# Numerical Methods

## (MTH4002)

## Lecture 02: Solving Nonlinear Equations

**Dr. Kundan Kumar**
Associate Professor
Department of ECE

Faculty of Engineering (ITER)
S'O'A Deemed to be University, Bhubaneswar, India-751030

# Outline

# Introduction

- A problem of great importance in science and engineering is that of determining the roots/ zeros of an equation of the form

$$f(x) = 0$$

where $f(x)$ is a polynomial of degree $n$ in $x$ or called transcendental function.

- A polynomial equation of the form

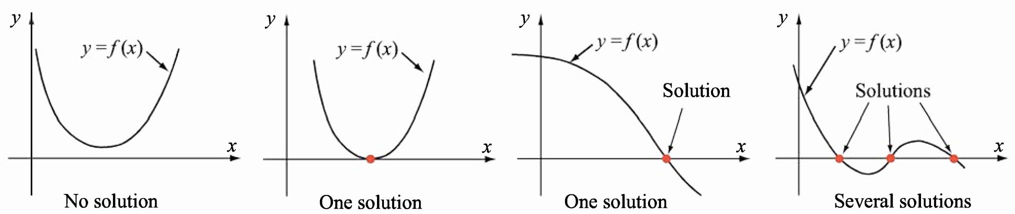
$$a_1x^n + a_2x^{n-1} + \ldots + a_nx + a_{n+1} = 0, \tag{1}$$

Ex.: $x^4 - 3x^2 + 1 = 0$.

- Exponential, logarithmic, trigonometric functions, etc. consisting of $\sin(x)$, $\cos(x)$, $\exp(x)$, $\log(x)$ etc.
Ex.: $xe^{2x} - 1 = 0$, $\cos(x) - xe^x = 0$, $\tan(x) = 0$.

# Introduction

- A number $\alpha$ for which $f(\alpha) \approx 0$ is called root of a equation $f(x) = 0$ or zero of $f(x)$.
- Geometric meaning?
  - A root of an equation $f(x) = 0$ is the value of $x$ at which the graph of the equation $y = f(x)$ intersect the $x$-axis.
- An equation might have no solution or can have one or several (possibly many) roots.



# Method of finding roots

- Direct methods: Analytical approach
  - Example: finding roots of quadratic equation $ax^2 + bx + c = 0$, $a \neq 0$

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

.

  - Difficult to solve the problem as the order will increase.
- Iterative methods: Numerical approach
  - Based on successive approximation

$$x_0, \quad x_1, \quad \ldots, \quad x_k$$

when $k \to \infty$, $x$ converge to exact root $\alpha$.

  - Starts with initial approximation (one or two)
  - How to find initial approximate value of $x$?

# Approach to solve the problems

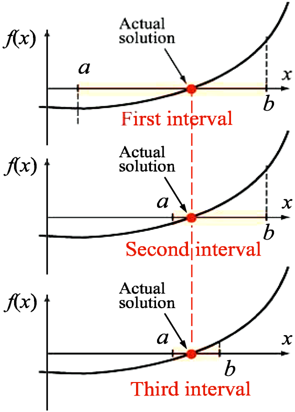
- Initially, find the approximate numerical solution of an equation $f(x) = 0$ by plotting $f(x)$ versus $x$ and looking for the point where the graph crosses the x-axis. It starts at one value of $x$ and then changes the value of $x$ in small increments.
- A change in the sign of $f(x)$ indicates that there is a root within the last increment. Decide the interval $[a, b]$, where graph crosses the x-axis.

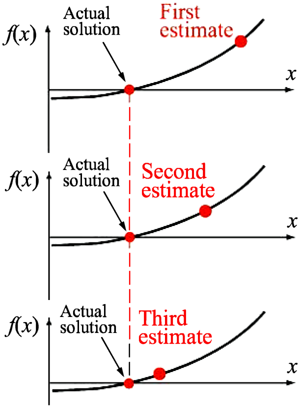
$$f(a)f(b) < 0$$

- Then, numerical methods can be used to solve for $x$ approaching to exact solution.
- When an equation has more than one root, a numerical solution is obtained one root at a time.

