數值分析

Chapter 2
Solving Nonlinear Equations

授課教師：劉耀先

國立陽明交通大學 機械工程學系 EE464

yhliu@nctu.edu.tw

110 學年度第一學期

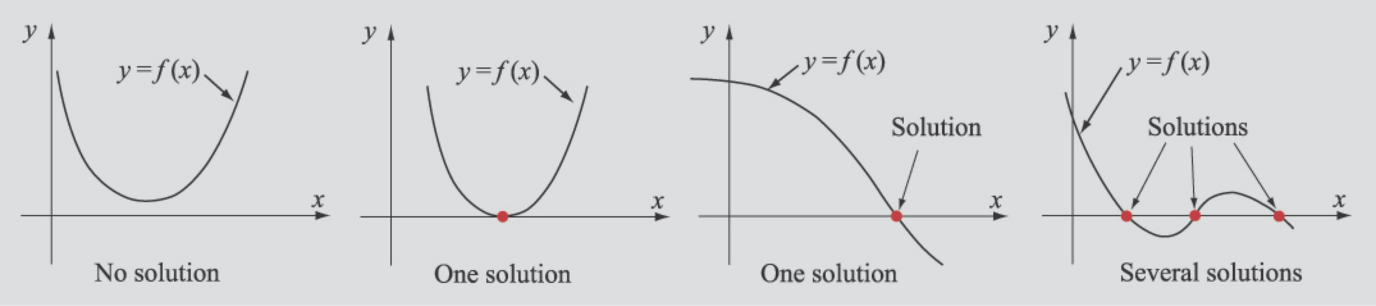
Outline

- Background
- Estimation of Errors in Numerical Solutions
- Bisection Method
- Regular Falsi Method
- Newton’s Method
- Secant Method
- Fixed Point Iteriation Method
- Use of MATLAB Built-in Functions
- Equations with Multiple Solutions
- System of Nonlinear Equations

2.1 Background

- A solution (or Root) to the equation is a numerical value of x that satisfies the equation

$$f(x) = 0 \overset{find}{\square} solution(root)$$



- When the equation is simple, the value of x can be determined analytically
- Numerically solving an equation ⇒ has to choose desired accuracy
- Approaches:

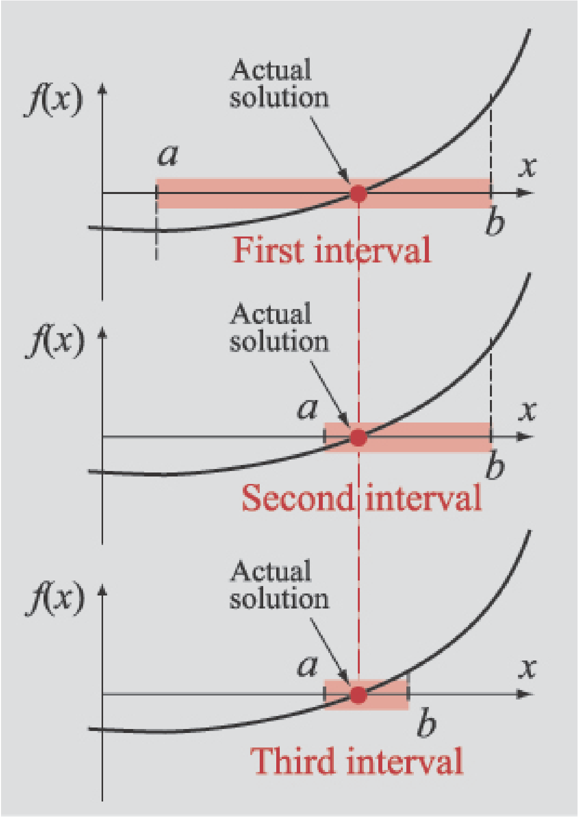
  Initial value x → an approximate solution → a more accurate solution

Overview of approaches in solving equations numerically

- A numerical solution is obtained in a process that starts by finding an approximate solution and is followed by a numerical procedure in which a better (more accurate) solution is determined.
- Evaluate f(x) at different values of x
- It starts at one value of x and then changes the value of x in small increments
- A numerical solution is obtained one root at a time

