

Week 10: Root Finding

Acknowledgement: Un-copyrighted photos and pictures were taken from the Internet

Root Finding

- The **root** of the function $f(x)$ is the **value of x** that $f(x) = 0$.
- In many **engineering problems**, we need to **find the roots** of certain functions.
- The **procedure of finding the roots of a function** is called **root finding**.
- Two important **applications of root finding** are the **eigenvalue problem** and **optimization**.

Root Finding

- We have formulas for computing the roots of quadratic functions.

$$\begin{array}{cc} \text{For Quadratic Equation} & \text{The Quadratic Formula} \\ \mathbf{ax^2 + bx + c = 0} & \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \end{array}$$

- For complex functions, we do not have formulas for computing their roots.

$$f(x) = x^{10} + x^9 - x^8 - x^7 + x - 1$$

- We can find the roots of **complex functions** using **numerical algorithms**.

Root Finding: Eigenvalue Problem

How to compute eigenvalues of A ?

- Step1: Form the matrix $A - \lambda I$ (λ is a variable!)
- I is an n -by- n **diagonal** matrix with all entries equal to one.
- I is called an identity matrix.

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}$$

$$A - \lambda I = \begin{bmatrix} A_{11} - \lambda & A_{12} & A_{13} & \cdots & A_{1n} \\ A_{21} & A_{22} - \lambda & A_{23} & \cdots & A_{2n} \\ A_{31} & A_{32} & A_{33} - \lambda & \cdots & A_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nn} - \lambda \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$

$$\lambda I = \begin{bmatrix} \lambda & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda \end{bmatrix}$$

Diagonal matrix with all diagonal elements equal to λ

Root Finding: Eigenvalue Problem

- Step2: Find the **characteristic polynomial** of A

$$|\mathbf{A} - \lambda \mathbf{I}| = a_0 + a_1\lambda + a_2\lambda^2 + \cdots + a_n\lambda^n$$

Determinant of $\mathbf{A} - \lambda \mathbf{I}$

Characteristic polynomial of A

$$\mathbf{A} - \lambda \mathbf{I} = \begin{bmatrix} A_{11} - \lambda & A_{12} & A_{13} & \cdots & A_{1n} \\ A_{21} & A_{22} - \lambda & A_{23} & \cdots & A_{2n} \\ A_{31} & A_{32} & A_{33} - \lambda & \cdots & A_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \cdots & A_{nn} - \lambda \end{bmatrix}$$

Root Finding: Eigenvalue Problem

3. Find the **roots** of the **characteristic polynomial**.

$$a_0 + a_1\lambda + a_2\lambda^2 + \cdots + a_n\lambda^n = 0$$

This equation has n solutions:

$$\lambda_1, \dots, \lambda_n$$

Eigenvalues are the **roots** of the **characteristic polynomial**.

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}$$

Root Finding: Eigenvalue Problem

- Suppose that you need to find the **eigenvalues of a 2-by-2** matrix.
- The characteristic polynomial is

$$\det(A - \lambda I) = (a - \lambda)(d - \lambda) - bc$$

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

- The **characteristic polynomial** is **quadratic** in this case.
- We can **easily find** its roots.

Root Finding: Eigenvalue Problem

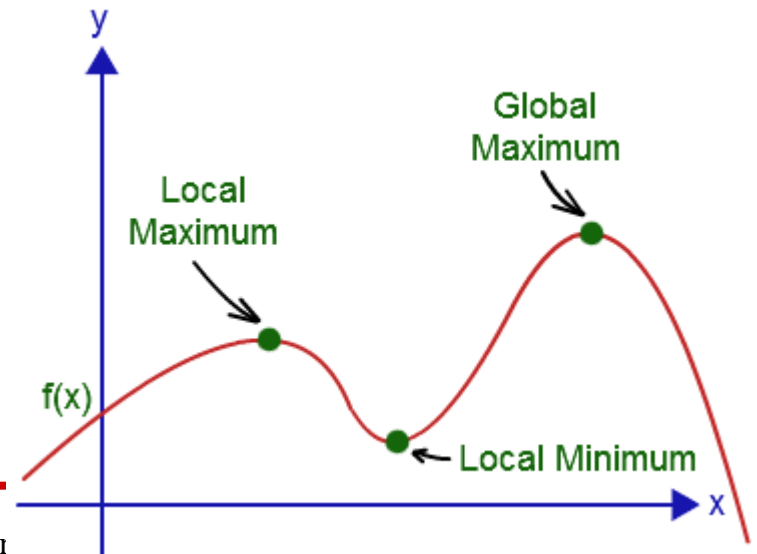
- Suppose that you need to find eigenvalues of a 10-by-10 matrix.
- The characteristic polynomial might be of order 10.
- There are **no formulas** for computing roots in this case.
- In this case, we need **efficient numerical algorithms** for **root finding**.

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}$$

$$a_0 + a_1\lambda + a_2\lambda^2 + \cdots + a_n\lambda^n = 0$$

Root Finding: Optimization Problem

- In many engineering problems, we need to **find the maximum or minimum** of a function.
- The (global) **maximum** is a point that $f(x)$ takes its **largest value**.
- The (global) **minimum** is a point that $f(x)$ takes its **smallest value**.
- The **minimum and maximum** of $f(x)$ are the solutions of $f'(x) = 0$.
- To find them, we need to find the **roots of $f'(x)$** .

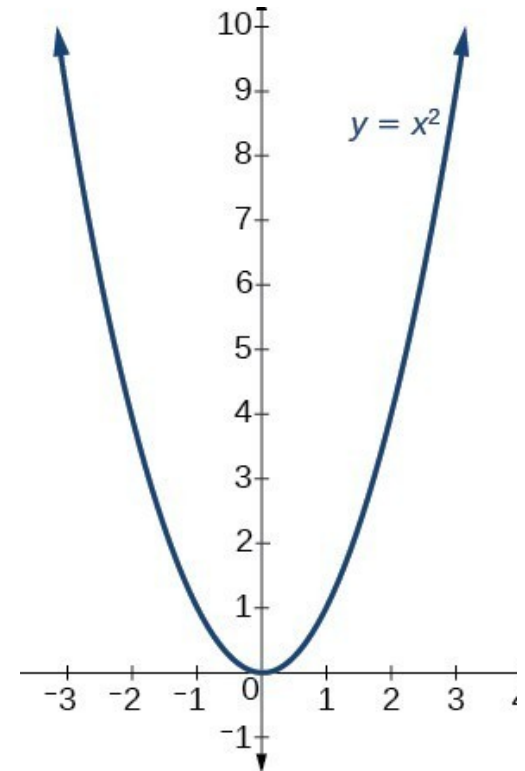


Root Finding: Optimization Problem

- Suppose that you need to find the **minimum** of $f(x) = (x)^2$
- To find it, we find the roots of $f'(x)$.

$$f'(x) = 2(x)$$

- So, the minimum is 0.
- In this case, it is easy to find the root of $2x=0$.



Root Finding: Optimization Problem

- Suppose that you need to find the minimum of

$$f(x) = x^{10} + x^9 - x^8 - x^7 + x - 1$$

- To find it, we find the roots of $f'(x)$.

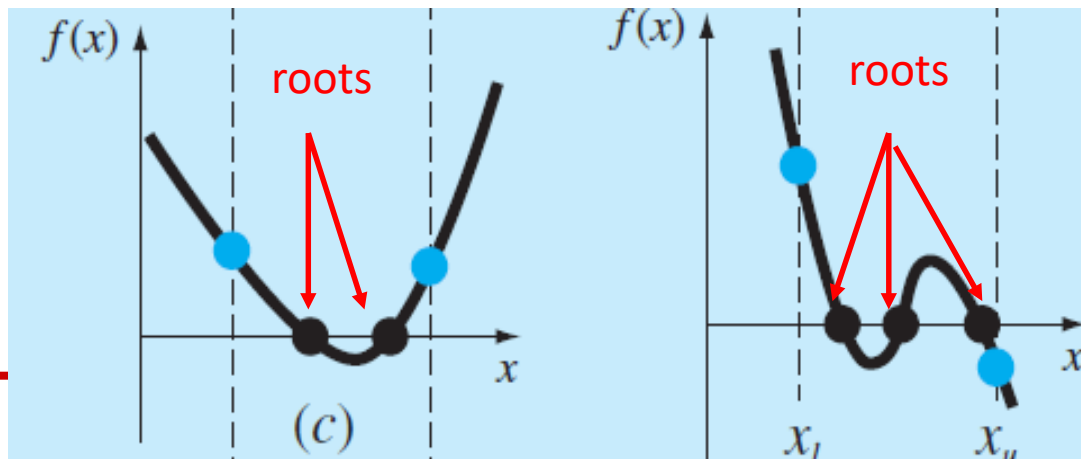
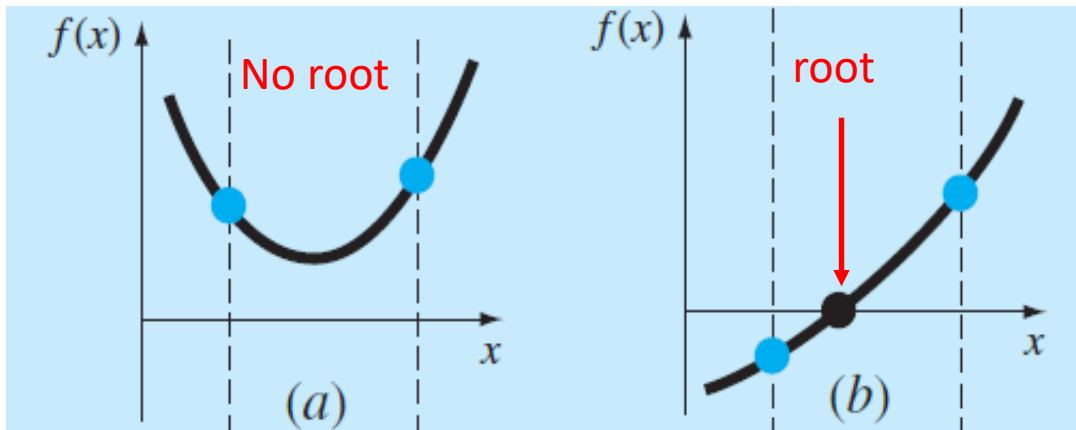
$$f'(x) = 10x^9 + 9x^8 - 8x^7 - 7x^6 + 1$$

- There are **no formulas** for computing roots in this case.
- In this case, we need **efficient numerical algorithms** for finding roots.
- This week, we will learn numerical algorithms for finding the roots of a function.

Roots of a Polynomial

Root Finding

- A function may have no (real) roots.
- A function may have 1, 2, 3, or even more roots.



