Chapter 2
Solutions of Equations in One Variable

Hung-Yuan Fan (范洪源)

Department of Mathematics,
National Taiwan Normal University, Taiwan

Spring 2016

Section 2.1
The Bisection Method
(二分法)

Root-Finding Problem (勘根問題)

- One of the most basic problems in numerical analysis.
- Try to find a **root** (or **solution**) p of a nonlinear equation of the form

$$f(x) = 0,$$

  given a real-valued function f, i.e. $f(p) = 0$.
- The root p is also called a **zero** (零根) of f.

Note: Three numerical methods will be discussed here:

- Bisection method
- Newton's (or *Newton-Raphson*) method
- Secant and False Position (or *Regula Falsi*) methods

The Procedure of Bisection Method

Assume that f is well-defined on the interval $[a, b]$.

- Set $\mathbf{a_1} = \mathbf{a}$ and $\mathbf{b_1} = \mathbf{b}$. Find the midpoint p_1 of $[a_1, b_1]$ by

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

- If $f(p_1) = 0$, set $\mathbf{p} = \mathbf{p_1}$ and we are done.
- If $f(p_1) \neq 0$, then we have
  - $f(p_1) \cdot f(a_1) > 0 \Rightarrow p \in (p_1, b_1)$. Set $\mathbf{a_2} = \mathbf{p_1}$ and $\mathbf{b_2} = \mathbf{b_1}$.
  - $f(p_1) \cdot f(a_1) < 0 \Rightarrow p \in (a_1, p_1)$. Set $\mathbf{a_2} = \mathbf{a_1}$ and $\mathbf{b_2} = \mathbf{p_1}$.
- Continue above process until convergence.

Illustrative Diagram of Bisection Method

Given $f \in C[a,b]$ with $f(a) \cdot f(b) < 0$.

Algorithm 2.1: Bisection

INPUT endpoints a, b; tolerance TOL; max. no. of iter. N_0.

OUTPUT an approx. sol. p.

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

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

- Step 3 Set $p = a + (b - a)/2$; $FP = f(p)$.
- Step 4 If $FP = 0$ or $(b - a)/2 < TOL$ then OUTPUT(p); **STOP**.
- Step 5 Set $i = i + 1$.
- Step 6 If $FP \cdot FA > 0$ then set $a = p$ and $FA = FP$.
  Else set $b = p$. (FA is unchanged)

Step 7 OUTPUT('Method failed after N_0 iterations') and **STOP**.

Stopping Criteria (停止準則)

In Step 4, other stopping criteria can be used. Let $\epsilon > 0$ be a given tolerance and $p_1, p_2, \ldots, p_N$ be generated by Bisection method.

(1) $|p_N - p_{N-1}| < \epsilon$,

(2) $\frac{|p_N - p_{N-1}|}{|p_N|} < \epsilon$ with $p_N \neq 0$,

(3) $|f(p_N)| < \epsilon$.

Note: The stopping criterion (2) is preferred in practice.

An Example for Bisection Method

Example 1, p. 50

(1) Show that the equation

$$f(x) = x^3 + 4x^2 - 10 = 0$$

has exactly one root in $[1, 2]$.

(2) Use Bisection method to determine an approx. root which is accurate to *at least* within 10^{-4}.

- The root is $\mathbf{p} = \mathbf{1.365230013}$ correct to 9 decimal places.

Solution

(1) By IVT with $f(1)f(2) = (-5)(14) < 0$, $\exists\ p \in (1,2)$ s.t. $f(p) = 0$. Since $f'(x) = 3x^2 + 8x > 0$ for $x \in (1,2)$, the root must be unique in $[1,2]$.

(2) After **13** iterations, since $|a_{14}| < |p|$, we have

$$\frac{|p - p_{13}|}{|p|} \le \frac{|b_{14} - a_{14}|}{|a_{14}|} \le \mathbf{9.0 \times 10^{-5}}.$$

Note that

$$|f(p_9)| < |f(p_{13})|$$

in the Table 2.1.

Numerical Results for Example 1

Thm 2.1 (二分法的收斂定理)

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

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

The rate of convergence is $\mathbf{O}(\frac{\mathbf{1}}{\mathbf{2^n}})$.

pf: For each $n \geq 1$, $p \in (a_n, b_n)$ and

$$b_n - a_n = \frac{b-a}{\mathbf{2^{n-1}}} \quad \text{by induction.}$$

Hence, we hve

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

Is the error bound tight?

Remark

Applying Thm 2.1 to Example 1, we see that

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

but the actual absolute value is $|p - p_9| \approx 4.4 \times \mathbf{10^{-6}}$. In this case, the error bound in Thm 2.1 is **much larger** than the actual error.

Example 2, p. 52

As in Example 1, use Thm 2.1 to estimate the smallest number N of iterations so that $|p - p_N| < 10^{-3}$.

Sol: Applying Thm 2.1, it follows that

$$|p - p_N| \leq \frac{2-1}{2^N} < 10^{-3} \iff 2^{-N} < 10^{-3},$$

or, equivalently, $(-N)\log_{10} 2 < -3 \iff N > \frac{3}{\log_{10} 2} \approx \mathbf{9.96}$. So, $\mathbf{10}$ iterations will ensure the required accuracy. But, in fact, we know that

$$|p - p_9| \approx 4.4 \times \mathbf{10^{-6}}.$$

Useful Suggestions for the Bisection Method

In Practical Computation...

- To avoid the round-off errors in the computations, use

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

- To avoid the overflow or underflow of $f(p_n) \cdot f(a_n)$, use

$$\text{sign}(f(p_n)) \cdot \text{sign}(f(a_n)) < 0 \quad \text{instead of} \quad f(p_n) \cdot f(a_n) < 0.$$

Note: The sign function is defined by

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

Section 2.2
Fixed-Point Iteration
(固定點迭代)

Def 2.2

The number p is called a fixed point (固定點) of a real-valued function g if $g(p) = p$.

Note: A **root-finding problem** of the form

$$f(p) = 0,$$

where p is a root of f, can be transformed to a **fixed-point form**

$$p = g(p),$$

for some *suitable* function g obtained by algebraic transposition. (函數 $g(x)$ 經由原函數 $f(x)$ 代數移項可得)

Thm 2.3 (固定點的存在性與唯一性)

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

(ii) If, in addition, $g'(x)$ exists on (a,b) and $\exists\ 0 < k < 1$ s.t.

$$|g'(x)| \leq k \quad \forall\ x \in (a,b),$$

then there is **exactly one** fixed point in $[a,b]$.