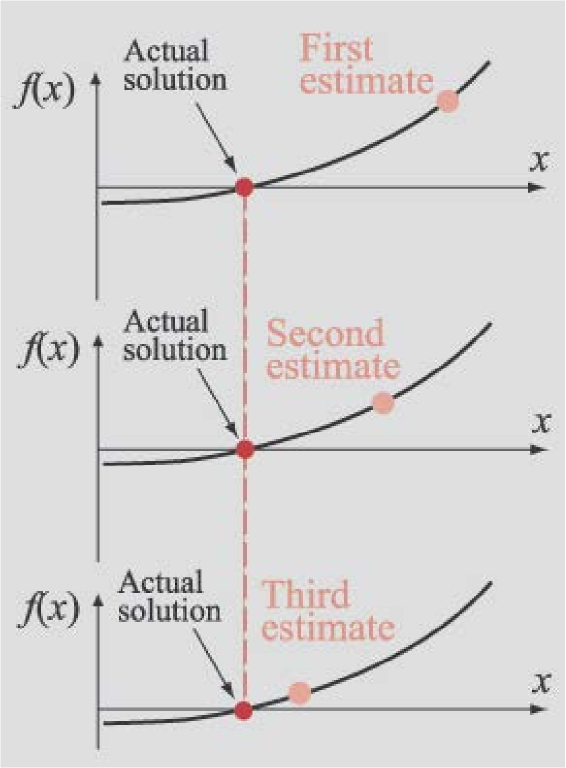
(1) Bracketing method

- Identify the interval that includes the solution
- The endpoints of the interval are the upper bound and lower bound of the solution
- Reduce the size of the interval by numerical scheme until the distance between the endpoints is less than the desired accuracy of the solution
- Always converge to the solution
- Bisection method, regula falsi method



(2) Open method

- Assume an initial (one point) for the solution
- The value of this initial guess should be close to the actual solution
- Using numerical scheme to calculate better (more accurate) values for the solution
- Usually more efficient but sometimes might not yield the solution
- Newton's method, secant method, fixed point iteration



2.2 Estimation of errors in numerical solutions

- X_{Ts}: exact(true) solution $f(X_{TS})=0$
- X_{Ns}: numerical solution $f(X_{NS})=\varepsilon$ a small number$\neq 0$

- (1) True error = $X_{TS}-X_{NS}$
- (2) Tolerance in f(x) = $\left|f(x_{TS})-f(x_{NS})\right| = |0-\varepsilon| = |\varepsilon|$

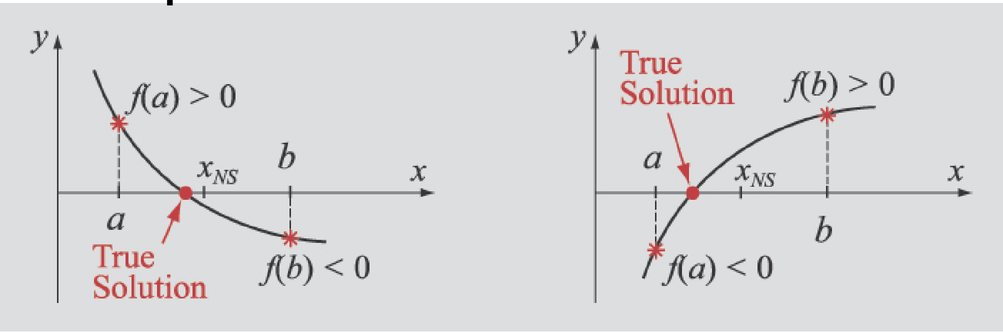
- (3) Tolerance in the solution : the maximum amount by which the true solution can deviate from an approximate numerical solution
- useful for bracketing method
- if solution is within [a,b], $x_{NS} = \frac{a+b}{2}$
- Tolerance= $\left|\frac{b-a}{2}\right|$