Root Finding

- Consider the following polynomial of **degree n**

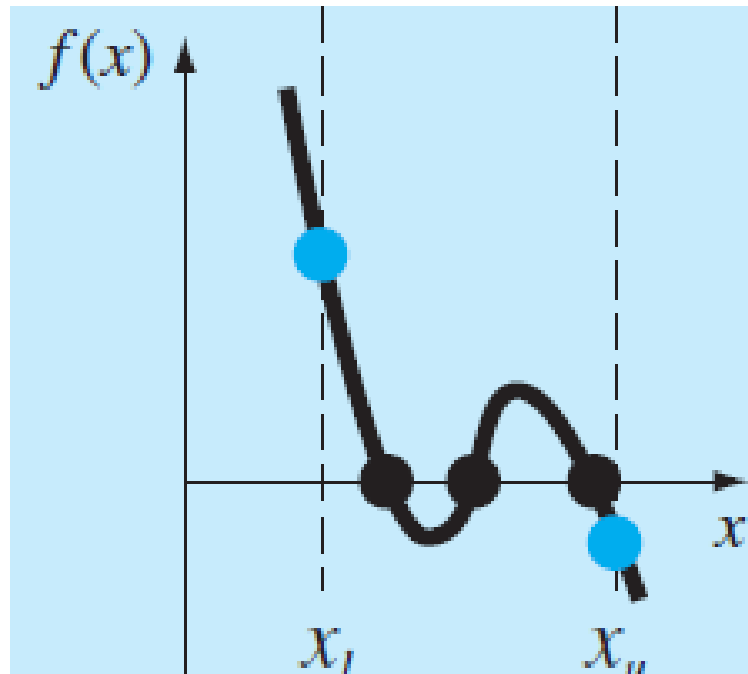
$$f_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

- It has **n roots**.
- The roots might be **real or complex numbers**.
- If **n is odd**, it has at least **one real root**.
- The **roots** are **not necessarily distinct**.
- The polynomial **$(x - 2)^2$** has a **repeated** root at **$x = 2$** .

Bisection Method

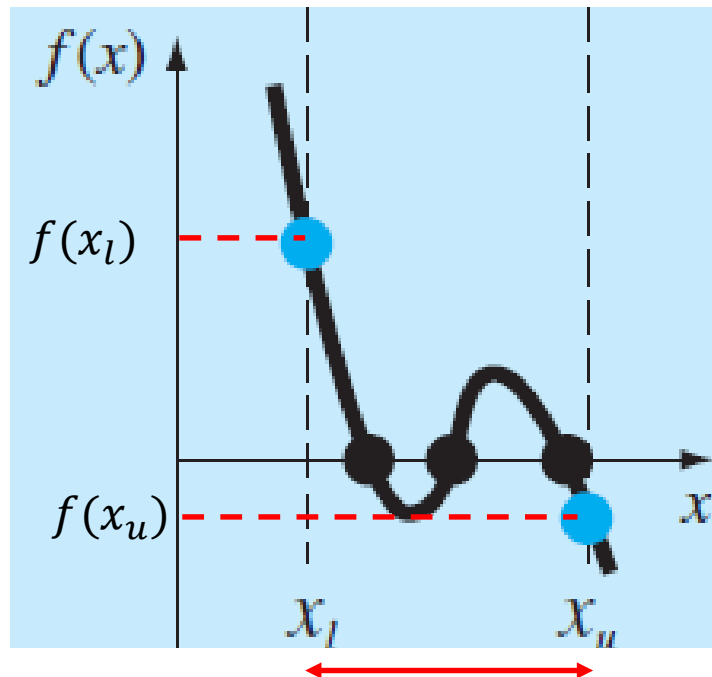
Bisection Method

- Consider the function $f(x)$ in the figure below.
- Observe that $f(x)$ changes sign on **opposite sides** of its **roots**.
- This is valid for certain roots.



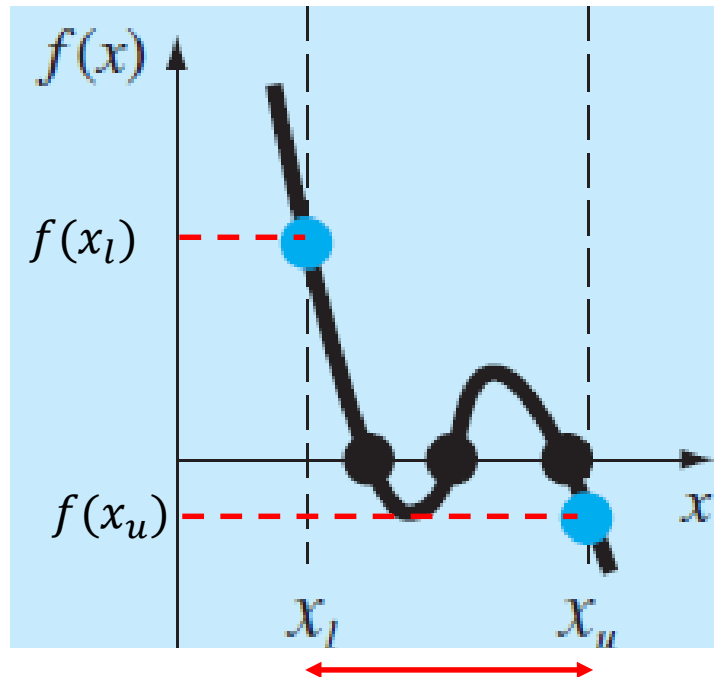
Bisection Method

- Suppose that $f(x)$ is **continuous** on the **interval** from x_l to x_u ($[x_l, x_u]$).
- Also, $f(x_l)$ and $f(x_u)$ have **opposite signs**, i.e., $f(x_l)f(x_u) < 0$
- Then, there is **at least one real root** between x_l and x_u .



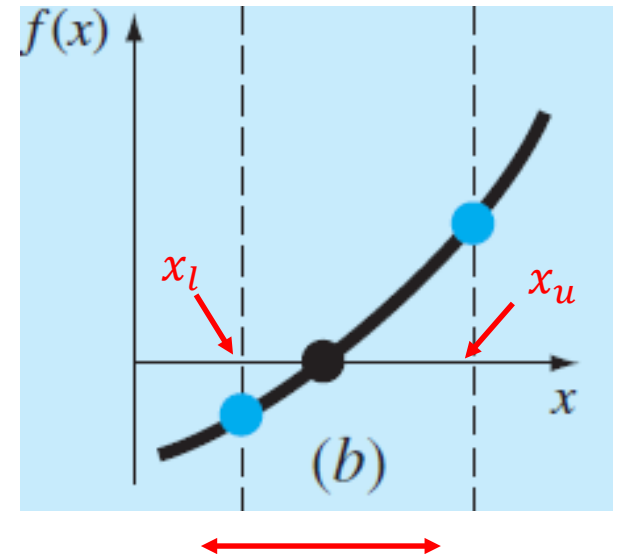
Bisection Method

- The value of $f(x)$ at x_l is positive.
- The value of $f(x)$ at x_u is negative.
- To reach from $f(x_l)$ to $f(x_u)$, the function must intersect the x-axis at least once.



Bisection Method

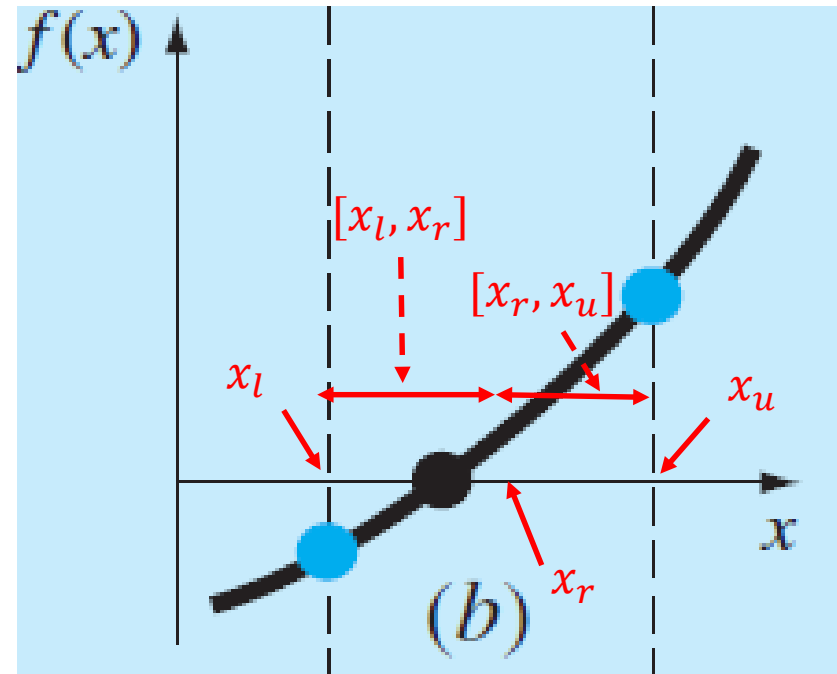
- In the bisection method, we start with an **interval** $[x_l, x_u]$.
- x_l is the left endpoint of the interval.
- x_u is the right endpoint of the interval.
- $f(x_l)$ and $f(x_u)$ must have **opposite signs**.
- So, there is at least one root in $[x_l, x_u]$.



Bisection Method

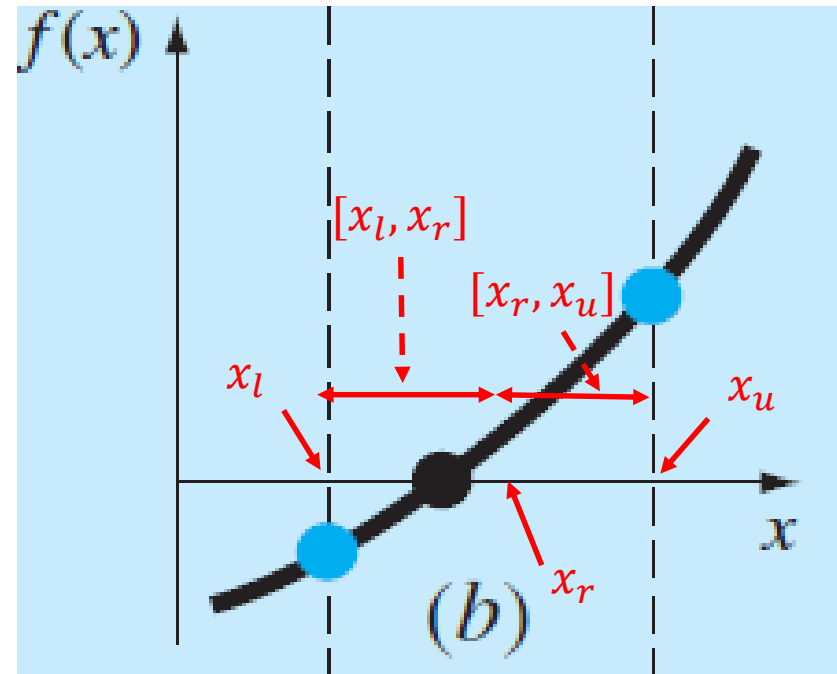
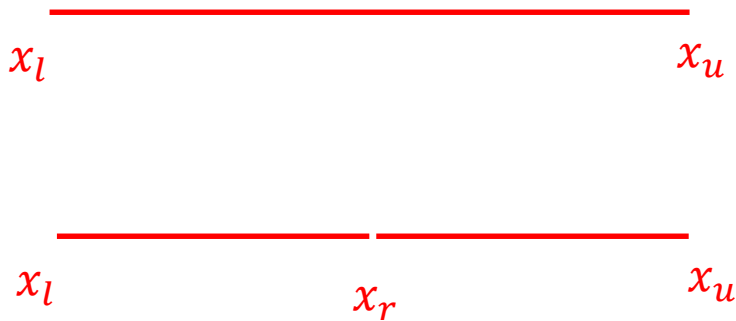
Step 1:

- Find the mid-point of the interval $[x_l, x_u]$ ($x_r = \frac{x_l + x_u}{2}$).
- We obtain two subintervals $[x_l, x_r]$ and $[x_r, x_u]$.