Geometrically, a fixed point $p \in [a, b]$ is just the point where the curves $y = g(x)$ and $y = x$ intersect.

Proof of Thm 2.3

(i) If $g(a) = a$ or $g(b) = b$, we are done. If not, then $g(a) > a$ and $g(b) < b$, since $g([a, b]) \subseteq [a, b]$. Note that the function $h(x) = g(x) - x \in C[a, b]$ and

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

By IVT, $\exists\ p \in (a, b)$ s.t. $h(p) = 0$ or $g(p) = p$.

(ii) Suppose that $\exists\ p \neq q \in [a, b]$ s.t. $g(p) = p$ and $g(q) = q$. By MVT, $\exists\ \xi$ between p and q s.t.

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

which is a contradiction! Hence, g must have a **unique** fixed point in $[a, b]$.

Example 3, p. 59

Although the sufficient conditions are NOT satisfied for $g(x) = 3^{-x}$ on the interval $[0, 1]$, there *does* exist a **unique** fixed point of g in $[0, 1]$.

Sol: Since $g'(x) = -3^{-x} \ln 3 < 0 \quad \forall\, x \in [0, 1]$, g is strictly decreasing on $[0, 1]$ and hence

$$\frac{1}{3} = g(1) \leq g(x) \leq g(0) = 1 \quad \forall\, x \in [0, 1],$$

i.e. $g \in C[0, 1]$ and $g([0, 1]) \subseteq [0, 1]$.

Example 3: Condition (ii) Is NOT Satisfied (2/2)

But also note that

$$g'(0) = -\ln 3 \approx -1.0986,$$

thus $|g'(x)| \not< 1$ on $(0,1)$ and condition (ii) of Thm 2.3 is not satisfied. Because g is strictly deceasing on $[0,1]$, its graph must intersect the graph of $y = x$ at **exactly one** fixed point $p \in (0,1)$.

Fixed-Point Iteration (固定點迭代)

Functional (or Fixed-Point) Iteration

Assume that $g \in C[a, b]$ and $g([a, b]) \subseteq [a, b]$. The fixed-point iteration generates a sequence $\{p_n\}_{n=1}^{\infty}$, with $p_0 \in [a, b]$, defined by

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

This method is also called the **functional iteration**. (泛函迭代)

Starting wirh $p_0 \in [a, b]$, we obtain a sequence of points

$$(p_0, p_1) \to (p_1, p_1) \to (p_1, p_2) \to (p_2, p_2) \to (p_2, p_3) \to \cdots (p, p),$$

where $p = g(p)$.

To find a sol. p to $x = g(x)$ given an initial approx. p_0.

Algorithm 2.2: Fixed-Point Iteration

INPUT initial approx. p_0; tolerance TOL; max. no. of iter. N_0.

OUTPUT approx. sol. p to $x = g(x)$.

Step 1 Set $i = 1$.

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

- Step 3 Set $p = g(p_0)$.
- Step 4 If $|p - p_0| < TOL$ then OUTPUT(p); **STOP**.
- Step 5 Set $i = i + 1$.
- Step 6 Set $p_0 = p$. (Update p_0)

Step 7 OUTPUT('Method failed after N_0 iterations'); **STOP**.

5 Possible Fixed-Point Forms

The root-finding problem

$$f(x) = x^3 + 4x^2 - 10 = 0$$

can be transformed to the following 5 fixed-point forms:

(a) $x = g_1(x) = x - x^3 - 4x^2 + 10$ (b) $x = g_2(x) = (\frac{10}{x} - 4x)^{1/2}$

(c) $x = g_3(x) = \frac{1}{2}(10 - x^3)^{1/2}$ (d) $x = g_4(x) = (\frac{10}{4 + x})^{1/2}$

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

Numerical Results with $p_0 = 1.5$

Some Questions

- Under what conditions does the fixed-point iteration (FPI)

$$p_n = g(p_{n-1}), \quad n = 1, 2, \ldots$$

  converge for any $p_0 \in [a, b]$?
- What is the error bound for the FPI?
- In addition, what is the rate of convergence?

Thm 2.4 (Fixed-Point Thm)

Suppose that $g \in C[a,b]$ and $g([a,b]) \subseteq [a,b]$. If $g'(x)$ exists on (a,b) and $\exists\ k \in (0,1)$ s.t.

$$|g'(x)| \leq k \quad \forall\, x \in (a,b),$$

then for any $p_0 \in [a,b]$, the sequence $\{p_n\}_{n=1}^{\infty}$ defined by

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

converges to the **unique** fixed point $p \in [a,b]$ of g.

- Thm 2.3 ensure that $\exists!\, p \in [a, b]$ s.t. $g(p) = p$.
- For each $n \geq 1$, it follows from MVT that $\exists\ \xi_n$ between p_{n-1} and p s.t.

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

- By induction $\Longrightarrow$ $|p_n - p| \leq k^n|p_0 - p|$ for $n \geq 0$.
- Since $0 < k < 1$, we see that

$$\lim_{n\to\infty} |p_n - p| = 0 \Longleftrightarrow \lim_{n\to\infty} p_n = p$$

by the Sandwich Thm.

[Q]: What is the order of convergence for the FPI?

Cor 2.5 (固定點迭代的誤差上界)

If g satisfies the hypotheses of Thm 2.4, then we have

(1) $|p_n - p| \le k^n \max\{p_0 - a, b - p_0\} \quad \forall\, n \ge 0$,

(2) $|p_n - p| \le \dfrac{k^n}{1-k}|p_1 - p_0| \quad \forall\, n \ge 1$.

pf: Inequality (1) followss immediately from the proof of Thm 2.4. For $m > n \ge 1$, by MVT inductively, we obtain

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

Hence, by taking $m \to \infty$, we have

$$|p - p_n| = \lim_{m\to\infty} |p_m - p_n| \le k^n\left(\sum_{i=0}^{\infty} k^i\right)|p_1 - p_0| = \frac{k^n}{1-k}|p_1 - p_0|.$$

Remarks

- The rate of convergence for the fixed-point iteration depends on k^n or $\frac{k^n}{1-k}$.
- The smaller the value of k, the faster the convergence.
- The convergence would be **very slow** if $k \approx 1$.

