Numerical Analysis

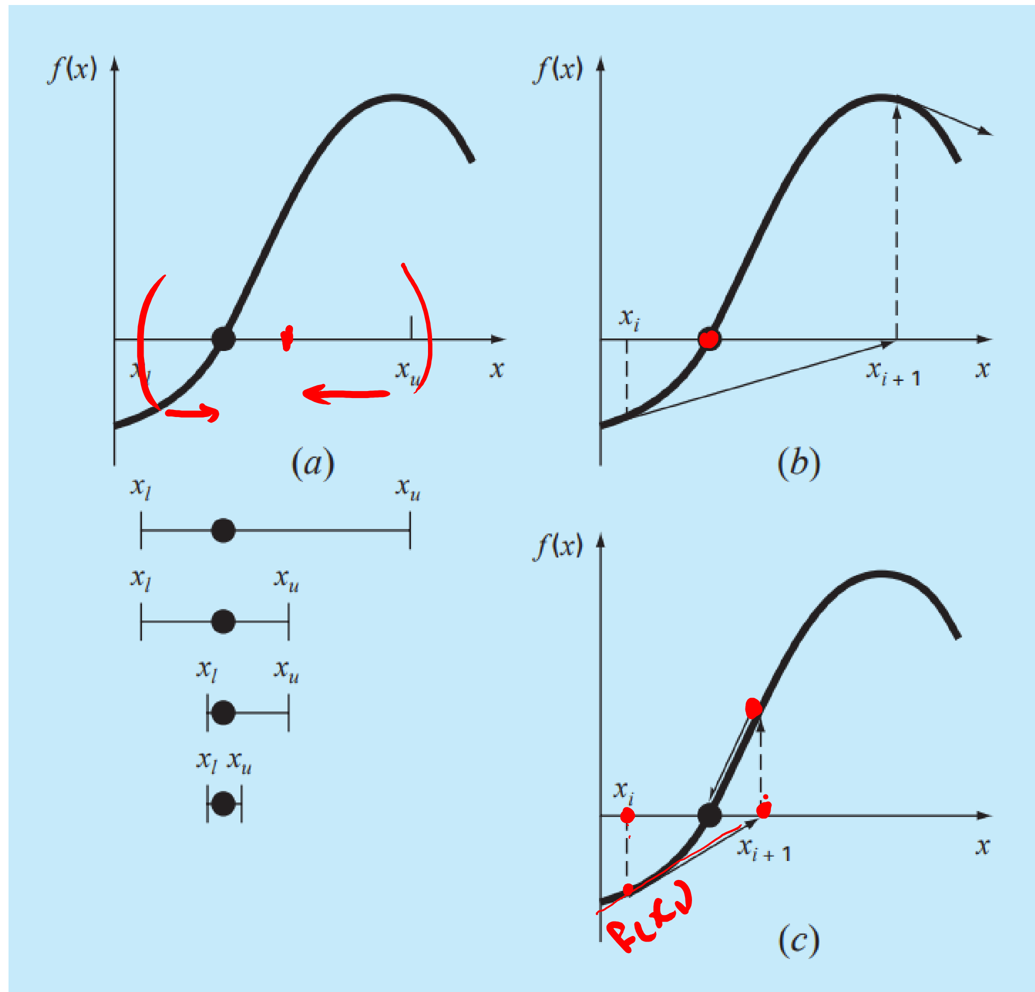
Prof. Osama Abdel Raouf

Solving non-linear equation using numerical methods

- **Bracketing Methods: (Figure a)**
  - The Bisection Method
- **Open Methods (Figure b, c)**
  - Simple Fixed-Point Iteration
  - The Newton-Raphson Method
  - The Secant Method