Bisection Method

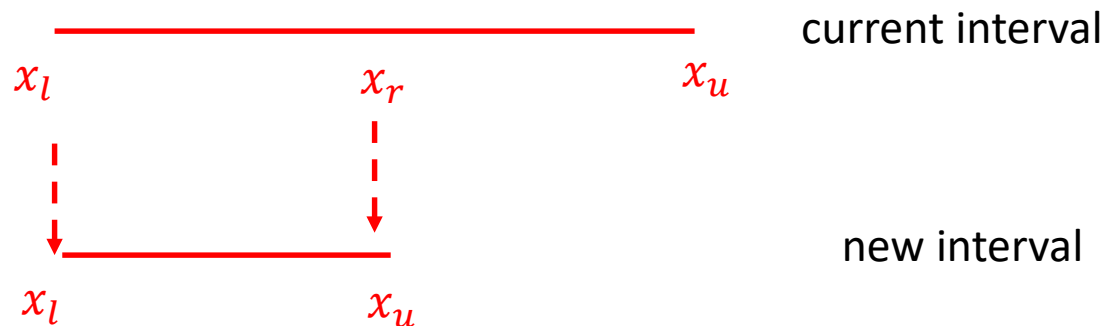
- We obtain two subintervals $[x_l, x_r]$ and $[x_r, x_u]$.
- In the next step, we find the root is in which subinterval.



Bisection Method

Step 2:

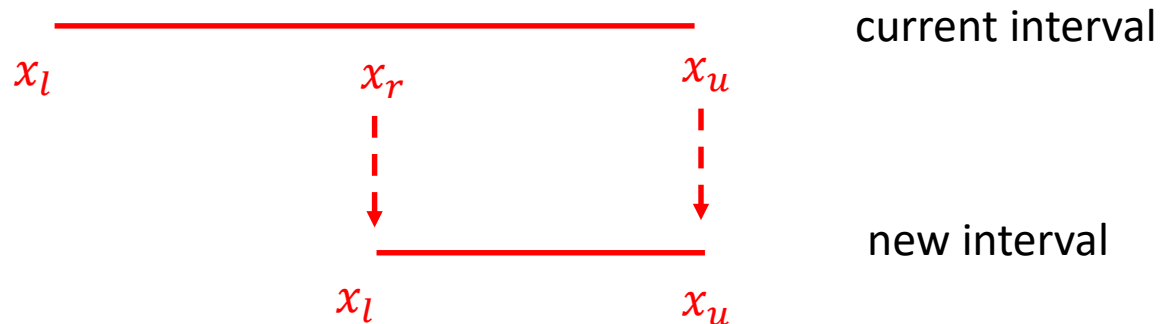
- If $f(x_l)$ and $f(x_r)$ have opposite signs, form a **new interval** as follows.
- The **left endpoint** of the **new interval** is x_l .
- The **right endpoint** of the **new interval** is x_r (set $x_u \leftarrow x_r$).



Bisection Method

Step 2:

- If $f(x_u)$ and $f(x_r)$ have opposite signs, form a new interval as follows.
- The **right endpoint** of the new interval is x_u .
- The **left endpoint** of the new interval is x_r (set $x_l \leftarrow x_r$).



Bisection Method

Step 3:

- If you are in the **first iteration** ($k = 1$), **skip the calculations below** and **go to Step 1**.
- **If not**, compute the **approximate percentage** of relative error using the following formula

$$\epsilon_a = \left| \frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}} \right| 100\%$$

- x_r^{new} is the midpoint in the **current iteration**.
- x_r^{old} is the midpoint in the **last iteration**.
- Stop when $\epsilon_a < \epsilon$, otherwise go to Step 1.

Bisection Method

- Each time that we apply the bisection method, the size of the interval reduces.
- The last generated x_r is the estimate of the root of $f(x)$.

Step 1: Choose lower x_l and upper x_u guesses for the root such that the function changes sign over the interval. This can be checked by ensuring that $f(x_l)f(x_u) < 0$.

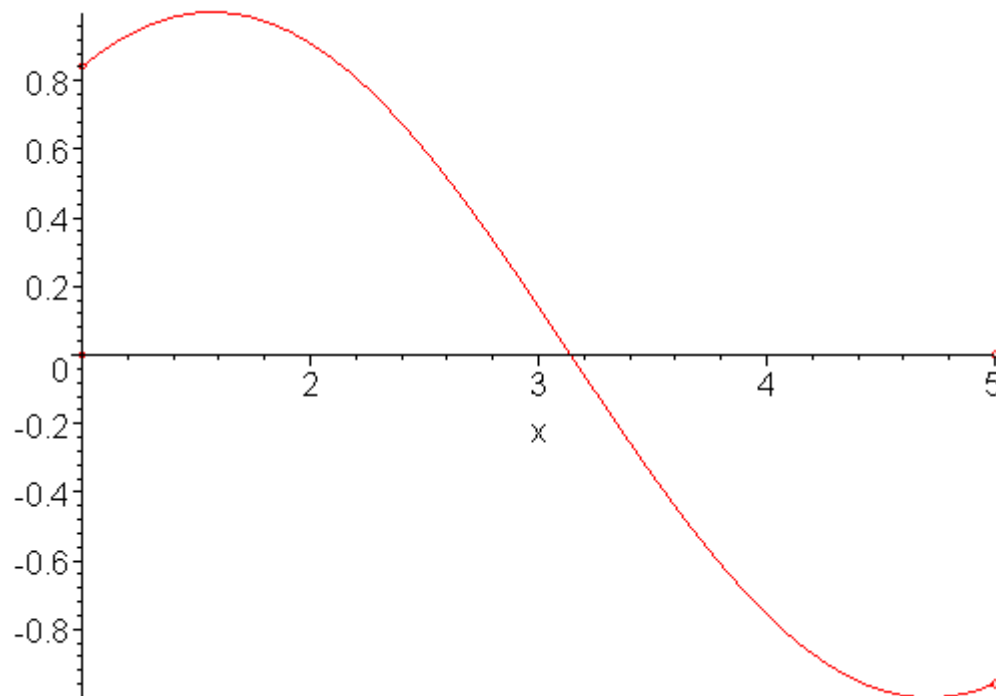
Step 2: An estimate of the root x_r is determined by

$$x_r = \frac{x_l + x_u}{2}$$

Step 3: Make the following evaluations to determine in which subinterval the root lies:

- (a) If $f(x_l)f(x_r) < 0$, the root lies in the lower subinterval. Therefore, set $x_u = x_r$ and return to step 2.
- (b) If $f(x_l)f(x_r) > 0$, the root lies in the upper subinterval. Therefore, set $x_l = x_r$ and return to step 2.

Example



Example

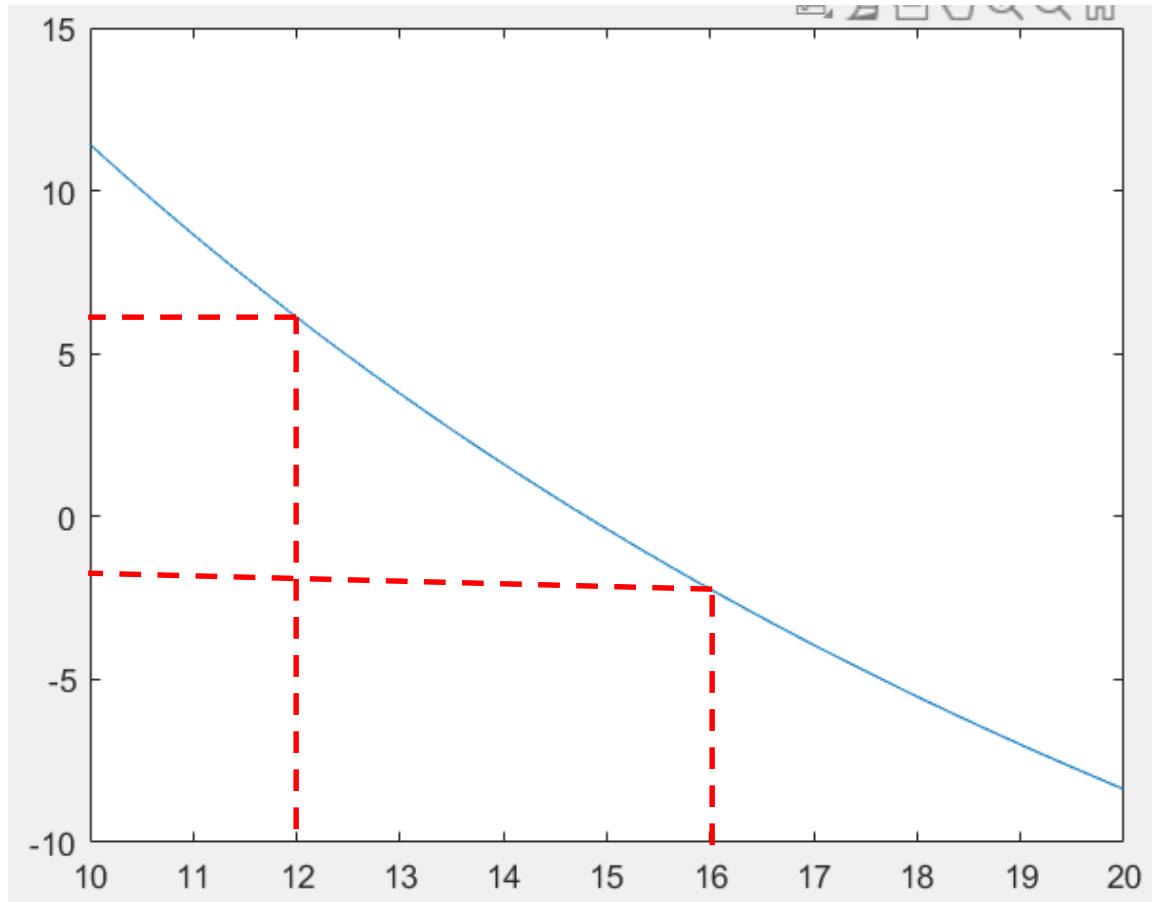
- Use the bisection method to find the root of the function

$$f(x) = \frac{668.06}{x} (1 - e^{-0.146843x}) - 40$$

Continue the iterations until the relative error is less than 0.5%. Start with the interval [12,16].