5 Possible Fixed-Point Forms

The root-finding problem

$$f(x) = x^3 + 4x^2 - 10 = 0$$

can be transformed to the following 5 fixed-point forms:

(a) $x = g_1(x) = x - x^3 - 4x^2 + 10$ (b) $x = g_2(x) = (\frac{10}{x} - 4x)^{1/2}$

(c) $x = g_3(x) = \frac{1}{2}(10 - x^3)^{1/2}$ (d) $x = g_4(x) = (\frac{10}{4+x})^{1/2}$

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

(a) $g_1([1,2]) \not\subseteq [1,2]$ and $|g_1'(x)| > 1$ for $x \in [1,2]$.

(b) $g_2([1,2]) \not\subseteq [1,2]$ and $|g_2'(x)| \not< 1$ for any interval containing $p \approx 1.36523$, since $|\mathbf{g_2'(p)}| \approx \mathbf{3.4}$.

(c) Since $p_0 = 1.5$, $1 < 1.28 \approx g_3(1.5) \leq g_3(x) \leq g_3(1) = 1.5$ and hence $\mathbf{g_3([1, 1.5]) \subseteq [1, 1.5]}$. We also note that g_3' satisfies $|g_3'(x)| \leq |g_3'(1.5)| \approx \mathbf{0.66}$ for $x \in [1, 1.5]$.

(d) $g_4([1,2]) \subseteq [1,2]$ and the derivative g_4' satisfies

$$|g_4'(x)| = \left| \frac{-5}{\sqrt{10}(4+x)^{3/2}} \right| < \frac{5}{\sqrt{10}(5)^{3/2}} \approx \mathbf{0.1414}.$$

(e) It is **Newton's method** satisfying $\mathbf{g_5'(p) = 0}$ theoretically!

Section 2.3
Newton's Method and Its Extensions
(牛頓法及其推廣)

Derivation of Newton's Method

- Suppose that $f(p) = 0$, $f'(p) \neq 0$ and $f \in C^2[a, b]$.
- Given an initial approximation $p_0 \in [a, b]$ with $f'(p_0) \neq 0$ s.t. $|p - p_0|$ **is sufficiently small**.
- By Taylor's Thm, $\exists\ \xi(p)$ between p and p_0 s.t.

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

- Since $|p - p_0|$ is sufficiently small, it follows that

$$0 \approx f(p_0) + f'(p_0)(p - p_0) \Longleftrightarrow p \approx p_0 - \frac{f(p_0)}{f'(p_0)}.$$

- This suggests the procedure of Newton's method:

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

Newton's Method v.s. Functional Iteration

Observations

Let g be a real-valued function defined by

$$g(x) = x - \frac{f(x)}{f'(x)}, \quad x \in [a, b],$$

- Newton's method can be viewed as a fixed-point iteration $p_n = g(p_{n-1}) \quad \forall\, n \geq 1$, where $|\mathbf{p_0} - \mathbf{p}|$ **is sufficiently small**.
- If $f(p) = 0$, $g(p) = p$, i.e., p is a **fixed-point** of g.
- $g \in C[a, b]$ and its first derivative is given by

$$g'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}, \quad x \in [a, b].$$

- If $f(p) = 0$, then $g'(p) = 0$ follows immediately.

Further Questions

- Under what conditions does Newton's method converge to p?
- What is the error bond for Newton's method?
- How to choose a **good** initial guess p_0?
- What is the rate of convergence for Newton's method?

To find a sol. to $f(x) = 0$ given an initial approx. p_0.

Algorithm 2.3: Newton's Method

INPUT initial approx. p_0; tolerance TOL; max. no. of iter. N_0.

OUTPUT approx. sol. p to $f(x) = 0$.

Step 1 Set $i = 1$.

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

- Step 3 Set $p = p_0 - f(p_0)/f'(p_0)$.
- Step 4 If $|p - p_0| < TOL$ then OUTPUT(p); **STOP**.
- Step 5 Set $i = i + 1$.
- Step 6 Set $p_0 = p$. (Update p_0)

Step 7 OUTPUT('Method failed after N_0 iterations'); **STOP**.

Example 1, p. 69

Use (a) fixed-point iteration and (b) Newton's method to find an approximate root p of the nonlinear equation

$$f(x) = \cos x - x = 0$$

with initial guess $p_0 = \frac{\pi}{4}$. The root is $p \approx \mathbf{0.739085133215161}$.

(a) Consider the fixed-point form $x = g(x)$, where

$$g(x) = \cos(x) \quad \forall\, x \in [0, \frac{\pi}{2}].$$

Then it is easily seen that

1. $g \in C[0, \frac{\pi}{2}]$,
2. $g([0, \frac{\pi}{2}]) \subseteq [0, 1] \subseteq [0, \frac{\pi}{2}]$,
3. $|g'(x)| = |-\sin(x)| < 1 \quad \forall\, x \in (0, \frac{\pi}{2})$.

From Thm 2.4 $\Longrightarrow$ the fixed-point iteration

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

must converge to the **unique** fixed point $p \in (0, \frac{\pi}{2})$ of g for any initial $p_0 \in [0, \frac{\pi}{2}]$!

Applying the FPI with an initial guess $p_0 = \frac{\pi}{4}$, we obtain the following numerical results.

The root is $p \approx \mathbf{0.739085133215161}$.
Note that **only 2 significant digits!**

(b) For the same initial approx. $p_0 = \pi/4$, applying Newton's method

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

The actual root is $p \approx \mathbf{0.739085133215161}$.
we obtain the following numerical results

Thm 2.6 (牛頓法的收斂定理)

Let $f \in C^2[a, b]$ and $p \in (a, b)$. If $f(p) = 0$ and $f'(p) \neq 0$, then $\exists\ \delta > 0$ s.t. Newton's method generates a sequence $\{p_n\}_{n=1}^{\infty}$ defined by

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

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

Note: The **local convergence** of Newton's method is guaranteed in Thm 2.6, but the order of convergence is NOT discussed here!

Sketch of Proof (1/2)

- Since $f'(p) \neq 0$, $\exists\ \delta_1 > 0$ s.t.

$$f'(x) \neq 0 \quad \forall\, x \in (p - \delta_1, p + \delta_1),$$

and hence

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

is well-defined on $(p - \delta_1, p + \delta_1)$.