# Broad classification

- Bracketing methods
- Open methods



# Different methods to solve transcendental function

- There are four techniques which may be used to find the root of a univariate (single variable) function:
  1. Bisection method
  2. False-position method
  3. Newton's method
  4. Secant method

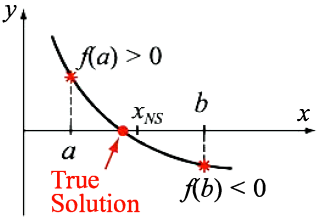
# Bisection Method

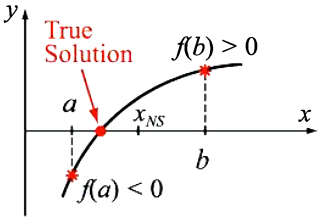
The bisection method is simple, robust, and straight-forward: take an interval $[a, b]$ such that $f(a)$ and $f(b)$ have opposite signs, find the midpoint of $[a, b]$, and then decide whether the root lies on $[a, (a + b)/2]$ or $[(a + b)/2, b]$. Repeat until the interval is sufficiently small.

# Bisection Method

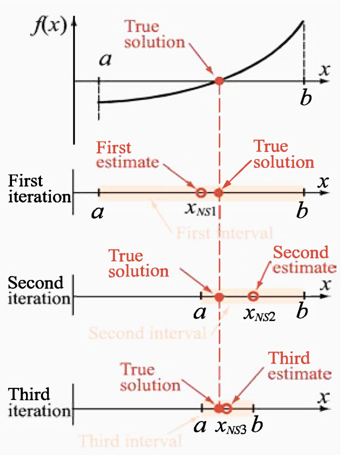
- The bisection method is a bracketing method for finding a numerical solution of an equation of the form $f(x) = 0$, given that
  - the equation has a solution and
  - $f(x)$ is continuous within a given interval $[a, b]$.
- When this is the case, $f(x)$ will have opposite signs at the end points of the interval.

$$\boxed{f(a) \cdot f(b) < 0}$$



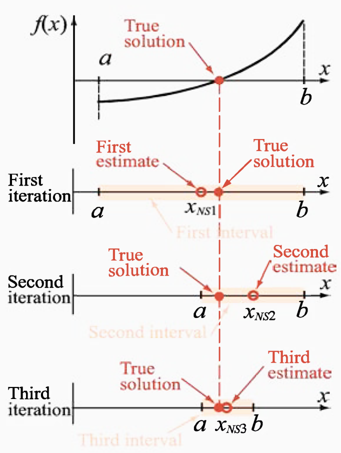
# Process of finding a solution




Question: Perform four iterations of the bisection method to obtain the smallest positive root of the equation

$$f(x) = x^3 - 5x + 1 = 0$$

# Process of finding a solution




- It starts by finding points $a$ and $b$ that define an interval where a solution exists.
- Such an interval is found either by plotting $f(x)$ and observing a zero crossing, or by examining the function for sign change.
- The midpoint of the interval $x_{NS1}$ is then taken as the first estimate for the numerical solution.
- The true solution is either in the section between points $a$ and $x_{NS1}$ or in the section between points $x_{NS1}$ and $b$.
- If the numerical solution is not accurate enough, a new interval that contains the true solution is defined.
- The new interval is the half of the original interval that contains the true solution, and its midpoint is taken as the new (second) estimate of the numerical solution.
- The process continues until the numerical solution is accurate enough according to a criterion that is selected.

# Algorithm for the Bisection Method

**Algorithm 1:** Algorithm for Bisection method

**Result:** To find the roots of $f(x) = 0$ in the interval $a \leq x \leq b$

**Initialization:** obtain $f(x), a, b$, error tolerance, maximum no of iterations;

1. Check the condition $f(a) \cdot f(b) < 0$. If does not satisfy then return a message and exit.
2. Calculate the first estimate of the numerical solution $x_{NS}$:
$$x_{NS} = \frac{(a+b)}{2}$$
3. Determine whether the true solution is between $a$ and $x_{NS}$ or between $x_{NS}$ and $b$. This is done by checking the sign of the product $f(a) \cdot f(x_{NS})$ :
   - If $f(a) \cdot f(x_{NS}) < 0$, the true solution is between $a$ and $x_{NS}$.
   - If $f(a) \cdot f(x_{NS}) > 0$, the true solution is between $x_{NS}$ and $b$.
4. Select the subinterval that contains the true solution ($a$ to $x_{NS}$, or $x_{NS}$ to $b$) as the new interval $[a, b]$ and go back to step 2.
5. Steps 2 through 4 are repeated until a specified tolerance or error bound is attained.