Example

- The signs of function at the endpoints of the interval $[12,16]$ are different.



Example

- First iteration

x_l	x_r	x_u	$f(x_l)$	$f(x_r)$	$f(x_u)$
12	14	16	6.1139	1.6111	-2.2303

current interval 


new interval

- Second iteration

x_l	x_r	x_u	$f(x_l)$	$f(x_r)$	$f(x_u)$
14	15	16	1.6111	-0.3845	-2.2303

current interval 



new interval

ϵ_a (%)

6.667

$x_r^{\text{new}} = 15$

$x_r^{\text{old}} = 14$

Example

- Third iteration

	x_l	x_r	x_u	$f(x_l)$	$f(x_r)$	$f(x_u)$	
	14	14.5	15	1.6111	0.5936	-0.3845	
current interval				$\epsilon_a (\%)$			
				3.448			
new interval				$x_r^{\text{new}} = 14.5$ $x_r^{\text{old}} = 15$			

- Fourth iteration

	x_l	x_r	x_u	$f(x_l)$	$f(x_r)$	$f(x_u)$	
	14.5	14.75	15	0.5936	0.0998	-0.3845	
current interval				$\epsilon_a (\%)$			
				1.695			
new interval				$x_r^{\text{new}} = 14.75$ $x_r^{\text{old}} = 14.5$			

Example

- Fifth iteration

	x_l	x_r	x_u	$f(x_l)$	$f(x_r)$	$f(x_u)$	
	14.75	14.875	15	0.0998	-0.1435	-0.3845	
current interval							
							
new interval							
				ϵ_a (%)			
				0.840			$x_r^{\text{new}} = 14.875$ $x_r^{\text{old}} = 14.75$

- Sixth iteration

	x_l	x_r	x_u	$f(x_l)$	$f(x_r)$	$f(x_u)$	
	14.75	14.8125	14.875	0.0998	-0.0222	-0.1435	
current interval							
							
new interval							
				ϵ_a (%)			
				0.422			$x_r^{\text{new}} = 14.8125$ $x_r^{\text{old}} = 14.875$

Example

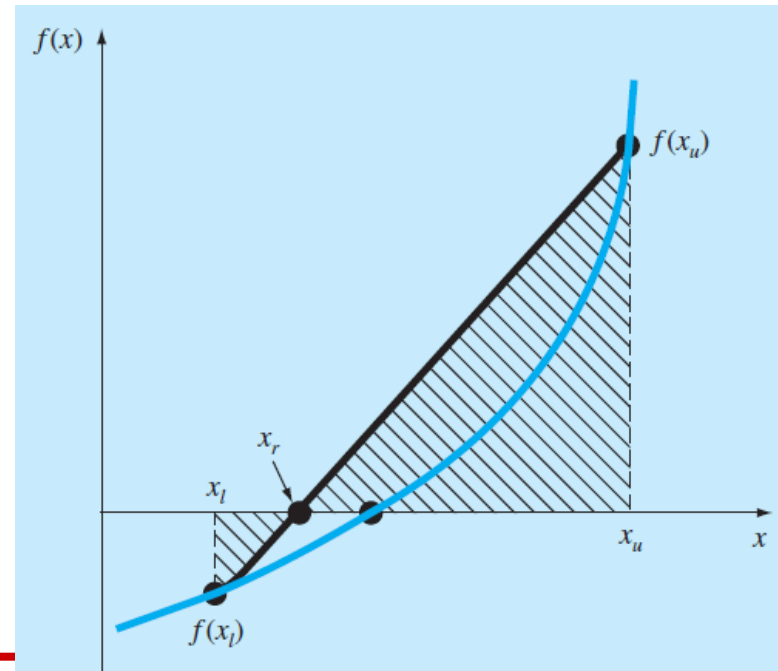
- We stop after 6 iterations.
- The estimate of the root is 14.8125.

Iteration	x_l	x_u	x_r	$f(x_l)$	$f(x_u)$	$f(x_r)$	ϵ_a (%)
1	12	16	14	6.1139	-2.2303	1.6111	
2	14	16	15	1.6111	-2.2303	-0.3845	6.667
3	14	15	14.5	1.6111	-0.3845	0.5936	3.448
4	14.5	15	14.75	0.5936	-0.3845	0.0998	1.695
5	14.75	15	14.875	0.0998	-0.3845	-0.1435	0.840
6	14.75	14.875	14.8125	0.0998	-0.1435	-0.0222	0.422

False-position Method

False-Position Method

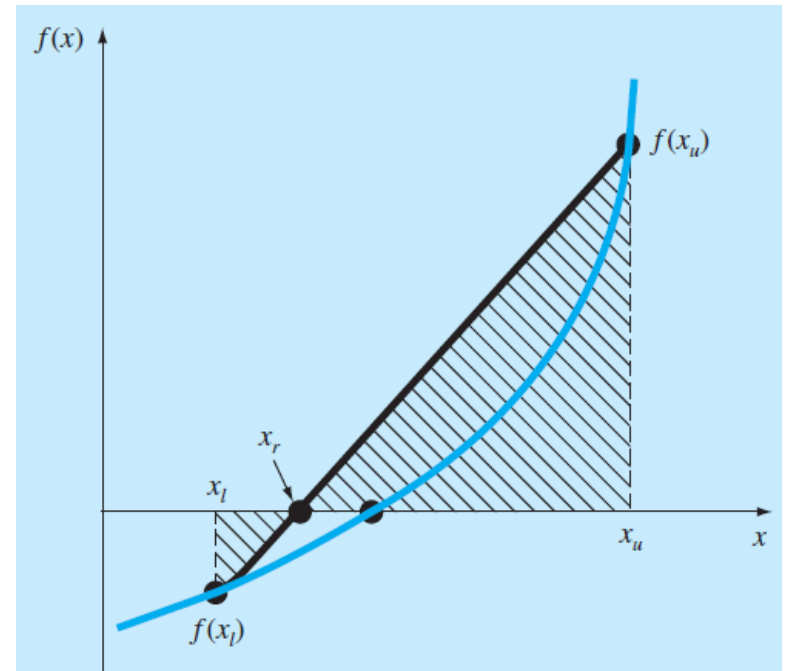
- **Bisection** method is a **brute-force** approach
- In dividing the intervals, **no account** is taken of the **magnitudes** of $f(x_l)$ and $f(x_u)$.
- For example, if $f(x_l)$ is much closer to zero than $f(x_u)$, it is likely that the root is closer to x_l than to x_u .



False-Position Method

- In the false position method, $f(x_l)$ and $f(x_u)$ are joined by a **straight line**.
- The **intersection of this line** with the **x-axis** is selected as x_r .
- The intersection can be calculated using the formula

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$



False-Position Method

- The only difference between the bisection and the false-position methods is in the way that x_r is calculated.
- In the **false-position method**, x_r is calculated using

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

- In the **bisection method**, x_r is calculated using $x_r = \frac{x_l + x_u}{2}$.

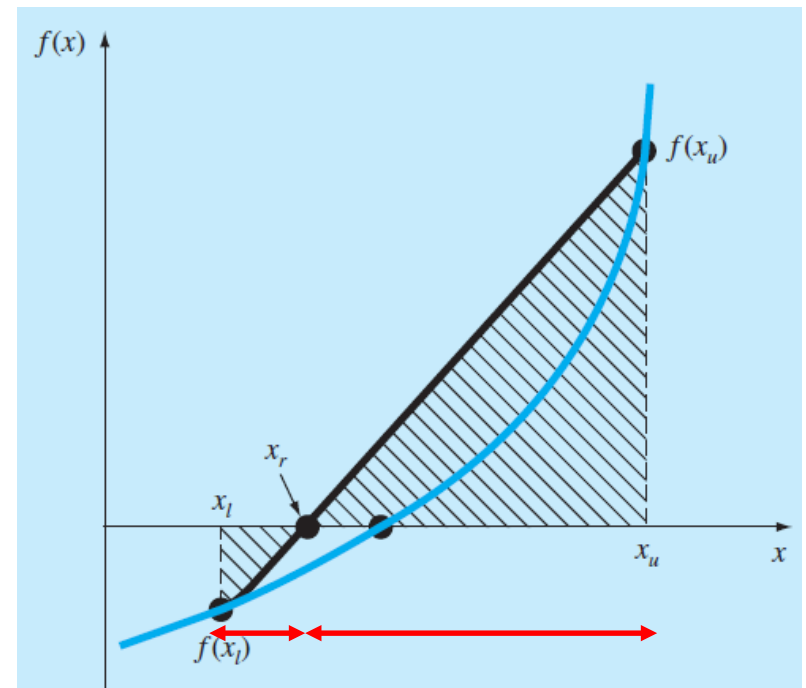
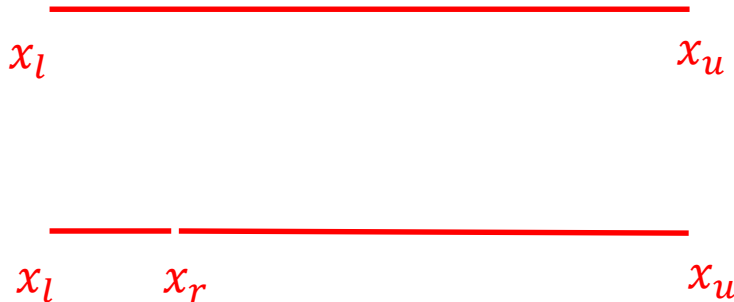
False-Position Method

Step 1:

- Find x_r

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

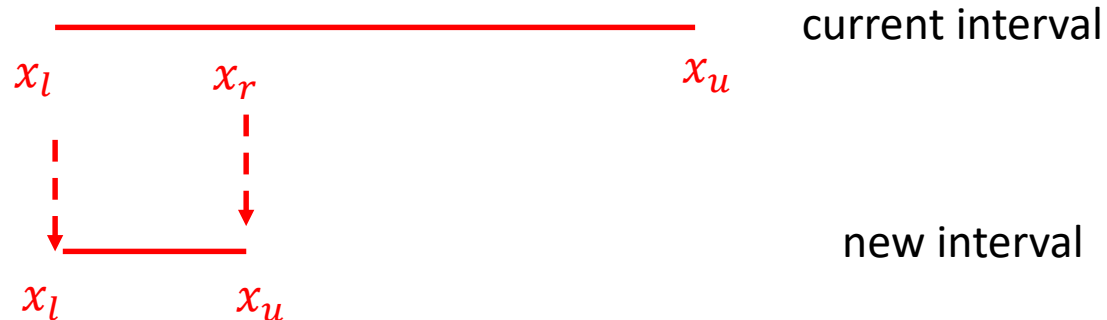
- We obtain two subintervals $[x_l, x_r]$ and $[x_r, x_u]$.



