ESO208A: Computational Methods in Engineering

Richa Ojha

Department of Civil Engineering

IIT Kanpur

Acknowledgements: Profs. Abhas Singh and Shivam Tripathi (CE)

Solution of non-linear equations

Mathematical Preliminaries

Solution of non-linear equations

Mathematical preliminaries

a. Intermediate value theorem for continuous functions

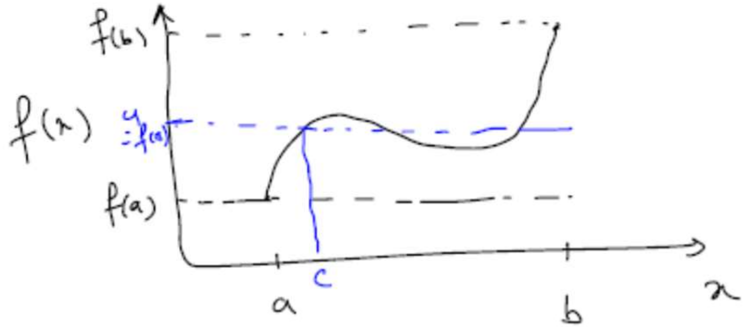
$I = [a, b] \qquad b > a$

Continuous function $f : I \to \mathbb{R}$

If u is a number between $f(a)$ and $f(b)$

i.e. $u \in (f(a), f(b))$

Then there is $c \in (a, b)$ such that $f(c) = u$



Mathematical Preliminaries

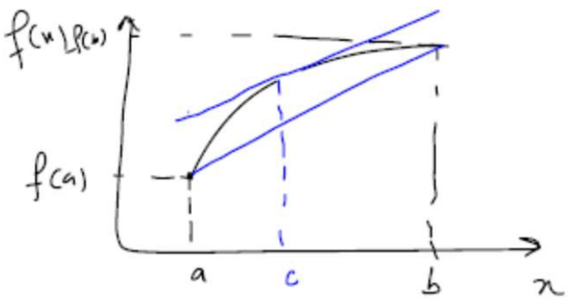
(b) Mean value theorem (MVT)

$$I = [a, b] \qquad b > a$$

$$f : I \to \mathbb{R}$$

There exists $c \in (a, b]$ such that

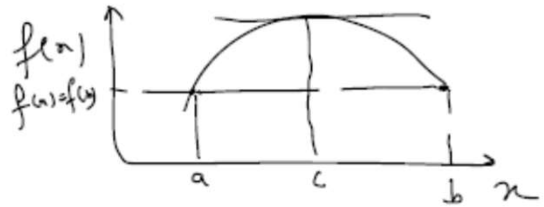
$$f'(c) = \frac{f(b) - f(a)}{b - a}$$



Rolle's theorem

$$f(a) = f(b)$$

$$f'(c) = 0$$



MVT for integrals

If $g(x)$ be a non-negative or non-positive integrable function

$$\int_a^b f(x)\, g(x)\, dx = f(c) \int_a^b g(x)\, dx$$

$$c \in (a, b)$$

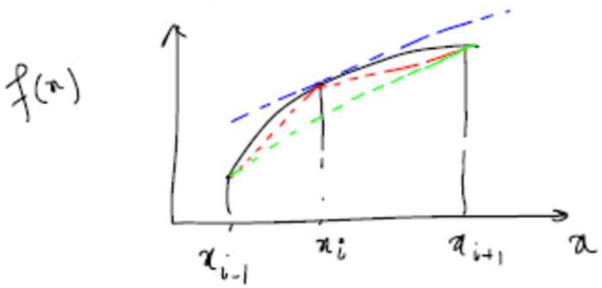
Mathematical Preliminaries

Numerical Differential

$$f(x_{i+1}) = f(x_i) + \underbrace{(x_{i+1} - x_i)}_{\Delta x} f'(x_i) + \frac{(x_{i+1}-x_i)^2}{2!} f''(x_i) + \frac{(x_{i+1}-x_i)^3}{3!} f'''(x_i) + \ldots \quad (1)$$

$$\Rightarrow \boxed{f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{\Delta x} + O(\Delta x)}$$

First forward defference



$$f(x_{i-1}) = f(x_i) + -\Delta x f'(x_i) + \frac{\Delta x^2}{2!} f''(x_i) - \frac{\Delta x^3}{3!} f'''(x_i) \quad (2)$$

$$\boxed{f'(x_i) = \frac{f(x_i) - f'(x_{i-1})}{\Delta x} + O(\Delta x)}$$

first Backward difference

1 - 2

$$\boxed{f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2\Delta x} + O(\Delta x^2)}$$

Central defference

Non-linear equation

$$f(x) = 0$$

To find the value of x

$$f(x) = ax^2 + bx + c = 0$$

- This is a quadratic equation, and has an analytical solution.
- Not all equations have analytical solution. So we may have to use computer

Non-linear equation

In computer, we have five approaches

- **Graphical method**
- **Bracketing methods:** Bisection, Regula-Falsi
- **Open methods:** Fixed point, Newton-Raphson, Secant, Muller
- **Special methods for polynomials:** Bairstow's
- **Hybrid methods:** Brent's