# Pseudocode

```
1.  Initialize f(x), a, b, epsilon
2.  Find f(a), f(b), N
3.  If (f(a)*f(b)>0)
        Display "Wrong choice of interval"
        Stop
    End If
4.  Set k = 0
5.  while (k <= N) or (|a-b| >= epsilon)
        xNS=(a+b)/2
        If ( f(a) * f(xNS) < 0 )
          new range [a, b] = [a, xNS]
        Else
          new range [a, b] = [xNS, b]
        End If
6.  Approximate_root = (a+b)/2; % (any point between [a, b] will do
                                % as the interval [a, b] is very ...
                                   small)
7. Stop
```

# Stopping criteria

- Ideally, the bisection process should be stopped when the true solution is obtained. This means that the value of $x_{NS}$ is such that $f(x_{NS}) = 0$.
- In practice, the process is stopped when the estimated error is smaller than some predetermined value.
- If the error of tolerance $(\epsilon)$ is given then the number of iteration required is

$$N = \left\lceil \frac{(\ln(|b-a|) - \ln(\epsilon_{step}))}{\ln(2)} \right\rceil$$

$$\Rightarrow N = \left\lceil \log_2 \frac{|a-b|}{\epsilon_{step}} \right\rceil$$

- If the absolute value of function is less than predefined error of tolerance $(\epsilon_{abs})$

$$|f(a)| < \epsilon_{abs} \quad \text{and} \quad |f(b)| < \epsilon_{abs}$$

# Merits and demerits

- The bisection method is always convergent. Since the method brackets the root, the method is guaranteed to converge.
- As iterations are conducted, the interval gets halved. So one can guarantee the error in the solution of the equation reduces.
- The bisection method is slower than other methods.
- The bisection method only finds roots where the function crosses the x axis. It cannot find roots where the function is tangent to the $x$ axis (Example: $f(x) = x^2$).
- The bisection method cannot find complex roots of polynomials.
- For functions $f(x)$ where there is a singularity and it reverses sign at the singularity, the bisection method may converge on the singularity. (A singularity in a function is defined as a point where the function becomes infinite. For example, for a function such as $f(x) = 1/x$, the point of singularity is $x = 0$ as it $f(0) = \infty$.)

# Example

Example 01: The function $h(x) = x\sin(x)$ occurs in the study of undamped forced oscillations. Find the value of $x$ that lies in the interval $[0, 2]$, where the function takes on the value $h(x) = 1$ (the function $\sin(x)$ is evaluated in radians).

# Example

Example 02: Find the roots of $f(x) = \frac{1}{4}x^2 - 3$ using the bisection method in the range $[2, 4]$. (Assume the error tolerance $\delta = 10^{-2}$)

# Example

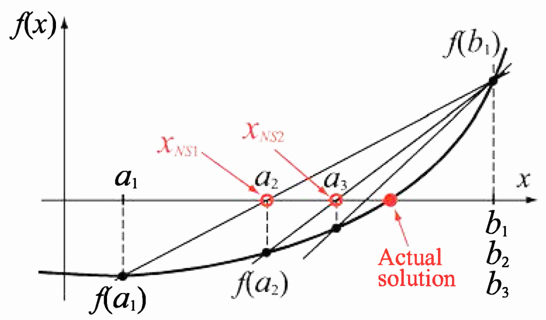
Example 03: You are working for a company that makes floats for different applications. The floating ball has a specific gravity of 0.6 and has a radius of 5.5 cm. You are asked to find the depth to which the ball is submerged when floating in water. The equation that gives the depth $x$ to which the ball is submerged under water is given by

$$x^3 - 0.165x^2 + 3.993 \times 10^{-4} = 0$$

Use the bisection method of finding roots of equations to find the depth $x$ to which the ball is submerged under water. Conduct three iterations to estimate the root of the above equation. Find the absolute relative approximate error at the end of each iteration, and the number of significant digits at least correct at the end of each iteration.