False-Position Method

Step 2:

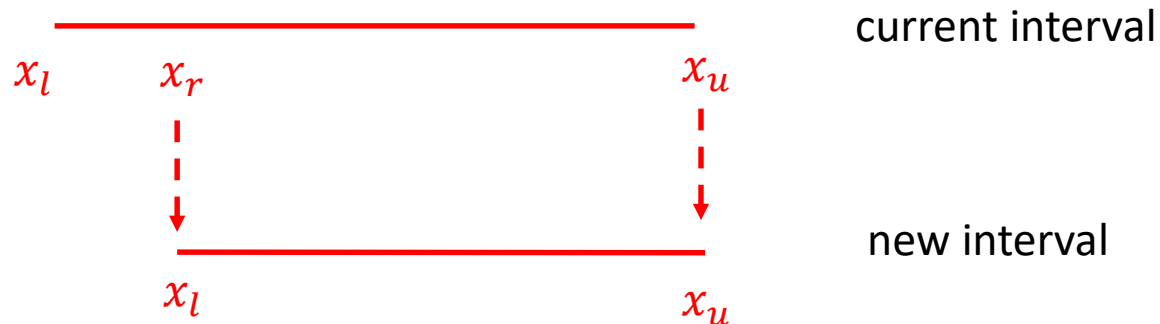
- If $f(x_l)$ and $f(x_r)$ have opposite signs, form a **new interval** as follows.
- The **left endpoint** of the **new interval** is x_l .
- The **right endpoint** of the **new interval** is x_r (set $x_u \leftarrow x_r$).



False-Position Method

Step 2:

- If $f(x_u)$ and $f(x_r)$ have opposite signs, form a new interval as follows.
- The **right endpoint** of the new interval is x_u .
- The **left endpoint** of the new interval is x_r (set $x_l \leftarrow x_r$).



False-Position Method

Step 3:

- If you are in the **first iteration** ($k = 1$), **skip the calculations below** and **go to Step 1**.
- **If not**, compute the **approximate percentage** of relative error using the following formula

$$\epsilon_a = \left| \frac{x_r^{\text{new}} - x_r^{\text{old}}}{x_r^{\text{new}}} \right| 100\%$$

- x_r^{new} is the **intersection** in the **current iteration**.
- x_r^{old} is the **intersection** in the **last iteration**.
- Stop when $\epsilon_a < \epsilon$, otherwise go to Step 1.

False-Position Method

Step 1: Choose lower x_l and upper x_u guesses for the root such that the function changes sign over the interval. This can be checked by ensuring that $f(x_l)f(x_u) < 0$.

Step 2: An estimate of the root x_r is determined by

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

Step 3: Make the following evaluations to determine in which subinterval the root lies:

(a) If $f(x_l)f(x_r) < 0$, the root lies in the lower subinterval. Therefore, set $x_u = x_r$ and return to step 2.

(b) If $f(x_l)f(x_r) > 0$, the root lies in the upper subinterval. Therefore, set $x_l = x_r$ and return to step 2.

Example

- Compute the first two iterations of the false-position method for the function

$$f(x) = \frac{668.06}{x} (1 - e^{-0.146843x}) - 40$$

Start with the interval [12,16].

Example

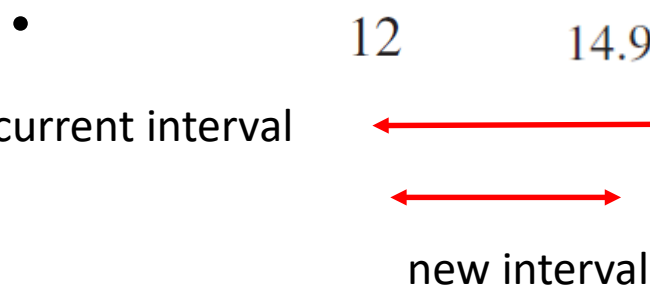
- Iteration 1:

$$x_l = 12 \quad f(x_l) = 6.1139$$

$$x_u = 16 \quad f(x_u) = -2.2303$$

$$x_r = 16 - \frac{-2.2303(12 - 16)}{6.1139 - (-2.2303)} = 14.9309$$

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$



$$f(x_l) = 6.1139$$

$$f(x_r) = -0.251$$

$$f(x_u) = -2.2303$$

- So, we set $x_u = x_r$.
- The new interval becomes $[12, 14.9309]$.

Example

- Iteration 2:

$$x_l = 12 \quad f(x_l) = 6.1139$$

$$x_u = 14.9309 \quad f(x_u) = -0.2515$$

$$x_r = 14.9309 - \frac{-0.2515(12 - 14.9309)}{6.1139 - (-0.2515)} = 14.8151$$

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

$$f(x_l) = 6.1139$$

$$f(x_r) = -0.02714$$

$$f(x_u) = -0.2515$$

current interval

12 14.8151 14.9309

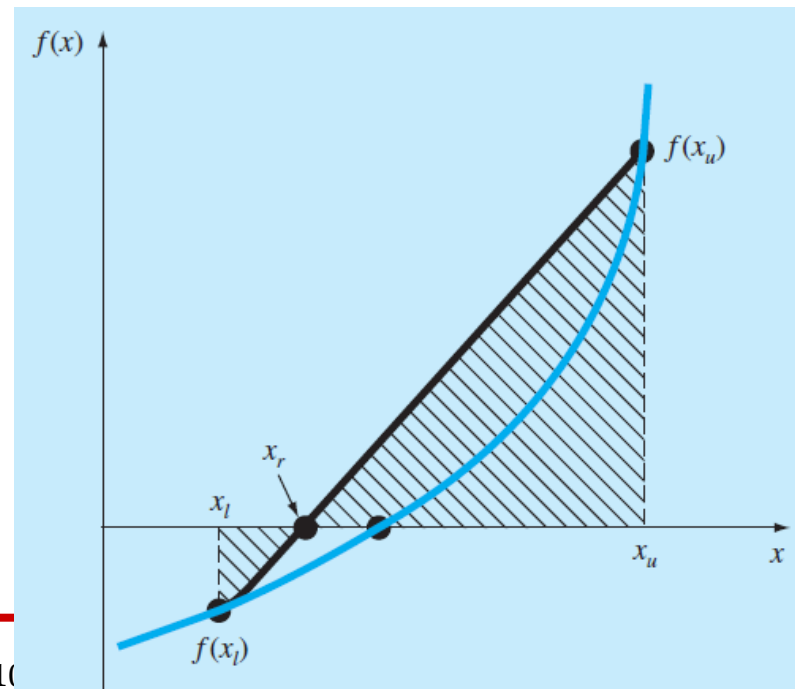


new interval

- So, we set $x_u = x_r$.
- The new interval becomes $[12, 14.8151]$.

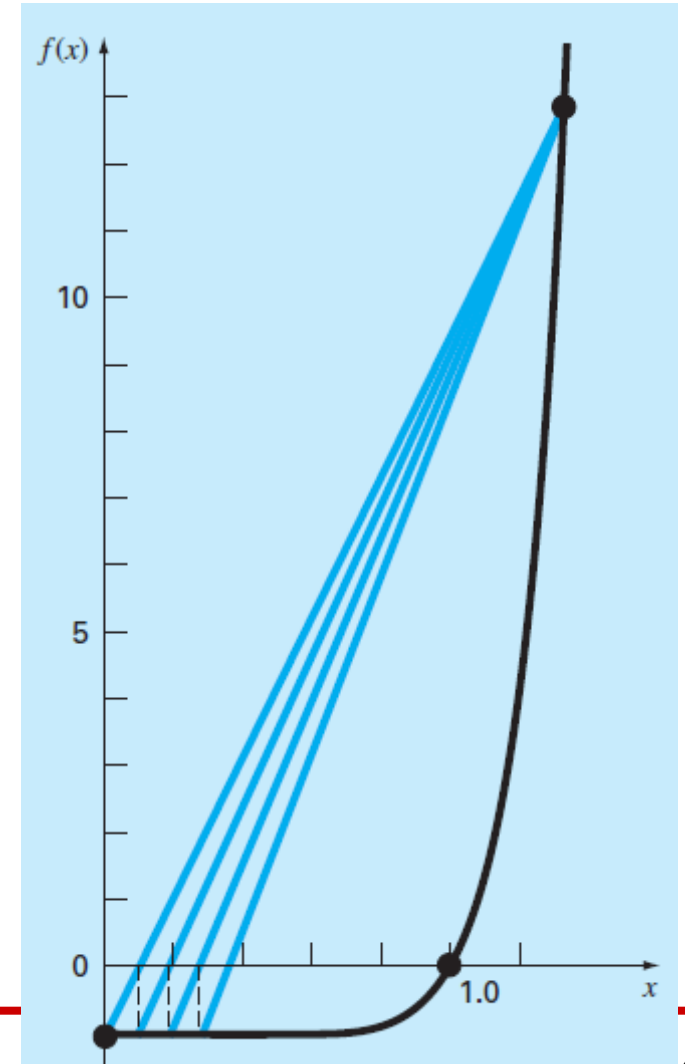
False-Position Method

- The **false-position method** generally finds the root **faster than** the **bisection method**.
- It relies on the **assumption** that if the **value of the function** at an **endpoint** is **close to zero**, the **root is located** close to that **endpoint**.
- This assumption may not be always true.



False-Position Method

- As iterations are proceeding, one of the endpoints (x_l or x_u) will tend to stay fixed.
- This can lead to poor convergence.



Multiple Roots

- A function might have **multiple roots** at **one point**. Such a point is called a **multiple root**.
- The function below has 2 roots at $x = 1$

$$f(x) = (x - 3)(x - 1)(x - 1)$$

- $x = 1$ is called a **multiple root** (or a **double root**) of this function.
- The function below has 3 roots at $x = 1$

$$f(x) = (x - 3)(x - 1)(x - 1)(x - 1)$$

- $x = 1$ is called a **triple root** of this function.