Graphical Method

One of the best methods to get an insight.

$$f(x) = 0$$

Example (i) $f(x) = e^{-x} - x = 0$

(ii) $f(x) = (1-x)^6 = 0$

$f(x) = 1 - 6x + 15x^2 - 20x^3 + 15x^4 - 6x^5 + 6$
$= 0$

In rare cases it is possible to find exact solution

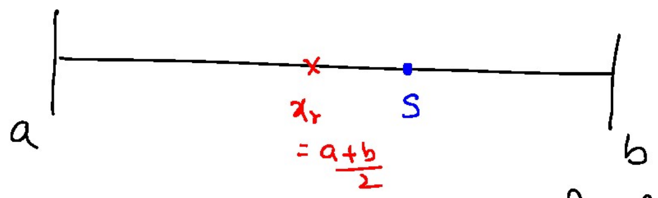
Final value depends on how much you zoom in

Bracketing Methods

1. Bisection Method

We know that the solution lies between a and b

$$x_r = \frac{a+b}{2}$$

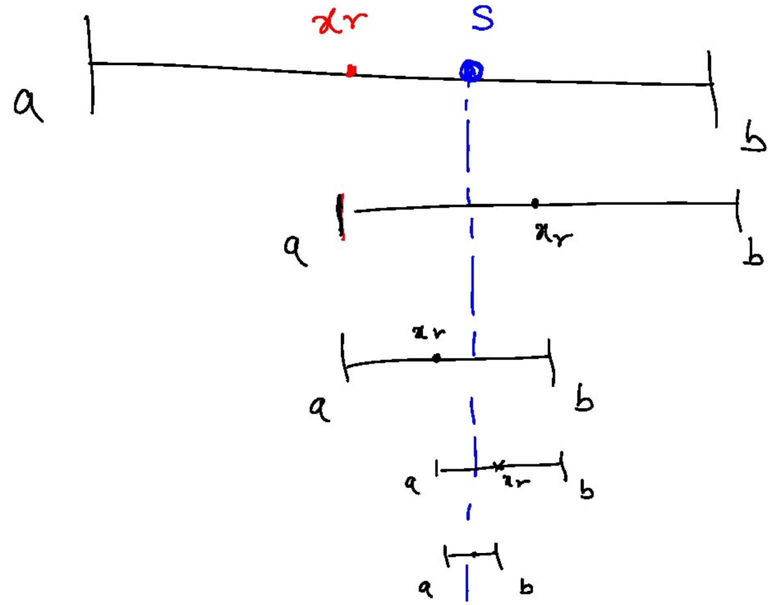


$f(a)f(x_r) < 0$

$f(a)f(x_r) > 0$

$f(a)f(x_r) = 0$

Bisection Method



Bisection Method

- Principle: Choose an initial interval based on intermediate value theorem and halve the interval at each iteration step to generate the nested intervals.

- Initialize: Choose a_0 and b_0 such that, $f(a_0)f(b_0) < 0$. This is done by trial and error.

- Iteration step k:
  - Compute mid-point $m_{k+1} = (a_k + b_k)/2$ and functional value $f(m_{k+1})$
  - If $f(m_{k+1}) = 0$, m_{k+1} is the root. (It's your lucky day!)
  - If $f(a_k)f(m_{k+1}) < 0$: $a_{k+1} = a_k$ and $b_{k+1} = m_{k+1}$; else, $a_{k+1} = m_{k+1}$ and $b_{k+1} = b_k$
  - After n iterations: size of the interval $d_n = (b_n - a_n) = 2^{-n} (b_0 - a_0)$, stop if $d_n \leq \varepsilon$
  - Estimate the root ($x = \alpha$ say!) as: $\alpha = m_{n+1} \pm 2^{-(n+1)} (b_0 - a_0)$

Bisection Method

Maximum error at 0^{th} step

Error Analysis

$$E^0 = |x_l - x_u| = \Delta x^0$$

$$E^1 = \frac{\Delta x^0}{2}$$

$$\vdots$$

$$\boxed{E^n = \frac{\Delta x^0}{2^n}}$$

Error bound reduces with iterations hence the algorithm will converge

$$\frac{|E_{i+1}|}{|E_i|} = \frac{1}{2}$$

$p = 1$

Linear convergence

$c = 1/2$

Rate of convergence for an iterative sequence

If an iterative sequence $x_r^1, x_r^2, \ldots$ converges to the solution S, and the true error $e^i = S - x_r^i$

and if

$$\lim_{i \to \infty} \frac{|e_{i+1}|}{|e_i|^p} = c$$

Then p — order of convergence

c — asymptotic error constant

$c > 1$ diverging $\quad c < 1$ converging

Bisection Method

Stopping Criteria

(i) maximum number of iterations

(ii) $\varepsilon = x_r^{new} - x_r^{old}$ $\quad |\varepsilon| \leq \underline{\underline{\varepsilon_s}}$

Threshold

or $\varepsilon_r = \dfrac{x_r^{new} - x_r^{old}}{x_r^{new}} \times 100$.