- Moreover, since its derivative is given by

$$g'(x) = \frac{f(x) f''(x)}{[f'(x)]^2} \quad \forall\, x \in (p - \delta_1, p + \delta_1),$$

it follows that $g \in C^1(p - \delta_1, p + \delta_1)$ because $f \in C^2[a, b]$.

- Note that $f(p) = 0 \Longrightarrow g(p) = p$ and $g'(p) = 0$.

Sketch of Proof (2/2)

- Because g' is conti. at p, for any $k \in (0,1)$, $\exists\ 0 < \delta < \delta_1$ s.t.

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

- For $x \in [p-\delta, p+\delta]$, from MVT $\Rightarrow \exists\, \xi$ between x and p s.t.

$$|g(x)-p| = |g(x)-g(p)| = |g'(\xi)|\,|x-p| < \delta.$$

  Hence, $g([p-\delta, p+\delta]) \subseteq [p-\delta, p+\delta]$.

- From Thm 2.4 $\Longrightarrow$ the seq. generated by Newton's method

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

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

Questions

- How to guess a good initial approximation p_0?
- How to estimate $\delta > 0$ derived in Thm 2.6?
- What is the order of convergence for Newton's method?
- How to modify Newton's method if $f'(x)$ is difficult to be evaluated in practice? Use **Secant Method!**

Derivation of Secant Method (割線法)

- In many applications, it is often difficult to evaluate the derivative of f.
- Since $f'(p_{n-1}) = \lim_{x \to p_{n-1}} \frac{f(x)-f(p_{n-1})}{x-p_{n-1}}$, we have

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

  for any $n \geq 2$.
- With above spprox. for the derivative, Neton's method is rewritten as

$$p_n = p_{n-1} - \frac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})} \quad \forall\, n \geq 2.$$

  This is called the **Secant method** with initial approximations p_0 and p_1.

Illustrative Diagram for Secant Method

Key Steps of Secant Method

Given two initial p_0 and p_1 with $q_0 \leftarrow f(p_0)$ and $q_1 \leftarrow f(p_1)$, the followings are performed **repeatedly** in the Secant method:

1. Compute the new approximation

$$p \leftarrow p_1 - \frac{q_1(p_1 - p_0)}{q_1 - q_0};$$

2. Update $p_0 \leftarrow p_1$ and $q_0 \leftarrow q_1$; $p_1 \leftarrow p$ and $q_1 \leftarrow f(p)$.

To find a sol. to $f(x) = 0$ given initial approx. p_0 and p_1.

Algorithm 2.4: Secant Method

INPUT initial approx. p_0, p_1; tolerance TOL; max. no. of iter. N_0.

OUTPUT approx. sol. p to $f(x) = 0$.

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

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

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

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

Step 5 Set $i = i + 1$.

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

Step 7. OUTPUT('Method failed after N_0 iterations'); **STOP**.

Example 2, p. 72

Use the Secant method to find a sol. to

$$f(x) = \cos x - x = 0$$

with initial approx. $p_0 = 0.5$ and $p_1 = \pi/4$. Compare the results with those of Newton's method obtained in Example 1.

Sol: Applying the Secant method

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

we see that its approximation p_5 is accurate to **10** significant digits, whereas Newton's method produced the same accuracy after **3** iterations.

The Secant method is much faster than fixed-point iteration, but slower than Newton's method.

- The method of False Position is also called *Regula Falsi* method. **The root is always bracketed between successive approximations**.
- Firstly, find p_2 using the **Secant method** . How to determine the next approx. p_3?
  - If $f(p_2) \cdot f(p_1) < 0$ (or $\mathbf{sign(f(p_2)) \cdot sign(f(p_1)) < 0}$), then p_3 is the x-intercept of the line joining $\mathbf{(p_1, f(p_1))}$ and $\mathbf{(p_2, f(p_2))}$.
  - If not, p_3 is the x-intercept of the line joining $\mathbf{(p_0, f(p_0))}$ and $\mathbf{(p_2, f(p_2))}$, and then **interchange the indices on $\mathbf{p_0}$ and $\mathbf{p_1}$**.
- Continue above procedure until convergence.

Secant Method v.s. Method of False Position

Key Steps of False Position Method

Given two initial p_0 and p_1 with $q_0 \leftarrow f(p_0)$ and $q_1 \leftarrow f(p_1)$, the followings are performed **repeatedly** in the False Position method:

1. Compute the new approximation
$$p \leftarrow p_1 - \frac{q_1(p_1 - p_0)}{q_1 - q_0};$$
2. Compute $q \leftarrow f(p)$;
3. If $q \cdot q_1 < 0$, update $p_0 \leftarrow p_1$ and $q_0 \leftarrow q_1$;
4. Update $p_1 \leftarrow p$ and $q_1 \leftarrow q$.

To find a sol. to $f(x) = 0$ given initial approx. p_0 and p_1.

Algorithm 2.5: Method of False Position

INPUT initial approx. p_0, p_1; tolerance TOL; max. no. of iter. N_0.

OUTPUT approx. sol. p to $f(x) = 0$.

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

Step 2 While $i \leq N_0$ do **Steps 3–7**

- Step 3 Set $p = p_1 - q_1(p_1 - p_0)/(q_1 - q_0)$.
- Step 4 If $|p - p_1| < TOL$ then OUTPUT(p); **STOP**.
- Step 5 Set $i = i + 1$; $q = f(p)$.
- Step 6 If $q \cdot q_1 < 0$ then $p_0 = p_1$; $q_0 = q_1$.
- Step 7 Set $p_1 = p$; $q_1 = q$.

Step 8 OUTPUT('Method failed after N_0 iterations'); **STOP**.

Example 3, p. 74

Use the method of False Position to find a sol. to

$$f(x) = \cos x - x = 0$$

with $p_0 = 0.5$ and $p_1 = \pi/4$. Compare the results with those obtained by Newton's method and Secant method.

Section 2.4
Error Analysis for Iterative Methods

Def 2.7 (收斂階數的定義)

A sequence $\{p_n\}_{n=0}^{\infty}$ converges to p of order α, with asymptotic error constant λ if $\exists\ \alpha \geq 1$ and $\lambda \geq 0$ with

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

(i) $\alpha = 1$ and $0 < \lambda < 1 \Longrightarrow \{p_n\}_{n=0}^{\infty}$ is **linearly convergent**.