Multiple Roots

- The function below has 4 roots at $x = 1$

$$f(x) = (x - 3)(x - 1)(x - 1)(x - 1)(x - 1)$$

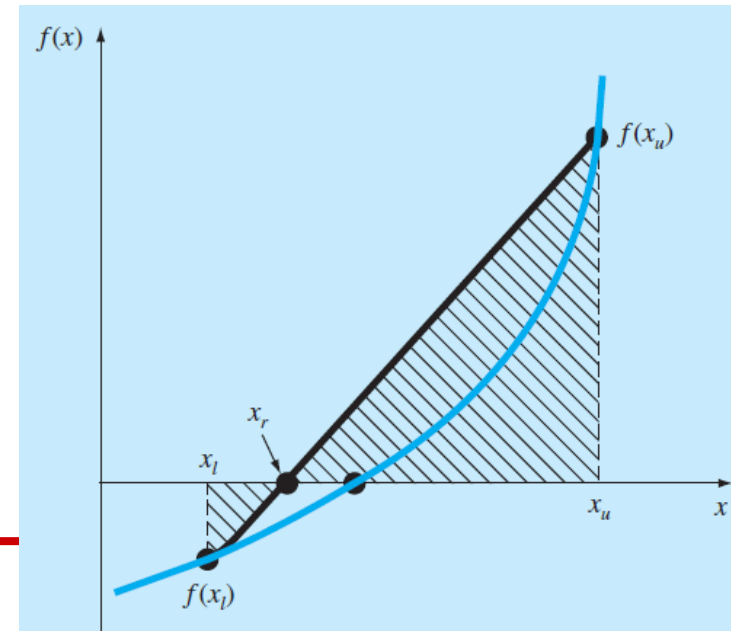
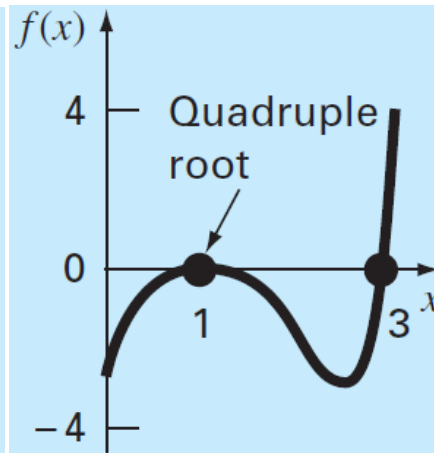
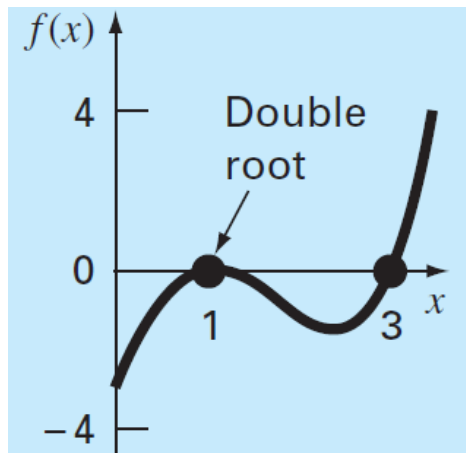
- $x = 1$ is called a **quadruple root** of this function.
- A function has an **even multiple root** at x if it has **an even number of roots** at x .
- Even multiple root** at $x = 1$

$$f(x) = (x - 3)(x - 1)(x - 1)(x - 1)(x - 1)$$

$$f(x) = (x - 3)(x - 1)(x - 1)$$

Even Multiple Roots

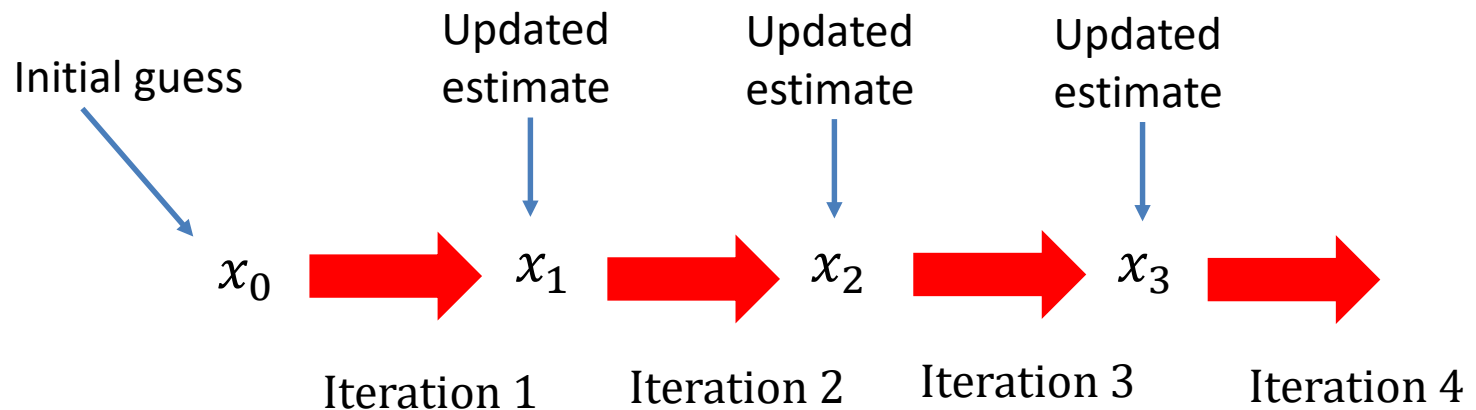
- A **function does** not change signs at **even multiple roots**.
- For the **bisection** and **false-position methods**, $f(x_u)$ and $f(x_l)$ must have **opposite signs**.
- Therefore, these methods cannot be used to find **even multiple roots**.
- We need to use **other methods** for finding them.



Newton-Raphson Method

Newton-Raphson Method

- Newton-Raphson is an **iterative method** for finding roots.
- Suppose that we need to find the root of the function $f(x)$.
- In this method, we start with an **initial guess** for the **root** (x_0).
- At each **iteration**, the **estimate of the root** is updated based on **last estimate**.



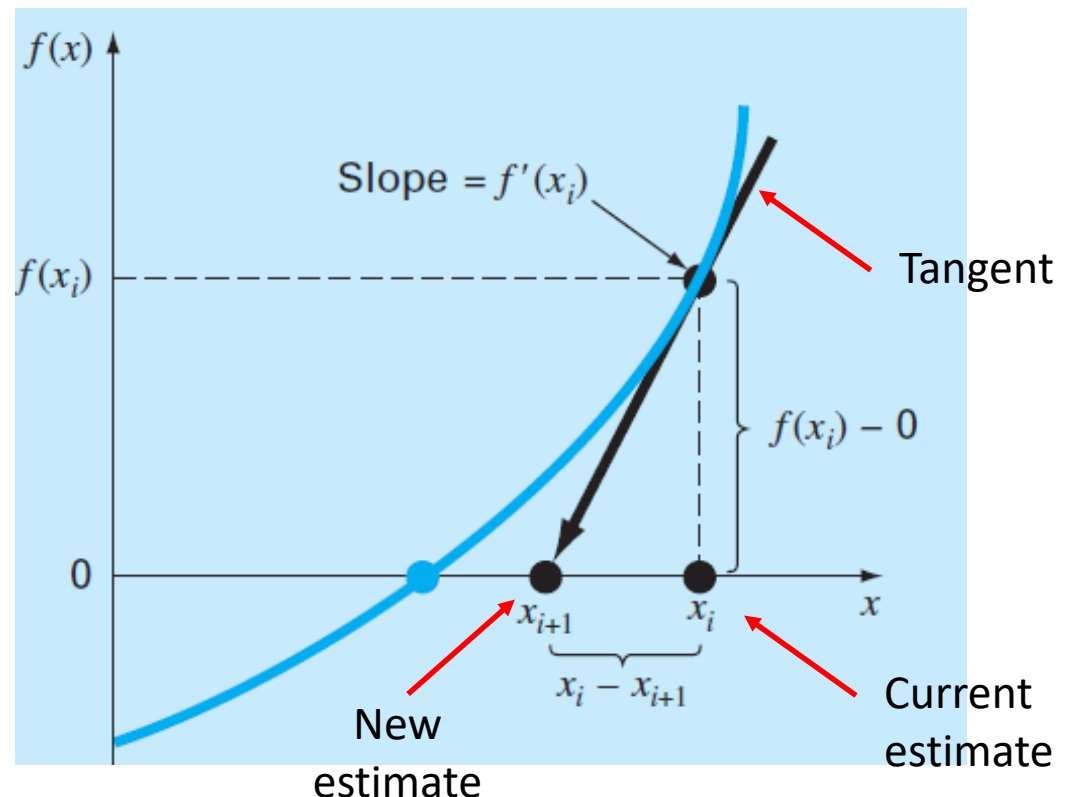
Newton-Raphson Method

- To **update the estimate of the root**, at each iteration, we find the **tangent line** to the **function $f(x)$** at the **current estimate**.
- The **point** where this **tangent crosses the x -axis** is the **new estimate of the root**.

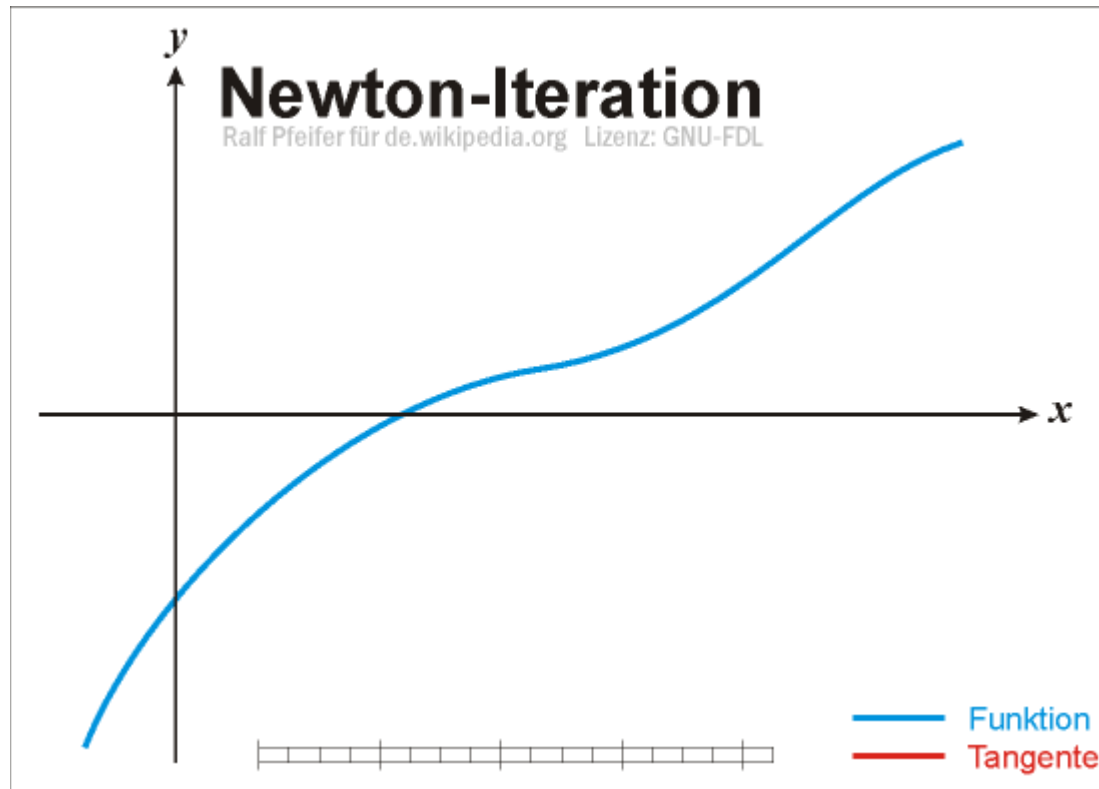
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

New estimate

Current estimate



Newton-Raphson Method



Newton-Raphson Method

1. $k = 0$ (iteration counter).
 2. x_0 : initial guess (initial estimate)
 3. Update the estimate using the equation $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$
 4. Compute the approximate percentage of the relative error
$$\epsilon_{k+1} = \left| \frac{x_{k+1} - x_k}{x_{k+1}} \right| \times 100\%$$
 5. Stop if ϵ_{k+1} is less than the desired error level ϵ .
 6. Otherwise, $k \leftarrow k + 1$ and go to step 3.
- The last generated x_k is (the estimate of) the root of the function $f(x)$.