(iii) Maximum number of iterations can be estimated a priori

$$E^n \leq \alpha \qquad \alpha = 0.01$$

$$\Rightarrow \frac{\Delta x^0}{2^n} \leq \alpha$$

$$\Rightarrow \boxed{n \geq \frac{1}{\log(2)} \log\left(\frac{\Delta x^0}{\alpha}\right)}$$

(iv) Approximate error is always greater than true error

Approximate error is an exact upper bound for the true error

Bisection Method

Stopping Criteria

(i) maximum number of iterations

(ii) $\varepsilon = x_r^{new} - x_r^{old}$ $\qquad |\varepsilon| \leq \varepsilon_s$

Threshold

or $\varepsilon_r = \frac{x_r^{new} - x_r^{old}}{x_r^{new}} \times 100$

(iii) Maximum number of iterations can be estimated a priori

$$E^n \leq \alpha \qquad \alpha = 0.01$$

$$\Rightarrow \frac{\Delta x^0}{2^n} \leq \alpha$$

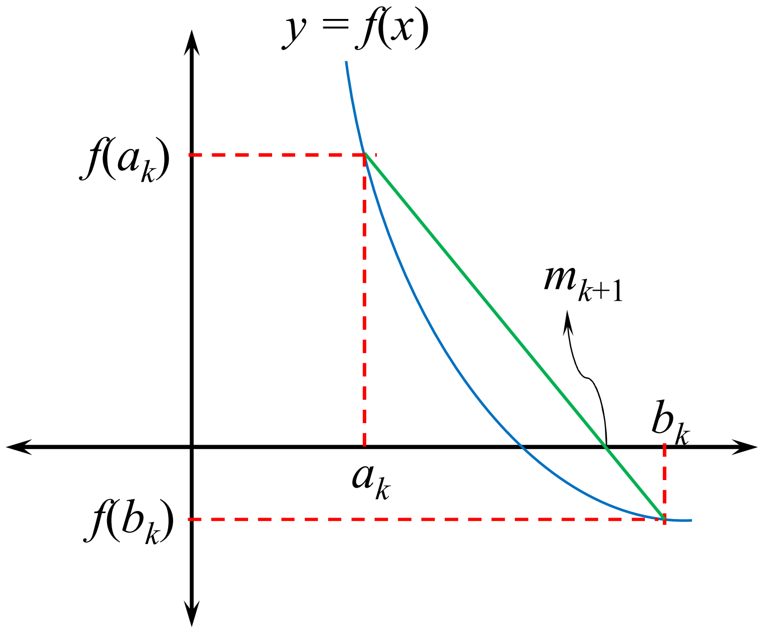
$$\Rightarrow \boxed{n \geq \frac{1}{\log(2)} \log\left(\frac{\Delta x^0}{\alpha}\right)}$$

(iv) Approximate error is always greater than true error

Approximate error is an exact upper bound for the true error

Bracketing Methods

2. The method of false position



Regula-Falsi or Method of False Position

- Principle: In place of the mid point, the function is assumed to be linear within the interval and the root of the linear function is chosen.
- Initialize: Choose a_0 and b_0 such that, $f(a_0)f(b_0) < 0$. This is done by trial and error.
- Iteration step k:
  - A straight line passing through two points $(a_k, f(a_k))$ and $(b_k, f(b_k))$ is given by: $\frac{x-a_k}{f(x)-f(a_k)} = \frac{b_k-a_k}{f(b_k)-f(a_k)}$
  - Root of this equation at $f(x) = 0$ is: $x = m_{k+1} = a_k - \frac{b_k-a_k}{f(b_k)-f(a_k)} f(a_k)$
  - If $f(m_{k+1}) = 0$, m_{k+1} is the root. (It's your lucky day!)
  - If $f(a_k)f(m_{k+1}) < 0$: $a_{k+1} = a_k$ and $b_{k+1} = m_{k+1}$; else, $a_{k+1} = m_{k+1}$ and $b_{k+1} = b_k$
  - After n iterations: size of the interval $d_n = (b_n - a_n)$, stop if $d_n \le \varepsilon$
  - Estimate the root ($x = \alpha$ say!) as: $\alpha = a_n - \frac{b_n-a_n}{f(b_n)-f(a_n)} f(a_n)$

Bracketing method

- False position method also has linear convergence. The constant may be different from ½.
- False position method works faster than bisection method.
- No one algorithm can be claimed to be universally superior then other. (No free lunch theorem!)
  - If you have more than one solution. The bisection method will find only one of them. If you want to find multiple roots have separate bounds for different roots

Look for Modified False Position Method!

Bracketing Methods

Advantages

- Convergence to a root is guaranteed (may not get all the roots, though!)
- Simple to program
- Computation of derivative not needed

Disadvantages

- Slow convergence
- For more than one roots, may find only one solution by this approach.

Summary

- What is bisection method?
- What is false-position method?