefore **quadratic** $(r = 2)$
3. But iterations must **start close enough to root** to converge.

Multiple Root Problem with Newton's Method

- Suppose equation $f(x)$ has m multiplicity at p, then we can rewrite it as $f(x) = (x-p)^m q(x)$.
- Thus by Newton's method, we have

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

- So

$$g(p) = p$$

and

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

- **Conclusion:**
  - For a simple root, the Newton's method has quadratic convergence rate;
  - For multiple root,the Newton's method is only linear convergent.

- To avoid multiple root, we define a new function μ by
$$\mu(x) = f(x)/f'(x).$$
- If p is a zero of multiplicity m and $f(x)$ then we can rewrite it as
$$\begin{aligned}\mu(x) &= \frac{(x-p)^m q(x)}{m(x-p)^{m-1}q(x) + (x-p)^m q'(x)} \\ &= (x-p)\frac{q(x)}{mq(p) + (x-p)q'(x)},\end{aligned}$$
also has a zero at p.

- However, since $q(p) \neq 0$,

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

so p is a zero of multiplicity 1 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

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

- Thus the convergence rate is also quadratic.

Convergence Rate OF Fixed Point

- Let p be a fixed point, p_{k+1} be the approximate solution at the kth iteration, generated by

$$p_{k+1} = g(p_k),$$

then the error can be described as

$$e_{k+1} = p_{k+1} - p = g(p_k) - g(p).$$

- Suppose $g(x) \in C^1[a, b]$, then by the Mean Value Theorem, there is a point θ_k between p_k and p, such that

$$\begin{aligned} e_{k+1} &= p_{k+1} - p = g(p_k) - g(p) \\ &= g'(\theta_k)(p_k - p) \\ &= g'(\theta_k) e_k \end{aligned}$$

- Since $|g'(p)| < 1$, and the starting iteration close enough to p, we can assure that there exist a constant C, such that

$$|g'(\theta_k)| < C < 1, k = 0, 1, \cdots$$

- Thus we have

$$|e_{k+1}| \leq C|e_k| \leq C^2|e_{k-1}| \leq \cdots \leq C^{k+1}|e_0|.$$

- But $C < 1$ implies that $C^k \to 0$, so $e_k \to 0, k \to \infty$, and the sequence converges to the solution p.

- And we also can see that

$$\lim_{k\to\infty}\frac{|e_{k+1}|}{|e_k|}=\lim_{k\to\infty}g'(\theta_k)=g'(p)$$

So the asymptotic convergence rate is linear.

- Further if $g'(p)=0$, then by the Taylor's theorem

$$e_{k+1}=p_{k+1}-p=g(p_k)-g(p)=g''(\xi_k)(p_k-p)^2/2$$

for some ξ_k between p_k and p.

- Thus

$$\lim_{k\to\infty}\frac{|e_{k+1}|}{|e_k|^2}=\lim_{k\to\infty}\frac{|g''(\xi_k)|}{2}$$

and hence the convergence rate is **at least quadratic.**

Remarks on Convergence of Fixed-Point Iteration

1. **Local Convergent:** If $p = g(p)$ and $|g'(p)| < 1$, then there is an interval containing p such that iteration $p_{k+1} = g(p_k)$ converges to p if started with a point within that interval.
2. If $|g'(p)| > 1$, then iterative scheme **diverges** with any starting point other than p.
3. **Convergence rate** of fixed-point iteration is usually **linear**, with constant $C = |g'(p)|$
4. But if $g'(p) = 0$, then convergence rate is **at least quadratic**.

1. **Local Convergent:** If $p = g(p)$ and $|g'(p)| < 1$, then there is an interval containing p such that iteration $p_{k+1} = g(p_k)$ converges to p if started with a point within that interval.
2. If $|g'(p)| > 1$, then iterative scheme **diverges** with any starting point other than p.
3. **Convergence rate** of fixed-point iteration is usually **linear**, with constant $C = |g'(p)|$
4. But if $g'(p) = 0$, then convergence rate is **at least quadratic**.