Estimation of errors in numerical solutions

- Relative error
- True relative error = $\left|\dfrac{x_{TS} - x_{NS}}{x_{TS}}\right|$
- Estimated Relative Error = $\left|\dfrac{x_{NS}^{(n)} - x_{NS}^{(n-1)}}{x_{NS}^{(n-1)}}\right|$
- When the estimated numerical solutions are close to the true solution, it is anticipated that the difference ($x_{NS}^{(n)} - x_{NS}^{(n-1)}$) is small compared to the value of $x_{NS}^{(n)}$, and the Estimated Relative Error is approximately the same as the True Relative Error

2.3 Bisection method

- A bracketing method to find f(x)=0 when
  - solution is within a known value of [a,b]
  - f(x) is continuous
  - the equation has one solution

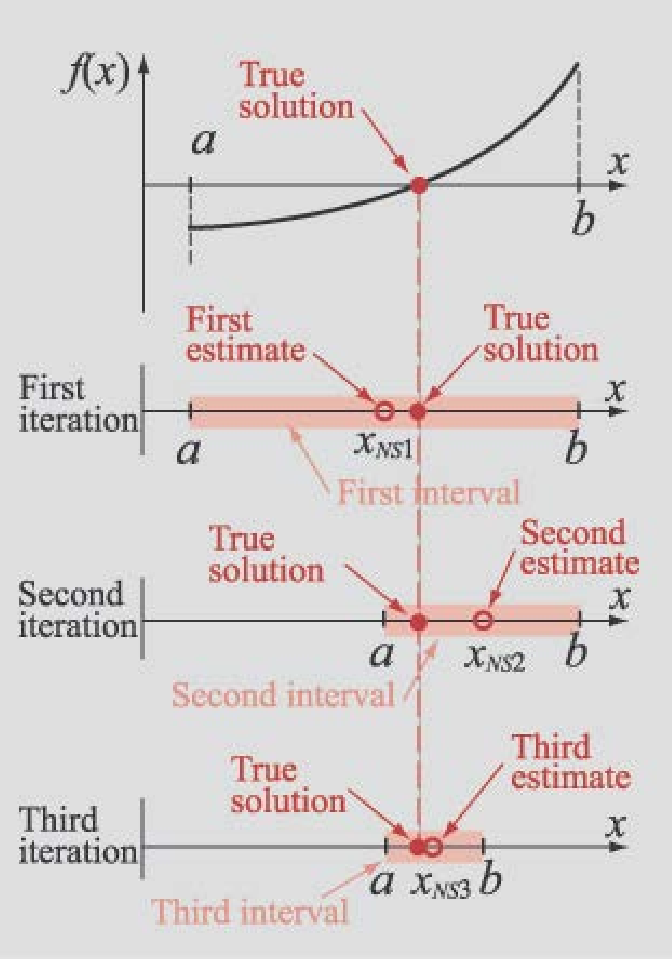


- Procedure
  - f(a)·f(b)<0
  - use the midpoint $\left(\frac{a+b}{2}\right)$ as the new estimate X_{NS1}
  - assign X_{NS1} to a or b depending on the sign
  - the new interval is the half of the original interval

Bisection method

- (1) Choose [a,b] so that f(a)f(b)<0
- (2) $X_{NS1} = \frac{a+b}{2}$
- (3) Check
  - $f(a) \cdot f(X_{NS1}) < 0 \Rightarrow [a, X_{NS1}]$
  - $f(a) \cdot f(X_{NS1}) > 0 \Rightarrow [X_{NS1}, b]$
- (4) Select a new interval [a,b] and go back to step2
- (5) Repeat until a specified tolerance

- Always converge but slowly

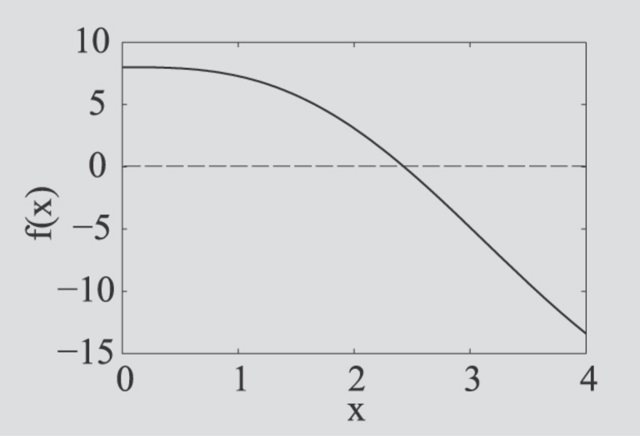


Example 2.1

- Write a MATLAB program in a script file to determine the solution of the equation using bisection method:

$$8 - 4.5(x - \sin x) = 0$$

- The solution should have a tolerance of less than 0.001 rad.
- Create a table that displays the values of a, b, x_{NS}, $f(x_{NS})$, and the tolerance for each iteration of the bisection process