Example

- Compute four iterations of the **Newton-Raphson** for the function $f(x) = e^{-x} - x$. Use $x_0 = 0$ as the **initial guess**.

Example

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

- The first derivative of $f(x) = e^{-x} - x$ is

$$f'(x) = -e^{-x} - 1$$

- The estimate of the root is updated using the equation

$$x_{i+1} = x_i - \frac{e^{-x_i} - x_i}{-e^{-x_i} - 1}$$

Iteration 1	$x_0 = 0$		$x_1 = 0.5000000000$	$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$
Iteration 2	$x_1 = 0.5000000000$		$x_2 = 0.566311003$	$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$
Iteration 3	$x_2 = 0.566311003$		$x_3 = 0.567143165$	$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$
Iteration 4	$x_3 = 0.567143165$		$x_4 = 0.567143290$	$x_4 = x_3 - \frac{f(x_3)}{f'(x_3)}$

Example

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

- The first derivative of $f(x) = e^{-x} - x$ is

$$f'(x) = -e^{-x} - 1$$

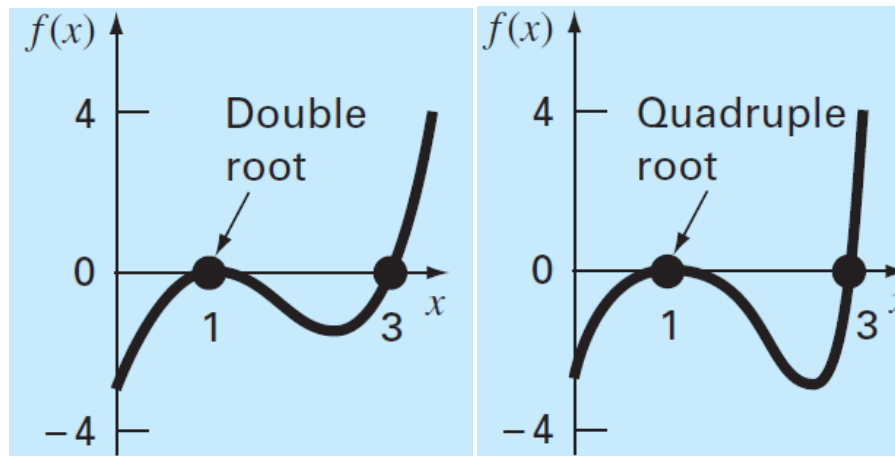
- The estimate of the root is updated using the equation

$$x_{i+1} = x_i - \frac{e^{-x_i} - x_i}{-e^{-x_i} - 1}$$

i	x_i	$\varepsilon_t(\%)$
0	0	100
1	0.500000000	11.8
2	0.566311003	0.147
3	0.567143165	0.0000220
4	0.567143290	$< 10^{-8}$

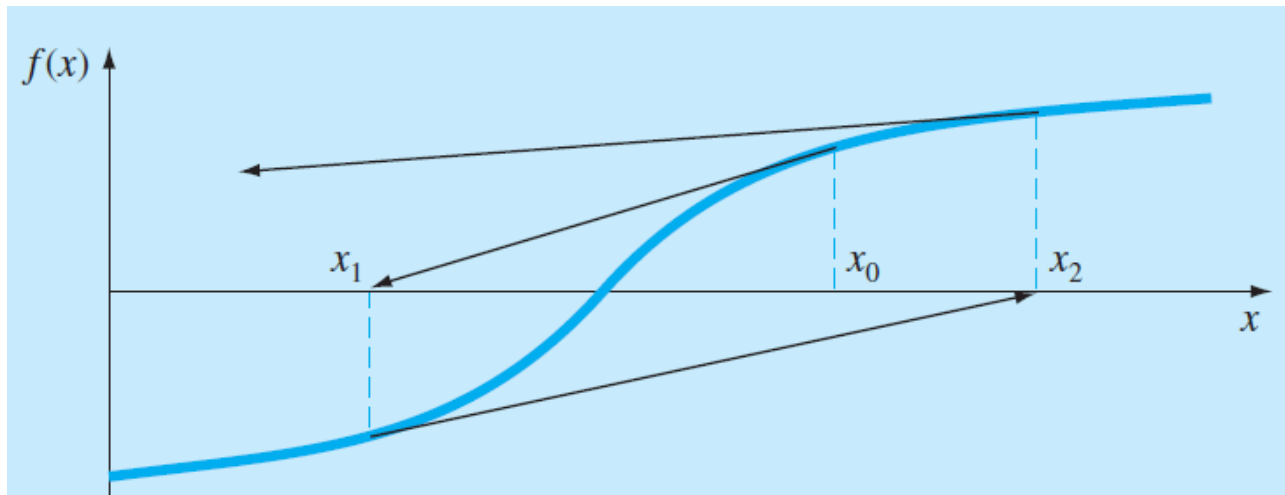
Newton-Raphson Method

- The **Newton-Raphson** method only requires **one initial guess** (x_0).
- The **false-position** and **bisection** methods require x_l and x_u .
- The **false-position** and **bisection** methods require $f(x_l)$ and $f(x_u)$ to have **opposite signs**.
- **Newton-Raphson** usually finds the root very **fast**.



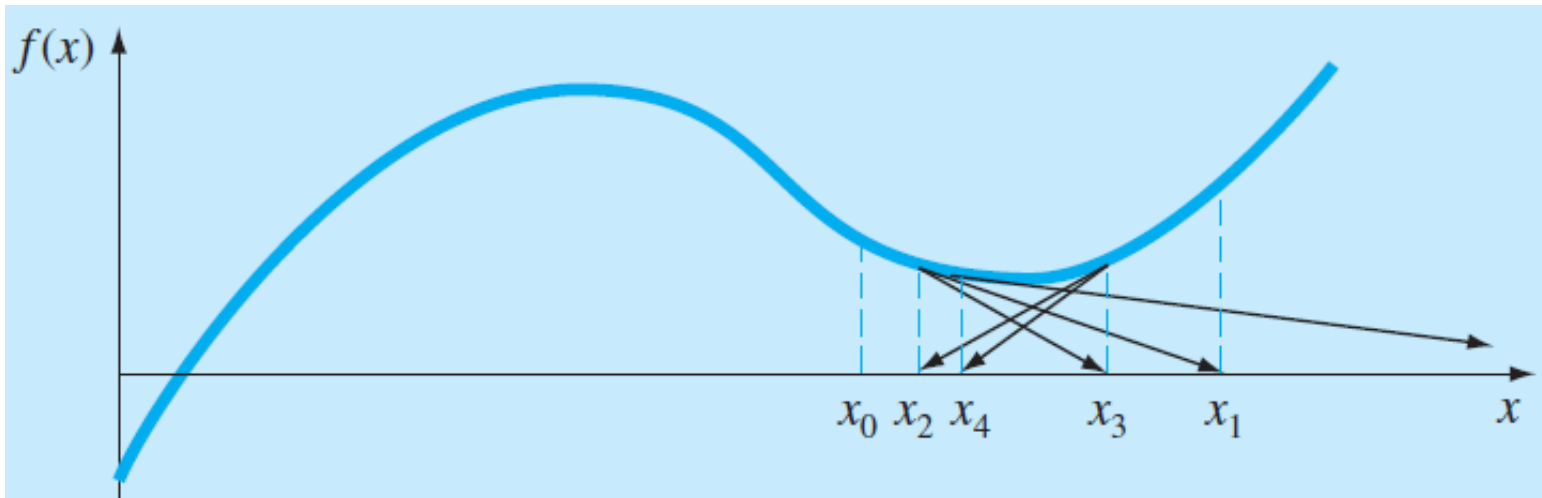
Newton-Raphson Method

- In certain cases, it **may not find the root**.
- For example, when there is an **inflection point close to a root**.
- The **point x** is an **inflection point** if $f''(x) = 0$.



Newton-Raphson Method

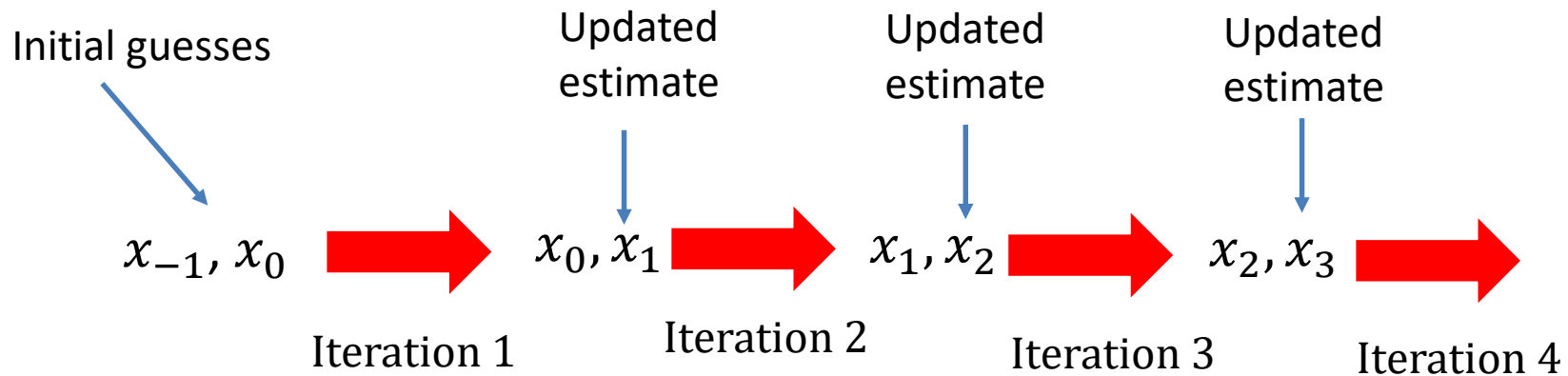
- The **Newton-Raphson** method tends to **oscillate** around the points where $f'(x)$ is close to zero.
- The only remedy is to have an **initial guess** that is “**sufficiently**” close to the **root**.



Secant Method

Secant Method

- Secant is an **iterative method** for finding roots.
- It is very similar to the Newton-Raphson method.
- In this method, we start with two **initial guesses** (x_0, x_{-1}).
- At each **iteration**, the **estimate of the root** is updated.



Secant Method

- A **potential problem** in implementing the **Newton-Raphson method** is the **evaluation of the derivative**.