(ii) $\alpha = 1$ and $\lambda = 0 \Longrightarrow \{p_n\}_{n=0}^{\infty}$ is **superlinearly convergent**.

(iii) $\alpha = 2 \Longrightarrow \{p_n\}_{n=0}^{\infty}$ is **quadratically convergent**.

Note: The higher-order convergence is always expected in practical computation!

Thm 2.8 (固定點迭代的線性收斂性)

Suppose that $g \in C[a,b]$ and $g([a,b]) \subseteq [a,b]$. If $g' \in C(a,b)$, $\exists\, k \in (0,1)$ s.t. $|g'(x)| \le k \quad \forall\, x \in (a,b)$ and $g'(p) \neq 0$, then for any $p_0 \in [a,b]$, the sequence

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

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

Proof of Thm 2.8

- Thm 2.4 (Fixed-Point Thm) ensures that the sequence p_n converges to the unique fixed point $p \in [a, b]$.
- For each $n \geq 1$, by MVT $\Longrightarrow \exists\, \xi_n$ between p_n and p s.t.

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

- Since $\lim_{n\to\infty} p_n = p$, $\xi_n \to p$ as $n \to \infty$. Thus,

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|} = \lim_{n\to\infty} |g'(\xi_n)| = |g'(p)| > 0$$

because $g' \in C(a, b)$, i.e., the sequence p_n converges to p only linearly!

Thm 2.9 (固定點迭代的二次收斂性)

If $g(p) = p$, $g'(p) = 0$ and $\exists$ open interval I containing p where

$$g'' \in C(I) \quad \text{and} \quad |g''(x)| < M \quad \forall\, x \in I,$$

then $\exists\, \delta > 0$ s.t. the sequence defined by

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

converges **at least quadratically** to p for any $p_0 \in [p - \delta, p + \delta]$. Moreover, we have

$$|p_{n+1} - p| < \frac{M}{2} |p_n - p|^2 \quad \text{for sufficiently large values of } n.$$

- For any $k \in (0,1)$, since $\lim_{x \to p} g'(x) = g'(p) = 0$, $\exists\ \delta > 0$ s.t.

$$|g'(x)| \leq k < 1 \quad \forall\ x \in [p-\delta, p+\delta] \subseteq I. \qquad (1)$$

- From (1) and MVT $\Rightarrow$ $|p_n - p| < \delta \quad \forall\ n \in \mathbb{N}$ if $|p_0 - p| < \delta$, and $g([p-\delta, p+\delta]) \subseteq [p-\delta, p+\delta]$. Hence, $\lim_{n \to \infty} p_n = p$ by Fixed-Point Thm (Thm 2.4).
- For $n \geq 1$, from Taylor's Thm $\Rightarrow \exists\ \xi_n$ between p_n and p s.t.

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

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

Corollary (牛頓法的二次收斂性)

Let $f(p) = 0$, $f'(p) \neq 0$ and define the real-valued function

$$g(x) = x - \frac{f(x)}{f'(x)} \quad \forall\ x \in I,$$

where I is an open interval containing p. If $g'' \in C(I)$ with $|g''(x)| < M \quad \forall x \in I$, then $\exists\ \delta > 0$ s.t. Newton's method generates a sequence $\{p_n\}_{n=0}^{\infty}$ converging **at least quadratically** to p for any $p_0 \in [p - \delta, p + \delta]$. The asymptotic error constant is $\lambda = \frac{|g''(p)|}{2}$.

(當初始值 p_0 充分接近零根 p, 牛頓法是一個二次收斂的算法!)

Def 2.10 (零根的重數)

A number p is called a zero of multiplicity $m \in \mathbb{N}$ (重數) of f if for any $x \neq p$, we can write

$$f(x) = (x-p)^m q(x) \quad \text{with} \quad \lim_{x \to p} q(x) \neq 0.$$

Note: A root p of f is called **simple** (單根) if it is a zero of multiplicity one, i.e., $m = 1$.

Thm 2.11 (單根的充分必要條件)

$f \in C^1[a,b]$ has a **simple zero** at $p \in (a,b) \iff f(p) = 0$, but $f'(p) \neq 0$.

Proof (1/2)

$(\Longrightarrow)$ If f has a simple zero at p, then $f(x) = (x-p)q(x)$ for $x \in [a,b] \backslash \{p\}$. So, $f(p) = 0$ and we also see that

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

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

Proof (2/2)

($\Longleftarrow$) Suppose that $f \in C^1[a,b]$ with $f(p) = 0$ and $f'(p) \neq 0$ for some $p \in (a,b)$.

- From Taylor's Thm $\Longrightarrow$ for any $x \in [a,b] \backslash \{p\}$, $\exists\, \xi(x)$ between x and p s.t.

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

  where $q(x) \equiv (f' \circ \xi)(x) = f'(\xi(x))$ for $x \in [a,b] \backslash \{p\}$.

- Because $\xi(x) \to p$ as $x \to p$ and f' is continuous at p,

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

Hence, f must have a simple zero at p by Def.

Thm 2.12 (重根的充分必要條件)

$f \in C^m[a, b]$ has a zero of multiplicity m at $p \in (a, b)$
$\Longleftrightarrow f(p) = f'(p) = f''(p) = \cdots f^{(m-1)}(p) = 0$, but $f^{(m)}(p) \neq 0$.

Homework: do **Exercise 12** for the proof.

Note: In practice, Newton's method usually converges **linearly** to a zero p of multiplicity $m \geq 2$, even though an initial guess p_0 is chosen close to p.

Example 1, p. 83

Consider $f(x) = e^x - x - 1 = 0$ for all $x \in \mathbb{R}$.

(a) Show that f has a zrero of multiplicity $\mathbf{2}$ at $p = 0$.

(b) Use Newton's method to compute an approximation to p with $\mathbf{p_0 = 1}$.

Sol:

(a) Since

$$f(0) = e^0 - 0 - 1 = 0, f'(0) = e^0 - 1 = 0, f''(0) = e^0 = 1 \neq 0,$$

it follows that $p = 0$ is a zero of multiplicity 2 by Thm 2.12.

(b) The following table shows the **linear convergence** of Newton's method when a multiple zero occurs:

Suppose that $f \in C^m[a,b]$ and consider $\mu(x) = \frac{f(x)}{f'(x)}$.