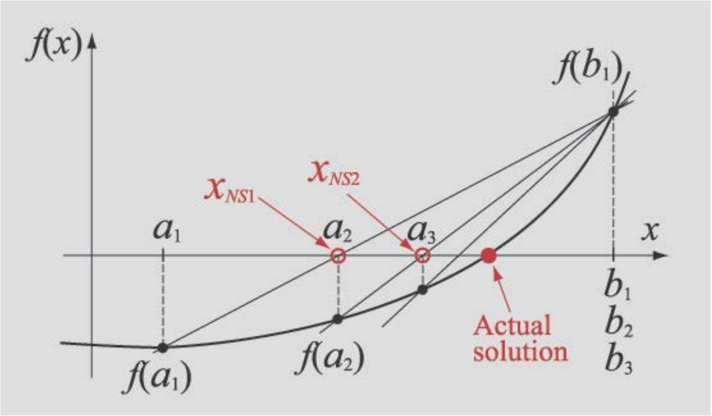
2.4 Regula falsi method

- False position method, linear interpolation method
- Bracketing method to find f(x)=0
  - (1) solution is within [a,b]
  - (2) f(x) is continuous
  - (3) the equation has a solution
- The equation for the straight line that connects (a,f(a)) and (b, f(x)):

$$y = \frac{f(b) - f(a)}{b - a}(x - b) + f(b) \qquad \text{<2-10>}$$

- the point X_{NS1} is the point when y=0 in the above equation

$$x_{NS} = \frac{af(b) - bf(a)}{f(b) - f(a)} \qquad \text{<2-11>}$$



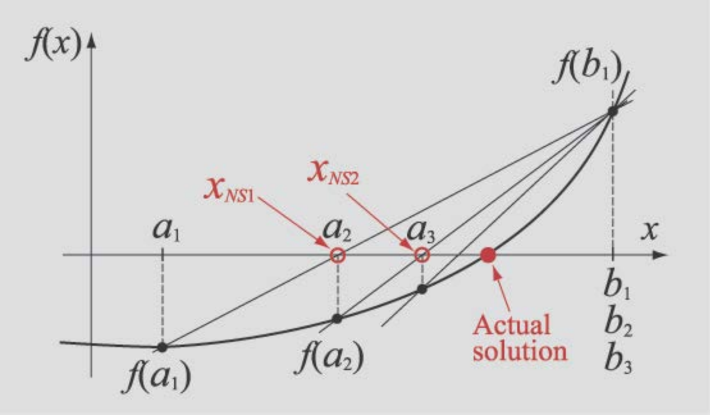
Regula falsi method

- (1) Find [a,b] so that f(a)f(b)<0
- (2) Calculate $\mathrm{X_{NS1}}$ by <2-11>
- (3) Check the solution within [a, $\mathrm{X_{NS1}}$] or [$\mathrm{X_{NS1}}$,b]
  - If f(a)·f($\mathrm{X_{NS1}}$)<0 ⇒ [a, $\mathrm{X_{NS1}}$]
  - If f(a)·f($\mathrm{X_{NS1}}$)>0 ⇒ [$\mathrm{X_{NS1}}$,b]
- (4) Select the subinterval that contains the solution