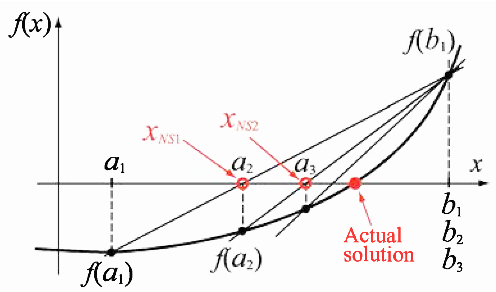
# Regula-Falsi Method

# Regula-Falsi Method




- Also called False-position method.
- Regual-Falsi method is a bracketing method for finding a numerical solution of an equation of the form $f(x) = 0$ when it is known that, within a given interval $[a, b]$, $f(x)$ is continuous and the equation has a solution.

# Regula-Falsi Method




- For a given interval $[a, b]$, the equation of a straight line that connects point $(b, f(b))$ to point $(a, f(a))$ is given by:

$$y = \frac{f(b) - f(a)}{b - a}(x - b) + f(b)$$

- The point $x_{NS}$ where the line intersects the x-axis is determined by substituting $y = 0$, and solving the equation for $x$:

$$x_{NS} = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

# Process of finding solution

- The solution starts by finding an initial interval $[a_0, b_0]$ that brackets the solution.
- The values of the function at the endpoints are $f(a_0)$ and $f(b_0)$.
- The endpoints are then connected by a straight line, and the first estimate of the numerical solution, $x_{NS0}$, is the point where the straight line crosses the $x$-axis.

# Process of finding solution

- For the next iteration a new interval, $[a_1, b_1]$ is defined. The new interval is a subsection of the first interval that contains the solution. It is either $[a_0, x_{NS0}]$ ($a_0$ is assigned to $a_1$, and $x_{NS0}$ to $b_1$) or $[x_{NS0}, b_0]$ ($x_{NS0}$ is assigned to $a_1$, and $b_0$ to $b_1$).
- The endpoints of this interval are next connected with a straight line, and the point where this new line crosses the $x$-axis is the next estimate of the solution, $x_{NS1}$.
- For the second iteration, a new subinterval $[a_2, b_2]$ is selected, and the iterations continue in the same way until the numerical solution is deemed accurate enough.

# Algorithm for the Regula-Falsi Method

**Algorithm 2:** Algorithm for Regula-falsi method

**Result:** To find the roots of $f(x) = 0$ in the interval $a \leq x \leq b$

**Initialization:** obtain $f(x), a, b$, error tolerance, maximum no of iterations;

1. Choose the first interval by finding points $a$ and $b$ such that a solution exists between them.
2. Calculate the first estimate of the numerical solution $x_{NS1}$ by:

$$x_{NS1} = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

3. Determine whether the true solution is between $a$ and $x_{NS1}$ or between $x_{NS1}$ and $b$. This is done by checking the sign of the product $f(a) \cdot f(x_{NS1})$ :
   - If $f(a) \cdot f(x_{NS1}) < 0$, the true solution is between $a$ and $x_{NS1}$.
   - If $f(a) \cdot f(x_{NS1}) > 0$, the true solution is between $x_{NS1}$ and $b$.
4. Select the sub-interval that contains the true solution ($a$ to $x_{NS1}$, or $x_{NS1}$ to $b$) as the new interval $[a, b]$, and go back to step 2.
5. Steps 2 through 4 are repeated until a specified tolerance or error bound is attained.

# Pseudocode

```
1.  Read f(x), a, b, epsilon, N
2.  Find f(a), f(b)
3.  If (f(a)*f(b)>0)
        Display "Wrong choice of interval"
        Stop
    End If
3.  Set k = 0
5.  while (k <= N) or (|a-b| >= epsilon)
        xNs=(af(b)-bf(a))/(f(b)-f(a));
        If ( f(a) * f(xNs) < 0 )
          new range [a, b] = [a, xNs]
        Else
          new range [a, b] = [xNs, b]
        End If
6.  Approximate_root = (af(b)-bf(a))/(f(b)-f(a)); % (any point ...
    between [a, b] will do as the interval [a, b] is very small)
7. Stop
```

# Stopping criteria

- The iterations are stopped when the estimated error, according to one of the measures is smaller than some predetermined value.