- If f has a zero of multiplicity $m(\geq 2)$ at p, then $f(x) = (x-p)^m q(x)$ and hence

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

- Since $q(p) \neq 0$, $\mu(\mathbf{p}) = \mathbf{0}$ and $\lim_{x\to p} \hat{q}(x) = \frac{\mathbf{1}}{\mathbf{m}} \neq \mathbf{0}$. So, $\mu(x)$ has a **simple zero** at p.

Improvement of Convergence (2/2)

- Applying Newton's method for solving the problem $\mu(x) = 0$, we obtain

$$g(x) = x - \frac{\mu(x)}{\mu'(x)} = x - \frac{f(x)f'(x)}{[f'(x)]^2 - f(x)f''(x)}. \quad \text{(Check!)}$$

- **Modified Newton's Method:** (修正的牛頓法)

$$p_n = p_{n-1} - \frac{f(p_{n-1})f'(p_{n-1})}{[f'(p_{n-1})]^2 - f(p_{n-1})f''(p_{n-1})}$$

for all $n \geq 1$.

Example 2, p. 84

Use the **Modified Newton's method** for solving the multiple root $x = 0$ of $f(x) = e^x - x - 1$ with $p_0 = 1$.

- A system with **10** digits of precision is used in this case.
- Both the numerator and the denominator approach $0 \Rightarrow$ **Loss of Significant Digits!**

Exercise 14, p. 86 (補充題)

It is shown from pp. 228–229 of [DaB] that if $f(p) = 0$ and the sequence $\{p_n\}_{n=0}^{\infty}$ generated by **Secant Method** converges to p, then $\exists\ C > 0$ s.t.

$$|p_{n+1} - p| \approx C|p_n - p||p_{n-1} - p|$$

for sufficiently large values of n. Apply this fact to prove the order of convergence for Secant Method is

$$\alpha = \frac{1+\sqrt{5}}{2} \approx \mathbf{1.62}. \quad \text{(golden ratio; 黃金比率)}$$

Let $\mathbf{e_n} = \mathbf{p_n} - \mathbf{p}$ for $n \geq 0$. If $p_n \to p$ of order α, then $\exists\ \lambda > 0$ s.t.

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \lambda > 0.$$

Then for sufficiently large values of n, $|e_{n+1}| \approx \lambda |e_n|^\alpha$. Thus,

$$|e_n| \approx \lambda |e_{n-1}|^\alpha \quad \text{or} \quad |e_{n-1}| \approx \lambda^{-1/\alpha} |e_n|^{1/\alpha}.$$

Using the hypothesis gives

$$\lambda |e_n|^\alpha \approx |e_{n+1}| \approx C|e_n| \cdot |e_{n-1}| \approx C\lambda^{-1/\alpha} |e_n|^{1+1/\alpha}$$

for sufficiently large values of n. So, we further have $|e_n|^\alpha \approx C\lambda^{-1/\alpha - 1} |e_n|^{1+1/\alpha}$. Since the powers of $|e_n|$ must agree,

$$\alpha = 1 + 1/\alpha \quad \text{or} \quad \alpha = \frac{1+\sqrt{5}}{2} \approx 1.62.$$

Remarks

- From **Exercise 14(a) of Section 2.5**, p. 91, we see that if a sequence $p_n \to p$ of order α for $\alpha > 1$, then it is superlinearly convergent.
- Thus, the Secant method must be superlinearly convergent!

Section 2.5
Accelerating Convergence
(加速收斂性)

Objective

We try to develop some accelerating techniques for a linearly convergent sequence $\{p_n\}_{n=0}^{\infty}$ generated by the fixed-point iteration.

1. Aitken's Δ^2 method (**more rapid convergence**)
2. Steffensen's method (**quadratic convergence**)

Def 2.13 (向前差分算子)

Let $\{p_n\}_{n=}^{\infty}$ be a sequence generated by some iterative method.

- The **forward difference** operator Δ is defined by

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

- Higher powers of Δ is defined recursively by

$$\Delta^k p_n = \Delta(\Delta^{k-1} p_n) \quad \forall\, k \geq 2.$$

Note: For $k = 2$ in above Def., we have

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

Assume that $p_n \to p$ and the signs of $p_{n+2} - p$, $p_{n+1} - p$, $p_n - p$ are the same.

- Moreover, suppose also that

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

  if n **is sufficiently large**.

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

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

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

$$\iff p \approx \frac{p_{n+2}p_n - p_{n+1}^2}{p_{n+2} - 2p_{n+1} + p_n}$$
$$= \frac{(p_n p_{n+2} - \mathbf{2p_n p_{n+1}} + \mathbf{p_n^2}) - (p_{n+1}^2 - \mathbf{2p_{n+1}p_n} + \mathbf{p_n^2})}{p_{n+2} - 2p_{n+1} + p_n}$$
$$\iff p \approx p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n} = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n} \text{ for } n \geq 0.$$

- **Aitken's Δ^2 Method:**

$$\hat{p}_n = p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n} \equiv \{\Delta^2\}(p_n) \quad \forall\, n \geq 0,$$

where the term $p_n = g(p_{n-1})$ is often generated by the **fixed-point iteration** for $n \geq 1$.

Thm 2.14 (Aitken 序列的收斂定理)

Suppose that $\{p_n\}_{n=0}^{\infty}$ is a sequence converging **linearly** to the limit p with

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

Then the Aitken's Δ^2 sequence $\{\hat{p}_n\}_{n=0}^{\infty}$ converges to p faster than $\{p_n\}_{n=0}^{\infty}$ in the sense that

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

Note: See Exercise 16 for the proof of this theorem.

Steffensen's Sequences

- Aitken's Δ^2 method constructs the terms in order:

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

- Steffensen's method constructs the same first 4 terms and every **third term** of the Steffensen sequence is generated by the Aitken's Δ^2 operator, i.e.

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

To find a sol. to $p = g(p)$ given initial approx. p_0.

Algorithm 2.6: Steffensen's Method

INPUT initial approx. p_0; tolerance TOL; max. no. of iter. N_0.

OUTPUT approx. sol. p to $x = g(x)$.

Step 1 Set $i = 1$.

Step 2 While $i \leq N_0$ do **Steps 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 $p_0 = p$. (Update p_0)

Step 7 OUTPUT('Method failed after N_0 iterations'); **STOP**.