Remarks on Convergence of Fixed-Point Iteration

1. **Local Convergent:** If $p = g(p)$ and $|g'(p)| < 1$, then there is an interval containing p such that iteration $p_{k+1} = g(p_k)$ converges to p if started with a point within that interval.
2. If $|g'(p)| > 1$, then iterative scheme **diverges** with any starting point other than p.
3. **Convergence rate** of fixed-point iteration is usually **linear**, with constant $C = |g'(p)|$
4. But if $g'(p) = 0$, then convergence rate is **at least quadratic**.

1. **Local Convergent:** If $p = g(p)$ and $|g'(p)| < 1$, then there is an interval containing p such that iteration $p_{k+1} = g(p_k)$ converges to p if started with a point within that interval.
2. If $|g'(p)| > 1$, then iterative scheme **diverges** with any starting point other than p.
3. **Convergence rate** of fixed-point iteration is usually **linear**, with constant $C = |g'(p)|$
4. But if $g'(p) = 0$, then convergence rate is **at least quadratic**.

1. **Convergence rate** of secant method is normally **superlinear,** with $r \approx 1.618$, which is lower than Newton's method.
2. Secant method need to evaluate two previous functions per iteration, there is no requirement to evaluate the derivative.
3. Its disadvantage is that it needs two starting guesses which close enough to the solution in order to converge.

1. **Convergence rate** of secant method is normally **superlinear,** with $r \approx 1.618$, which is lower than Newton's method.
2. Secant method need to evaluate two previous functions per iteration, there is no requirement to evaluate the derivative.
3. Its disadvantage is that it needs two starting guesses which close enough to the solution in order to converge.

1. **Convergence rate** of secant method is normally **superlinear,** with $r \approx 1.618$, which is lower than Newton's method.
2. Secant method need to evaluate two previous functions per iteration, there is no requirement to evaluate the derivative.
3. Its disadvantage is that it needs two starting guesses which close enough to the solution in order to converge.

- 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

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

So when n is sufficiently large,

$$p_n - p, p_{n+1} - p, p_{n+2} - p$$

agree with the same sign as λ, and

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

Then

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

so

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

and

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

Solving for p gives

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

Aitken's Δ^2 method

- Aitken's Δ^2 method is to define a new sequence $\{\hat{p}\}_{n=0}^{\infty}$:

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

- We can prove that the new sequence can converge to p more rapidly than does the originally sequence $\{p_n\}_{n=0}^{\infty}$.

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 $\Delta^k p_n$ are defined recursively by

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

- 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 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

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

The proof of this theorem take as homework.

Special Case: for the sequence generated by fixed point iteration $P_{n+1} = g(P_n)$

For a fixed point iteration, the procedure of convergence accelerating can be shown as follows:

$$p_0^{(0)}, p_1^{(0)} = g(p_0^{(0)}), p_2^{(0)} = g(p_1^{(0)});$$

$$p_0^{(1)} = p_0^{(0)} - \frac{(\Delta p_0^{(0)})^2}{\Delta^2 p_0^{(0)}}, p_1^{(1)} = g(p_0^{(1)}), p_2^{(1)} = g(p_1^{(1)});$$

$$p_0^{(2)} = p_0^{(1)} - \frac{(\Delta p_0^{(1)})^2}{\Delta^2 p_0^{(1)}}, p_1^{(2)} = g(p_0^{(2)}), p_2^{(2)} = g(p_1^{(2)})$$

$$\cdots, \cdots, \cdots;$$

$$p_0^{(n)} = p_0^{(n-1)} - \frac{(\Delta p_0^{(n-1)})^2}{\Delta^2 p_0^{(n-1)}}, p_1^{(n)} = g(p_0^{(n)}), p_2^{(n)} = g(p_1^{(n)})$$

This procedure belongs to **Steffensen**.

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, that is to let
- $p_0 = \hat{p}_0, p_1 = g(p_0), p_2 = g(p_1)$,
- $\hat{p}_0 = p_0 - (p_1 - p_0)^2/(p_2 - 2p_1 + p_0)$.
- ....