Remarks:

- The method always converges to an answer, provided a root is initially trapped in the interval $[a, b]$.
- Frequently, the function in the interval $[a, b]$ is either concave up or concave down. In this case, one of the endpoints of the interval stays the same in all the iterations, while the other endpoint advances toward the root.
- The convergence toward the solution could be faster if the other endpoint would also "move" toward the root.
- Several modifications have been introduced to the Regula-Falsi method that make the sub-interval in successive iterations approach the root from both sides.

# Example

Example 04: The function $h(x) = x\sin(x)$ occurs in the study of undamped forced oscillations. Find the value of $x$ that lies in the interval $[0, 2]$, where the function takes on the value $h(x) = 1$ using the method of false positions (the function $\sin(x)$ is evaluated in radians).

# Example

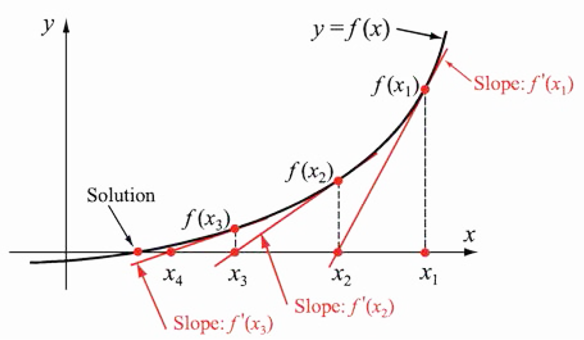
Example 05: Use the regula falsi method to determine the weight of a bungee jumper who can survive a bungee cord with a drag coefficient ($C_d$) of 0.25 without breaking while achieving a velocity of $36m/sec$ after $4sec$ of free fall.

Note: the acceleration of gravity is $9.81m/sec^2$. Start with initial guesses $a_0 = 100Kg$ and $b_0 = 200Kg$ and iterate until the relative error falls below 2%.

# Newton's Method

# Newton-Raphson method

- Newton's method (also called the Newton-Raphson method) is a scheme for finding a numerical solution of an equation of the form $f(x) = 0$ where $f(x)$ is continuous and differentiable and the equation is known to have a solution near a given point.



# Process of finding solution

- The solution process starts by choosing point $x_1$ as the first estimate of the solution. The second estimate $x_2$ is obtained by taking the tangent line to $f(x)$ at the point $(x_1, f(x_1))$ and finding the intersection point of the tangent line with the x-axis.
- The next estimate $x_3$ is the intersection of the tangent line to $f(x)$ at the point $(x_2, f(x_2))$ with the x-axis, and so on.
- Mathematically, for the first iteration, the slope, $f'(x_1)$, of the tangent at point $(x_1, f(x_1))$ is given by:

$$f'(x_1) = \frac{f(x_1 - 0)}{x_1 - x_2}$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

- Generalized solution

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

# Algorithm for Newton's method

- Choose a point $x_i$ as an initial guess of the solution.
- For $i = 1, 2, \ldots,$ until the error is smaller than a specified value, calculate $x_{i+1}$ using equation

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

Write the Pseudo-code for Newton's method.

# Stopping criterion

- Ideally, the iterations should be stopped when an exact solution is obtained. This means that the value of $x$ is such that $f(x) = 0$.
- In practice, the iterations are stopped when an estimated error is smaller than some predetermined value.

# Stopping criterion

- A tolerance in the solution, as in the bisection method, cannot be calculated since bounds are not known. Two error estimates that are typically used with Newton's method are:
  - Estimated relative error: The iterations are stopped when the estimated relative error is smaller than a specified value $\epsilon$:

$$\left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| \leq \epsilon$$

  - Tolerance in $f(x)$: The iterations are stopped when the absolute value of $f(x_i)$ is smaller than some number $\delta$:

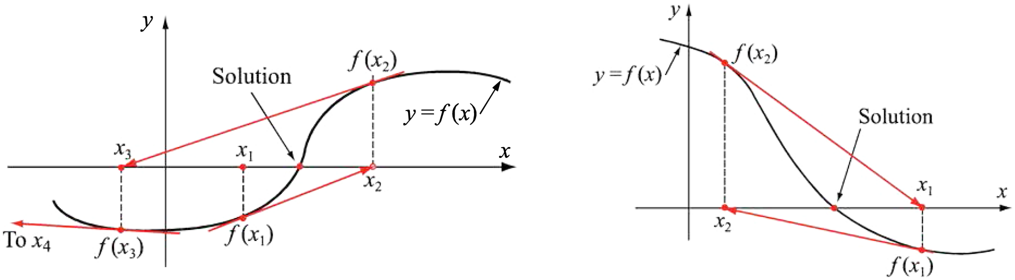
$$|f(x_i)| \leq \delta$$

# Remarks

- The method, when successful, works well and converges fast.
- When it does not converge, it is usually because the starting point is not close enough to the solution.
- Convergence problems typically occur when the value of $f'(x)$ is close to zero in the vicinity of the solution (where $f(x) = 0$).

# Convergence Criteria

- It is possible to show that Newton's method converges
  - if the function $f(x)$ and its first and second derivatives $f'(x)$ and $f''(x)$ are all continuous
  - if $f'(x)$ is not zero at the solution, and
  - if the starting value $x_1$ is near the actual solution.



# Finding roots of real positive numbers

- The Newton-Raphson method can be employed to find the roots of numbers.
- Lets assume

$$x = N^{\frac{1}{q}}, \; N > 0, q > 0 \tag{2}$$
$$\Rightarrow x^q = N \tag{3}$$

Writing

$$f(x) = x^q - N = 0 \tag{4}$$
$$\Rightarrow f'(x) = qx^{q-1} \tag{5}$$

# Finding roots of real positive numbers

- Using the Newton-Raphson formula

$$x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})} \tag{6}$$

$$\Rightarrow x_k = x_{k-1} - \frac{x_{k-1}^q - N}{q x_{k-1}^{q-1}} \tag{7}$$

$$\Rightarrow x_k = \frac{(q-1)x_{k-1}^q + N}{q x_{k-1}^{q-1}} \tag{8}$$

- As an example if we set $q = 2$ in the above equation can be employed for finding square roots.

$$x_k = \frac{(2-1)x_{k-1}^2 + N}{2 x_{k-1}^{2-1}} \tag{9}$$

$$\Rightarrow x_k = \frac{x_{k-1} + \frac{N}{x_{k-1}}}{2} \tag{10}$$

# Example

Example 06: Derive the Newton's method for finding $\frac{1}{N}$, where $N > 0$. Hence, find $\frac{1}{17}$, using the initial approximation as $x_0$ (i) $0.05$, (ii) $0.15$. Do the iterations converge ?

# Example

Example 07: Compute $17^{1/3}$ correct to four decimal places, assuming the initial approximation as $x_0 = 2$.

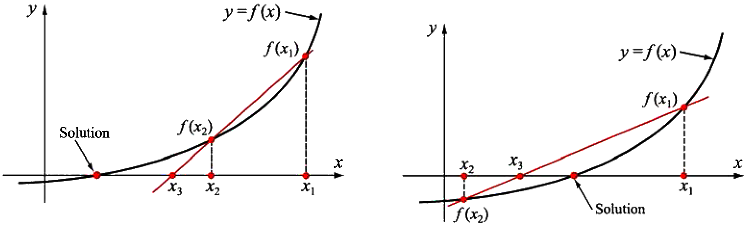
# Example

Example 08: Perform four iterations of the Newton's method to find the smallest positive root of the equation $f(x) = x^3 - 5x + 1 = 0$.

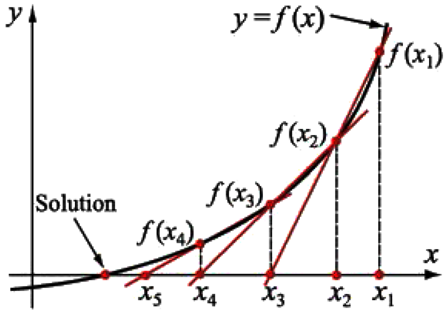
# Secant Method

# Secant Method

- The secant method is a scheme for finding a numerical solution of an equation of the form $f(x) = 0$.
- The method uses two points in the neighborhood of the solution to determine a new estimate for the solution.
- The two points (marked as $x_1$ and $x_2$ in the figure) are used to define a straight line (secant line), and the point where the line inter sects the $x$-axis (marked as $x_3$ in the figure) is the new estimate for the solution.