Example

Use the Steffensen's method to accelerate the fixed-point iteration $p_n = g(p_{n-1}), n \geq 1$, where

$$g(x) = g_4(x) = (\frac{10}{4+x})^{1/2},$$

for solving $f(x) = x^3 + 4x^2 - 10 = 0$ with $p_0 = 1.5$.

Sol: The **quadratic convergence** of Steffensen's method is shown. The computed sol. is accurate to the**9th** decimal place as Newton's method.

Thm 2.15 (Steffensen 序列的二次收斂性)

Suppose that $x = g(x)$ has the solution p with $g'(p) \neq 1$. If $\exists\, \delta > 0$ s.t. $g \in C^3[p-\delta, p+\delta]$, then Steffensen's method gives **quadratic convergence** for any $p_0 \in [p-\delta, p+\delta]$.

Section 2.6
Zeros of Polynomials and Müller's Method
(多項式的零根與 Müller 法)

Thm 2.16 (Fundamental Theorem of Algebra; FTA)

Every poly. of degree $n \geq 1$ with **real or complex coefficients**

$$P(x) = a_n x_n + a_{n-1} x_{n-1} + \cdots + a_1 x_1 + a_0$$

has **exactly** n roots (or zeros) in $\mathbb{C}$.

Two Questions

Q1 For any x_0, how to evaluate $P(x_0)$ efficiently and accurately in practical computation?

Q2 How to find the complex zeros of $P(x)$ numerically?

Cor 2.17 (多項式的因式分解)

If $P(x)$ is a poly. of degree n with real or complex coeffs., then $\exists$ distinct zeros $x_1, x_2, \cdots, x_k \in \mathbb{C}$ and $m_1, m_2, \cdots, m_k \in \mathbb{N}$ s.t.

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

Here, m_i is the multiplicity of the zero x_i for $i = 1, 2, \ldots, k$.

Cor 2.18 (多項式的相等)

Let $P(x)$ and $Q(x)$ be pplys. of degree **at most** n. If $\exists$ **distinct** $x_1, \cdots, x_k \in \mathbb{C}$ with $k > n$ s.t.

$$P(x_i) = Q(x_i), \quad i = 1, 2, \ldots, k,$$

then $P(x) \equiv Q(x)$, i.e., $P(x) = Q(x) \quad \forall\, x \in \mathbb{C}$.

Thm 2.19 (Horner's Method)

Let $P(x) = a_n x_n + a_{n-1} x_{n-1} + \cdots + a_1 x_1 + a_0$ and $x_0 \in \mathbb{R}$.
Define $b_n = a_n$ and

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

We then have

(1) $b_0 = P(x_0)$ can be evaluated in a nested manner, i.e.,

$$P(x_0) = a_0 + (\cdots a_{n-3} + (a_{n-2} + (a_{n-1} + a_n x_0) x_0) x_0 \cdots) x_0.$$

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

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

- It suffices to prove Part (2), since Part (1) is easily seen from the construction of b_k for $k = n, n-1, \ldots, 1, 0$.
- For the Part (2), we see that

$$
\begin{aligned}
& (x - x_0)Q(x) + b_0 \\
=& (x - x_0)(b_n x^{n-1} + b_{n-1}x^{n-2} + \cdots b_2 x + b_1) + b_0 \\
=& b_n x^n + b_{n-1}x^{n-1} + \cdots b_2 x^2 + b_1 x \\
& - b_n x_0 x^{n-1} - b_{n-1} x_0 x^{n-2} - \cdots - b_2 x_0 x - b_1 x_0 + b_0 \\
=& b_n x^n + (b_{n-1} - b_n x_0)x^{n-1} + \cdots + (b_1 - b_2 x_0)x + (b_0 - b_1 x_0) \\
=& a_n x^n + a_{n-1}x^{m-1} + \cdots + a_1 x + a_0 \\
=& P(x), \text{ since } a_n = b_n \text{ and } a_k = b_k - b_{k+1}x_0, \quad k = n-1, \ldots, 1, 0.
\end{aligned}
$$

Therefore, we have $b_0 = P(x_0)$.

Example 2, p. 93

Use **Horner's Method** to evaluate $P(x) = 2x^4 - 3x^2 + 3x - 4$ at $x_0 = -2$. The actual value is $P(x_0) = P(-2) = 10$.

Sol: Try to construct a table as follows.

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

So, $P(x) = (x+2)(2x^3 - 4x^2 + 5x - 7) + 10$ and hence $P(-2) = b_0 = 10$ by using Horner's Metod.

- From Thm 2.19, we obtain that

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

So, differentiating w.r.t. x gives

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

- For any $x_0 \in \mathbb{R}$, we have $P'(x_0) = Q(x_0)$, which can be evaluated efficiently using Horner's Method.
- Newton's Method can be rewritten as

$$p_k = p_{k-1} - \frac{P(p_{k-1})}{Q(p_{k-1})}, \quad k = 1, 2, \ldots,$$

where p_0 is a good initial approx. to the zero p of $P(x)$.

To evaluate $P(x_0)$ and $P'(x_0)$ for an nth-degree polynomial $P(x)$.

Algorithm 2.7: Horner's Method

INPUT degree n; coeff. $a_n, \cdots, a_1, a_0$; x_0.

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

Step 1 Set $y = a_n$; $z = a_n$.

Step 2 For $j = n-1, n-2, \ldots, 1$

Set $y = y \cdot x_0 + a_j$; (Compute b_j for P.)

$z = z \cdot x_0 + \text{y}$. (Compute b_{j-1} for Q.)

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

Step 4 OUTPUT(y, z); **STOP**.

Operation Counts (運算量)

- The usual method for evaluating

$$P(x_0) = a_n x_0^n + a_{n-1} x_0^{n-1} + \cdots + a_2 x_0^2 + a_1 x_0 + a_0$$
$$= a_n \cdot (x_0 \cdot x_0^{n-1}) + \cdots + a_2 \cdot (x_0 \cdot x_0) + a_1 \cdot x_0 + a_0$$

requires $\mathbf{2n-1}$ multiplications and n additions.

- The Horner's Method for evaluating the value $P(x_0)$ requires only $\mathbf{n}$ multiplications and n additions.

$\Longrightarrow$ **This avoids the loss of significance!**

Example 3, p. 94

Find an approximation to a zero of

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

using Newton's method with $\mathbf{x_0 = -2}$ and Horner's method.