$$[a,b]_{new} = [a, X_{NS1}] or [X_{NS1}, b]$$

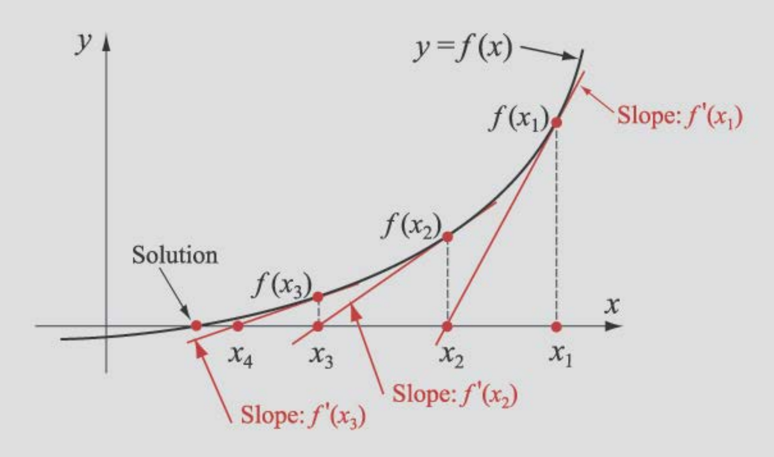
- (5) repeat (2) → (4)

- Always converges to an answer



2.5 Newton's method (Newton-Raphson method)

- Find the solution of $f(x)=0$
  - $f(x)$ is continuous and differentiable
  - Has a solution "near a given point"
- Process:
- Choose x_1 (first estimate)
- Find the intersection point of the tangent line at $(x_1, f(x_1))$ with the x axis as x_2
- Find the intersection point of the tangent line at $(x_2, f(x_2))$ with the x axis as x_3



- Since $f'(x_1) = \dfrac{f(x)-0}{x_1 - x_2}$ ➔ $x_2 = x_1 - \dfrac{f(x_1)}{f'(x_1)}$

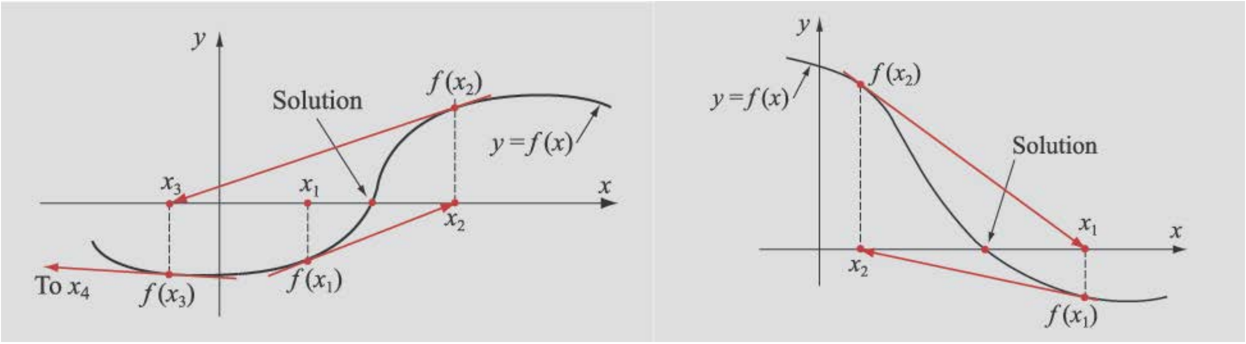
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad \text{<2-14>}$$

Newton's method

- If the equation is difficult to do the derivative, the slope can be determined numerically.