Steffensen Algorithm

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.

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.

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

- In this section, we will discuss the root finding methods for a polynomial of order n.
- **Definition 2.14**: 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 $a_i, i = n, n-1, \cdots, 1, 0$ are coefficients of $P(x)$, and $a_n \neq 0$. ■

- In this section, we will discuss the root finding methods for a polynomial of order n.
- **Definition 2.14**: 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 $a_i, i = n, n-1, \cdots, 1, 0$ are coefficients of $P(x)$, and $a_n \neq 0$.■

THEOREM 2.15 (Fundamental Theorem of Algebra:)

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

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

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

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.

- Let

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

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 $P(x) = Q(x)$. ■

- To find the **roots for a polynomial** $P(x) = 0$ 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.

Example: How to find a value at a given point x_0 of $P(x_0) =?$

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

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

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

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

Proof of Theorem 2.18

- 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,

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

- so

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

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

- 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.

- This equals to find b_0.
- **Horner's Method**:

	a_n	a_{n-1}	a_{n-2}	$\cdots$	a_1	a_0
		$+$	$+$	$\cdots$	$+$	$+$
x_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

♦ Computing the derivative $P'(x_0)$ of $P(x)$

- 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 $P'(x_0)$.
- 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

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

- $\Rightarrow$

$$
\begin{aligned}
c_n &= b_n, \\
c_k &= b_k + c_{k+1}x_0, \; k = n-1, n-2, \cdots, 2, 1
\end{aligned}
$$

- And $Q(x_0) = c_1 = P'(x_0)$

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

OUTPUT values of $P(x_0)$ and $P'(x_0)$.

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

Step 2 For $j = n-1, n-2, \cdots, 1$,set

$$y = a_j + y * x_0; \text{(compute } b_j \text{for } P)$$

$$z = y + z * x_0; \text{(compute } c_{j-1} \text{for } Q)$$

Step 3 Set $y = a_0 + y * x_0$,(compute b_0 for P)

Step 5 OUTPUT: $y, (y = P(x_0); z, (z = P'(x_0))$

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.

OUTPUT The root p of $P(x) = 0$ or message of failure.

Step 1 Set $i = 1$ and $p_0 = x_0$.

Step 2 while $n \leq N$, do Step 3-8

- **Step 3** Set $y = a_n$(compute b_n for P);
  $z = a_n$(compute c_{n-1} for Q);
- **Step 4** For $j = n-1, n-2, \cdots, 1$,set
  $y = a_j + y * p_0$; (compute b_j for P)
  $z = y + z * p_0$; (compute c_{j-1} for Q)
- **Step 5** Set $y = a_0 + y * p_0$, (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 $i = i + 1, p_0 = p$

Step 9 OUTPUT: (Method failed), STOP.

Remarks:

- 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

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

- Repeat this procedure, till $Q_{n-2}(x)$ which is an quadratic polynomial and can be solved by quadratic formula. we can get all approximate zeros of $P(x)$. This method is called **deflation method–压缩技术**
- Theoretically, if $P(x)$ is an nth-degree polynomial with n real zeros, the deflation method can be used to find all approximate zeros. It depends on repeated use of approximations and can lead to very inaccurate results.

- If a polynomial has complex roots, how can we get them by Newton's method?
- 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 $P(x)$, 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 $P(x)$.

- In this part, we consider another method to solve root finding problems especially for **approximating the zeros of polynomials**.
- **Present**: Müller's method is first presented by D.E.Müller in 1956, and can be thought as an extension of the Secant method.
- **Idea:** 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 through $(x_0, f(x_0)), (x_1, f(x_1))$ and $(x_2, f(x_2))$.

- 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

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

- 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

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

- 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 $P(x)$.

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

- $\Rightarrow$

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

- 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.

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

- 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.

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.

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 $i \le N$, 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 Set $h = -2f(x_2)/e; p = x_2 + h$.

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.