# Process of finding solution




- The two points can be on one side of the solution or the solution can be between the two points.
- The slope of the secant line is given by:

$$\frac{f(x_1)-f(x_2)}{x_1-x_2}=\frac{f(x_2)-0}{x_2-x_3}$$

$$x_3=x_2-\frac{f(x_2)(x_1-x_2)}{f(x_1)-f(x_2)}$$

General form,

$$x_{i+1}=x_i-\frac{f(x_i)(x_{i-1}-x_i)}{f(x_{i-1})-f(x_i)}$$

# Relationship to Newton's method

- Examination of the secant method shows that when the two points that define the secant line are close to each other, the method is actually an approximated form of Newton's method.

$$x_{i+1} = x_i - \frac{f(x_i)}{\frac{f(x_{i-1}) - f(x_i)}{(x_{i-1} - x_i)}}$$

- The denominator of the second term on the right-hand side of the equation is an approximation of the value of the derivative of $f(x)$ at $x_i$.
- In the secant method (unlike Newton's method), it is not necessary to know the analytical form of $f'(x)$.

# Pseudocode

```
1.   Start
2.   Get values of x0, x1 and e
     % x0 and x1 are the two initial guesses
     % e is the stopping criteria, absolute error or the desired
     % degree of accuracy
3.   Compute f(x0) and f(x1)
4.   Compute x2 = [x0*f(x1) - x1*f(x0)] / [f(x1) - f(x0)]
5.   Test for accuracy of x2
     If |(x2 - x1)/x2| > e,
         then assign x0 = x1 and x1 = x2
         goto step 4
     Else,
         goto step 6
     End If
5.   Display the required root as x2.
6.   Stop
```

# Merits and demerits

1. Secant method is faster than other numerical methods.
2. The order of convergence is 1.62.
3. Theoretically, Secant method is slower than Newton-Raphson method; however, practically Secant method is little faster than Newton-Raphson method.
4. It requires only one function evaluation per iteration, as compared with Newton's method which requires two.
5. It does not require use of the derivative of the function, something that is not available in a number of applications.
6. It may not converge.
7. Overall, this method proves to be the most economical one to find the root of a function.

# Example

**Example 09**: Solve the equation $f(x) = x^3 - 5x + 1 = 0$ using Secant Method.

# Order of convergence of iterative schemes

- Let $\{x_n\}_{n=1}^{\infty}$ converges to $s$
- Also, let

$$\epsilon_n = s - x_n$$

$$\epsilon_{n+1} = s - x_{n+1}$$

for $n \geq 0$ are error at $n^{th}$ and $(n+1)^{th}$ iterations respectively.

- Theorem: If two positive constant exist $A \neq 0$ and $R > 0$, and

$$\lim_{n\to\infty} \frac{|s - x_{n+1}|}{|s - x_n|^R} = \lim_{n\to\infty} \frac{|\epsilon_{n+1}|}{|\epsilon_n|^R} = A \ (+ve, \text{ non-zero, } < 1)$$

Then the sequence is said to be convergence to $s$ with order of convergence $R$. (Can be proven using Taylor's series)

# Order of convergence of Numerical Methods

- Bisection method: linear

$$\lim_{n\to\infty} \frac{|\epsilon_{n+1}|}{|\epsilon_n|^1} = \frac{1}{2} \; (+ve, \text{ non-zero, } < 1 \text{ for } R = 1)$$

- Regula-Falsi method: linear
  - Similarly as bisection method
- Newton-Raphson method: quadratic

$$R = 2$$

provided roots are simple (multiplicity 1)

- Secant method: superlinear

$$R = \frac{1}{2}(1+\sqrt{5}) = 1.618$$

# References


- Numerical Methods Using MATLAB by Matthews and Fink, Pearson
- Applied Numerical Methods with MATLAB for Engineers and Scientists, Third Edition Steven C. Chapra, McGraw-Hill
- Numerical Methods for Engineers and Scientists An Introduction with Applications using MATLAB, Third Edition, Amos Gilat and Vish Subramaniam, John Wiley & Sons

Thank you!