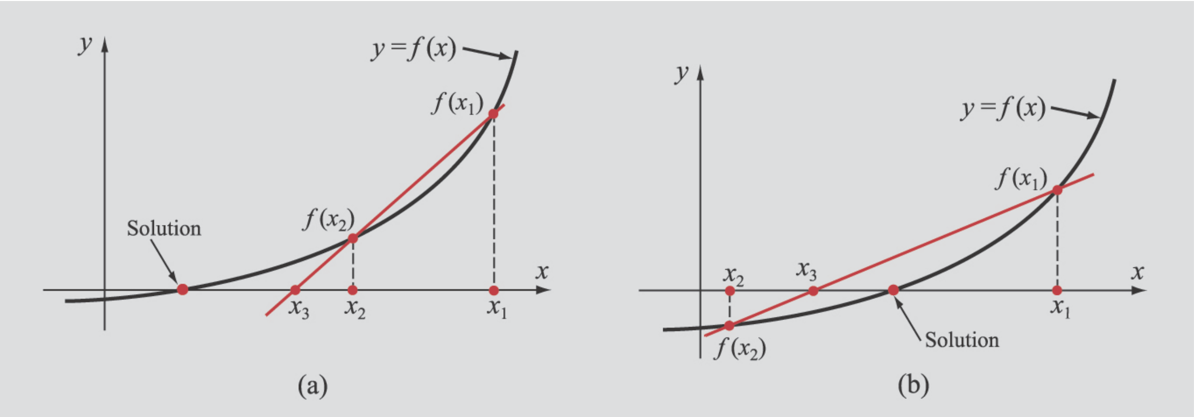
- Two error estimates:
  - Estimated relative error: $\left|\frac{x_{i+1} - x_i}{x_i}\right| \leq \varepsilon$
  - Tolerance in f(x): $\left| f(x_i) \right| \leq \delta$

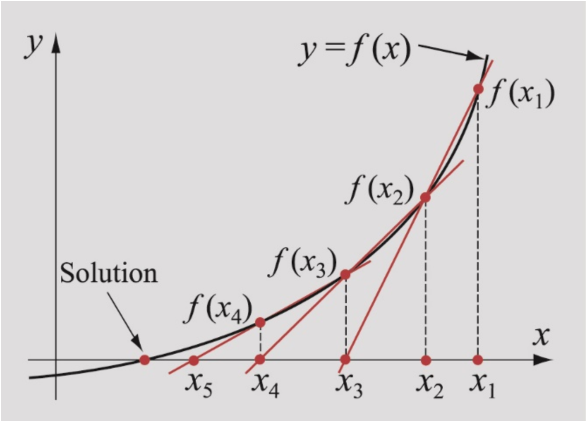
- It may not converge usually when
  - (1) Starting point is not close enough to the solution
  - (2) f'(x) is close to zero near the solution



2.6 Secant Method

- Use “two points”(x_1, x_2) in the neighborhood of the solution to determine a new estimate (x_3)
- Use $x_2$and x_3 to calculate x_4 ·······
- $$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)} \qquad \text{<2-26>}$$
- If the two points are close to each other, it is an approximated form of Newton’s method

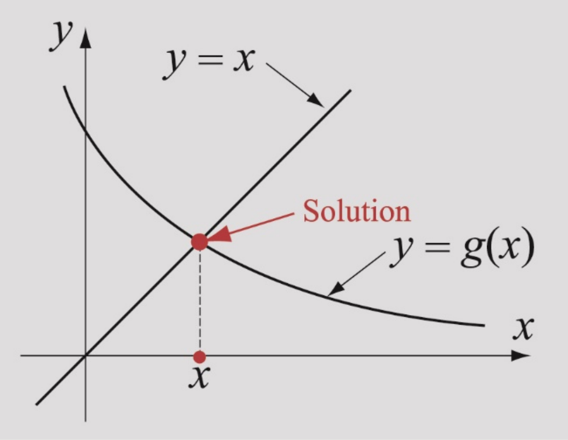




2.7 Fixed-point iteration method

- Solving the equation of f(x) = 0
- Rewrite the equation $f(x)=0 \xrightarrow{rewrite} x=g(x)$

$$\begin{cases} y = x \\ y = g(x) \end{cases}$$

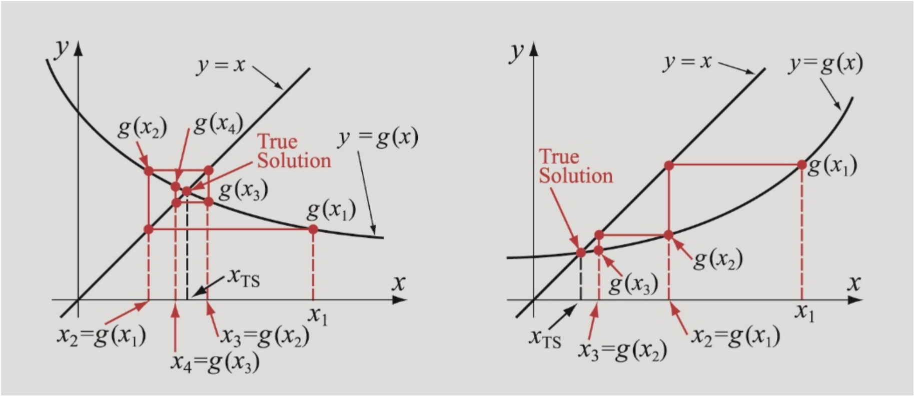


- The point of intersection of the two plots, called **fixed point**
- The numerical solution is determined by an iterative process using:

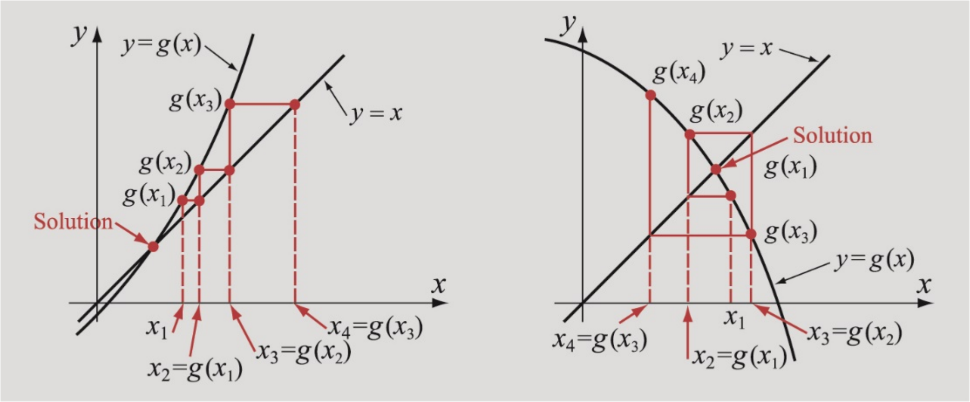
$$x_{i+1} = g(x_i) \quad \text{<2-29>}$$

Fixed-point iteration method

- Iteration function: $x_{i+1} = g(x_i)$



- The iterations may not converge toward the fixed point:



Fixed-point iteration method

- The iteration function is not unique
- Different iteration functions may yield different roots
- **Lipschitz continuous**: the fixed point method converge if the derivative of g(x) has an absolute value that is smaller than 1:

$$|g'(x)| < 1$$

- Example: $f(x) = xe^{0.5x} + 1.2x - 5 = 0$
- Case (a): $x = \dfrac{5 - xe^{0.5x}}{1.2} = g(x)$
- $g'(1) = -2.0609$ $\qquad$ $g'(2) = -4.5305$

Fixed-point iteration method

- Case (b): $x = \dfrac{5}{e^{0.5x} + 1.2}$
- $g'(1) = -0.5079$ $\qquad$ $g'(2) = -0.4426$
- Case(c): $x = \dfrac{5 - 1.2x}{e^{0.5x}}$
- $g'(1) =$ -1.8802 $\qquad$ $g'(2) = -0.9197$
- Case(b) satisfies and $x_{i+1} = \dfrac{5}{e^{0.5xi} + 1.2}$
- starting with $x_1 = 1$
- $x_2 = \dfrac{5}{e^{0.5 \cdot 1} + 1.2} = 1.7552 \quad x_3 = \dfrac{5}{e^{0.5 \cdot 1.7552} + 1.2} = 1.3869$

……. Until converging to a solution of x=1.5050

2.8 MATLAB built-in function

- fzero: solve an equation with one variable
- x = fzero(function, x_o)

(1) Write directly
(2) user-defined function
(3) Anonymous function (function handle)

- Example:

  >>Fun = @(x)8-4.5∗(x-sin(x))