Sol: Using Horner's Method with initial $x_0 = -2$, we have

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

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

Next, evaluate $P(x_1)$ and $P'(x_1) = Q(x_1) \Longrightarrow$

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

$\Longrightarrow x_2 = x_1 - \frac{P(x_1)}{Q(x_1)} = -1.796 - \frac{1.742}{-32.565} \approx \mathbf{-1.7425}$.
Similarly, $\mathbf{x_3} \approx \mathbf{-1.73897}$ and an actual zero to **5** decimal places is $\mathbf{-1.73896}$.

Thm 2.20 (實係數多項式的複數重根)

If $z = a + bi$ is a complex zero of **multiplicity** m of the poly. $P(x)$ with **real coefficients**, then

(1) $\bar{z} = a - bi$ is also a complex zero of **multiplicity** m of $P(x)$.

(2) $P(x) = (x^2 - 2ax + a^2 + b^2)^m Q(x)$, where $Q(x)$ is some poly. with $Q(z) \neq 0$.

Basic Ideas

- **Secant Method**: given p_0 and $p_1 \Rightarrow p_2$ is the x-intercept of the line through $(p_0, f(p_0))$ and $(p_1, f(p_1))$.
- **Müller's Method**: given p_0, p_1 and $p_2 \Rightarrow p_3$ is the x-intercept of the **parabola** through $(p_0, f(p_0))$, $(p_1, f(p_1))$ and $(p_2, f(p_2))$.

Derivation of Müller's Method (1/4)

- Consider the **quadratic polynomial**

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

passing through $(p_0, f(p_0))$, $(p_1, f(p_1))$ and $(p_2, f(p_2))$.

- So, we obtain the following linear system

$$\begin{aligned} f(p_2) &= a \cdot 0 + b \cdot 0 + c, \\ f(p_0) &= a(p_0 - p_2)^2 + b(p_0 - p_2) + c, \\ f(p_1) &= a(p_1 - p_2)^2 + b(p_1 - p_2) + c \end{aligned}$$

to determine the constants a, b and c uniquely.

- It follows from Cramer's Rule that

$$c = f(p_2), \tag{2}$$

$$b = \frac{(p_0 - p_2)^2[f(p_1) - f(p_2)] - (p_1 - p_2)^2[f(p_0) - f(p_2)]}{(p_0 - p_2)(p_1 - p_2)(p_0 - p_1)}, \tag{3}$$

$$a = \frac{(p_1 - p_2)[f(p_0) - f(p_2)] - (p_0 - p_2)[f(p_1) - f(p_2)]}{(p_0 - p_2)(p_1 - p_2)(p_0 - p_1)}. \tag{4}$$

- Let h_1, h_2, δ_1 and δ_2 be defined by

$$h_1 = p_1 - -p_0, \quad h_2 = p_2 - p_1,$$
$$\delta_1 = [f(p_1) - f(p_0)]/h_1, \quad \delta_2 = [f(p_2) - f(p_1)]/h_2. \tag{5}$$

- Substituting (5) into (3)–(4) gives that

$$a = \frac{(-h_2)(-h_1\delta_1 - h_2\delta_2) - [-(h_1+h_2)](-h_2\delta_2)}{-(h_1+h_2)(-h_2)(-h_1)} = \frac{h_2h_1(\delta_2-\delta_1)}{(h_1+h_2)h_2h_1} = \frac{\delta_2-\delta_1}{h_1+h_2}.$$

and

$$b = \frac{(h_1+h_2)^2(-h_2\delta_2) - h_2^2(-h_1\delta_1 - h_2\delta_2)}{-(h_1+h_2)(-h_2)(-h_1)} = \frac{(h_1+h_2)h_1h_2\delta_2 + h_2^2h_1(\delta_2-\delta_1)}{(h_1+h_2)h_2h_1} = \delta_2 + h_2 a.$$

- If p_3 is the intersection of the x-axis with $y = P(x)$, then

$$P(p_3) = a(p_3 - p_2)^2 + b(p_3 - p_2) + c = 0$$

with $c = f(p_2)$. So, we know that

$$p_3 = p_2 + \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = p_2 + \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$
$$= p_2 + \frac{-2c}{b + \text{sign}(b)\sqrt{b^2 - 4ac}}.$$ (取分母絕對值較大者)

- Repeat above procedure with the points $(p_1, f(p_1))$, $(p_2, f(p_2))$ and $(p_3, f(p_3))$ to obtain the next approx. p_4 to a zero of the nonlinear equation $f(x) = 0$.

To find a sol. to $f(x) = 0$ given **3** approx. p_0, p_1 and p_2.

Algorithm 2.8: Müller's Method

INPUT initial p_0, p_1, p_2; tol. TOLL; max. no. iter. N_0.

OUTPUT approx. sol. p.

Step 1 Set $i = 3$.

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

Step 3 Set $h_1 = p_1 - p_0$; $\delta_1 = [f(p_1) - f(p_0)]/h_1$;
$h_2 = p_2 - p_1$; $\delta_2 = [f(p_2) - f(p_1)]/h_2$;
$a = (\delta_2 - \delta_1)/(h_1 + h_2)$; $b = \delta_2 + h_2 a$; $c = f(p_2)$;
$D = \sqrt{b^2 - 4ac}$. (**May require complex arithmetic**.)

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

Step 5 Set $h = -2c/E$; $p = p_2 + h$.

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

Step 7 Set $p_0 = p_1$; $p_1 = p_2$; $p_2 = p$; $i = i + 1$.

Step 8 OUTPUT('Method failed after N_0 iterations'); **STOP**.

An Illustrative Example, p. 98

Use Müller's Method to find all zeros of the 4th-degree polynomial

$$f(x) = x^4 - 3x^3 + x^2 + x + 1$$

with $\mathbf{TOL} = \mathbf{10^{-5}}$ and the following initial approximations:

(1) $p_0 = 0.5,\ p_1 = -0.5,\ p_3 = 0$; (Complex zero)

(2) $p_0 = 0.5,\ p_1 = 1.0,\ p_3 = 1.5$; (Real zero of **small** magnitude)

(3) $p_0 = 1.5,\ p_1 = 2.0,\ p_3 = 2.5$. (Real zero of **large** magnitude)

Numerical Results (1/2)

One complex root z_1 is computed by the Müller's Method, and the other complex root z_2 can be obtained by taking $z_2 = \bar{z}_1$ **directly**.

Two distinct real roots are computed by the Müller's Method with different initial points.

Thank you for your attention!