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

- For **certain functions**, **derivatives** may be **extremely difficult** or **inconvenient** to evaluate.
- In the **secant method**, the derivative is **estimated numerically**.
- In such cases, the **derivative** can be **numerically estimated** as follows

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

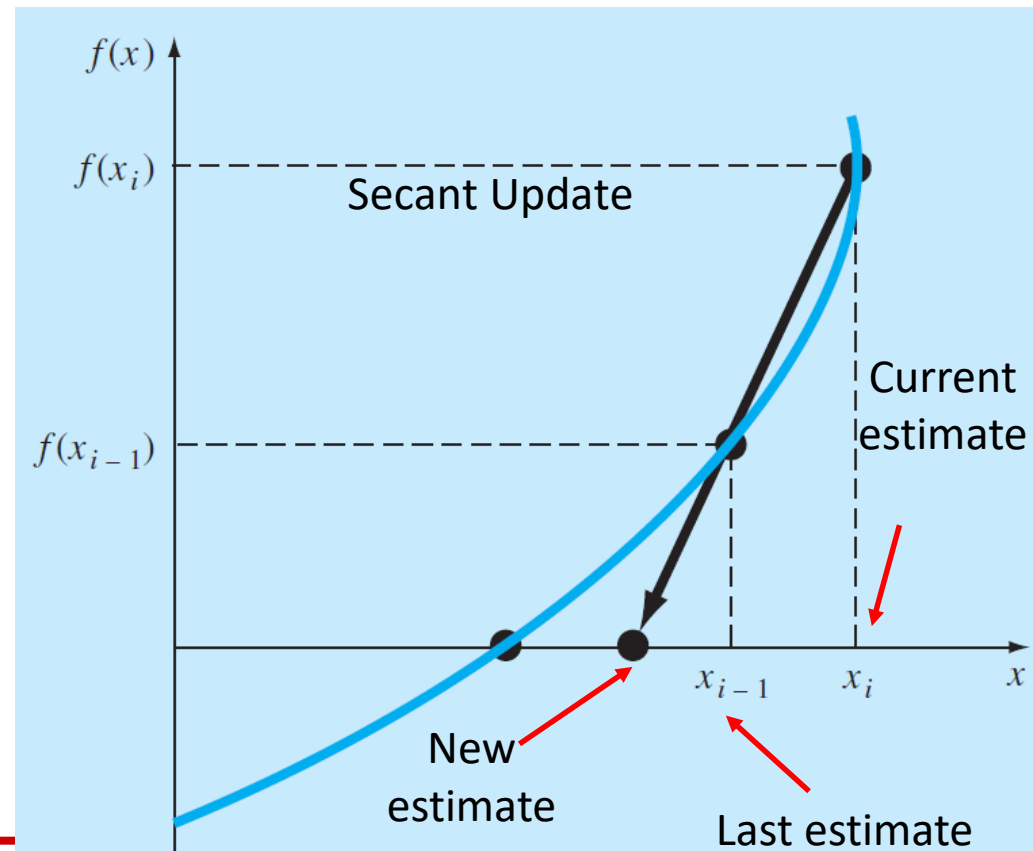
Secant Method

- Consider the line that connects $(x_k, f(x_k))$ and $(x_{k-1}, f(x_{k-1}))$.
- In the secant method, the intersection of this line with the x -axis is the updated estimate of the root.

- Secant update rule

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

- x_k is the **current estimate** of the **root**.
- x_{k-1} is the **estimate** of the **root** at the **last iteration**.

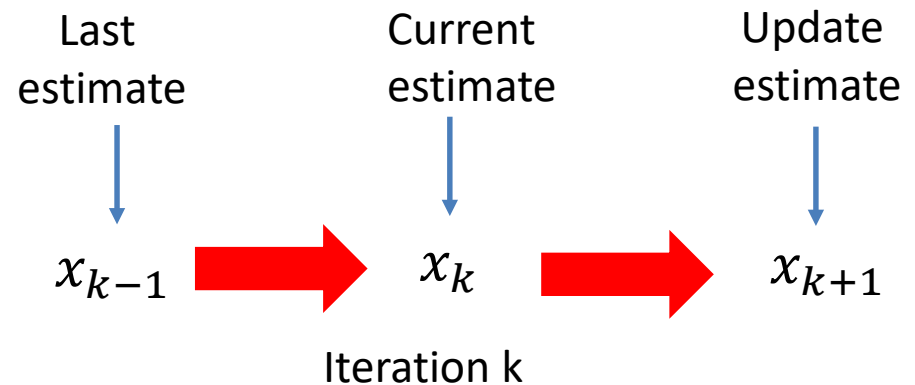


Secant Method

- The secant method needs to **initial values x_0 and x_{-1}** to estimate the derivative at x_0 .
- In the first iteration, the update rule becomes

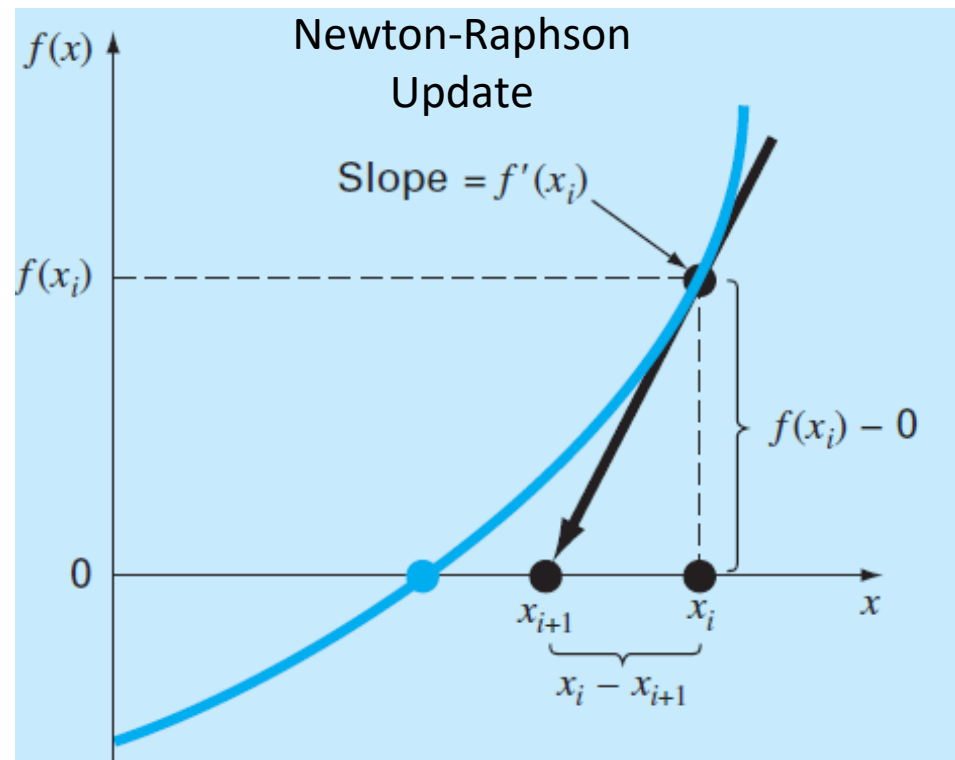
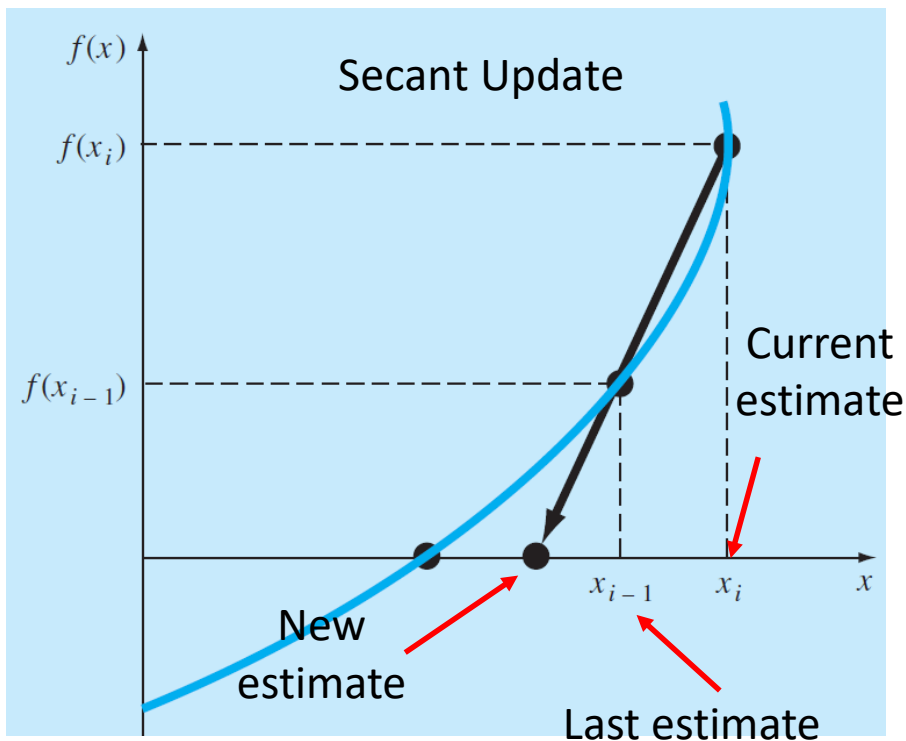
$$x_1 = x_0 - \frac{f(x_0)(x_0 - x_{-1})}{f(x_0) - f(x_{-1})}$$

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



Secant VS Newton-Raphson

- Consider the line that connects $(x_k, f(x_k))$ and $(x_{k-1}, f(x_{k-1}))$.
- In the secant method, the intersection of this line with the x axis is the updated estimate of root.



Secant Method

1. $k = 0$ (iteration counter).
 2. x_0, x_{-1} : initial guesses
 3. Update the estimate using the equation $x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$
 4. Compute the approximate percentage of the relative error
$$\epsilon_{k+1} = \left| \frac{x_{k+1} - x_k}{x_{k+1}} \right| \times 100\%$$
 5. Stop if ϵ_{k+1} is less than the desired error level ϵ .
 6. Otherwise, $k \leftarrow k + 1$ and go to step 3.
- The last generated x_k is the root of the function $f(x)$.

Example

- Compute three iterations of the Secant method for the function $f(x) = e^{-x} - x$. Use $x_0 = 1$ and $x_{-1} = 0$.

Example

- First iteration ($k = 0$)

$$x_1 = x_0 - \frac{f(x_0)(x_0 - x_{-1})}{f(x_0) - f(x_{-1})}$$

$$x_{-1} = 0 \quad f(x_{-1}) = 1.00000$$

$$x_0 = 1 \quad f(x_0) = -0.63212$$

$$x_1 = 1 - \frac{-0.63212(0 - 1)}{1 - (-0.63212)} = 0.61270$$

- Second iteration ($k = 1$)

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}$$

$$x_0 = 1 \quad f(x_0) = -0.63212$$

$$x_1 = 0.61270 \quad f(x_1) = -0.07081$$

$$x_2 = 0.61270 - \frac{-0.07081(1 - 0.61270)}{-0.63212 - (-0.07081)} = 0.56384$$

Example

- Third iteration ($k = 2$)

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

$$x_1 = 0.61270 \quad f(x_1) = -0.07081$$

$$x_2 = 0.56384 \quad f(x_2) = 0.00518$$

$$x_3 = 0.56384 - \frac{0.00518(0.61270 - 0.56384)}{-0.07081 - (0.00518)} = 0.56717$$

END

See you next week 😊!