  >>sol=fzero(Fun,2)
- Determine multiple roots separately

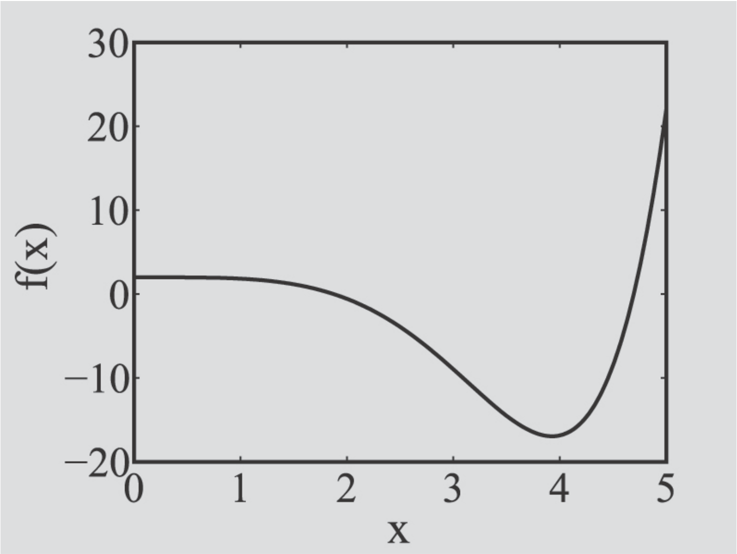
  >>Fun=@ (x) x^2-x-6 >>sol=fzero(Fun,2.5)

  >>sol=fzero(Fun,-1)

- roots: find the roots of a polynomial ➔ R=roots(p)
- >> P=[1,-1,-6]   >> R=roots(p)

2.9 Equations with multiple solutions

- Determine the approximate location of the roots by defining **smaller intervals** for each roots
- Apply any of the methods in the previous section over a **restricted subinterval**
- Use *fplot* to look for sign change at different interval



2.10 System of non-linear equations

- 2.10.1 Newton's method
- $f_1(x,y)=0$ (x_1,y_1) estimated solution
- $f_2(x,y)=0$ (x_2,y_2) true solution

- By Taylor series expansion

$$\begin{cases} f_1(x_2,y_2)= f_1(x_1,y_1)+(x_2-x_1)\dfrac{\partial f_1}{\partial x}_{|x1,y1}+(y_2-y_1)\dfrac{\partial f_1}{\partial y}_{|x1,y1}+\ldots=0 \\ f_2(x_2,y_2)= f_2(x_1,y_1)+(x_2-x_1)\dfrac{\partial f_2}{\partial x}_{|x1,y1}+(y_2-y_1)\dfrac{\partial f_2}{\partial y}_{|x1,y1}+\ldots=0 \end{cases}$$

- Neglecting high order terms

The only unknowns

$$\begin{cases} \dfrac{\partial f_1}{\partial x}_{|x_1,y_1}\Delta x+\dfrac{\partial f_1}{\partial y}_{|x_1,y_1}\Delta y=-f_1(x_{1,\backslash}y_1) \\ \dfrac{\partial f_2}{\partial x}_{|x_1,y_1}\Delta x+\dfrac{\partial f_2}{\partial y}_{|x_1,y_1}\Delta y=-f_2(x_{1,\backslash}y_1) \end{cases} \Rightarrow \begin{cases} \Delta x = x_2 - x_1 \\ \Delta y = y_2 - y_1 \end{cases} \Rightarrow \begin{cases} x_2 = x_1 + \Delta x \\ y_2 = y_1 + \Delta y \end{cases}$$

2.10 System of non-linear equations

- 2.10.2 Fixed point iteration method
- A system of n nonlinear equations
- $$\begin{cases} f_1(x_1, x_2, ...x_n) = 0 \\ f_2(x_1, x_2 ......x_n) = 0 \\ \vdots \\ f_n(x_1, x_2 ......x_n) = 0 \end{cases} \Rightarrow \begin{cases} x_1 = g_1(x_1, x_2, ...x_n) \\ x_2 = g_2(x_1, x_2, ...x_n) \\ \vdots \\ x_n = g_n(x_1, x_2, ...x_n) \end{cases}$$
- Guess $(x_1, x_2, ...x_n)_{1st}$ ➔ substitute into $g_1 ...... g_n$
- Second estimate $(x_1, x_2, ...x_n)_{2nd}$ ➔ substitute into $g_1 ...... g_n$
- The procedure is repeated until convergence