$x^2-4=0$

$f(x)=0,$



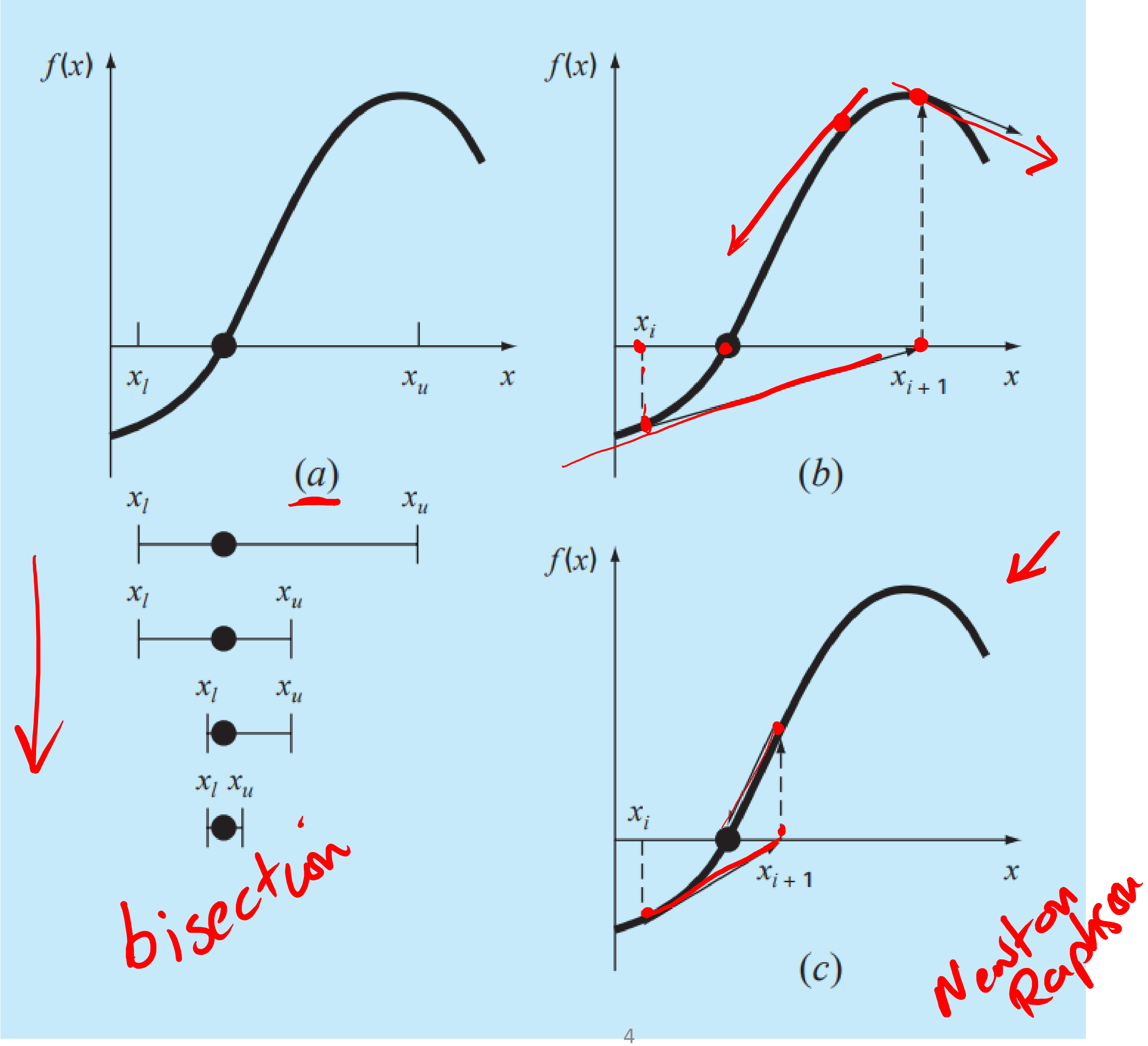
Open Method

- For the bracketing methods, the root is located within an interval prescribed by a lower and an upper bound.
- Repeated application of these methods always results in closer estimates of the true value of the root.
- Such methods are said to be convergent because they move closer to the truth as the computation progresses.
- **In contrast, the open methods are based on formulas that require only a** single starting value of x **or** two starting values that do not necessarily bracket **the root.**

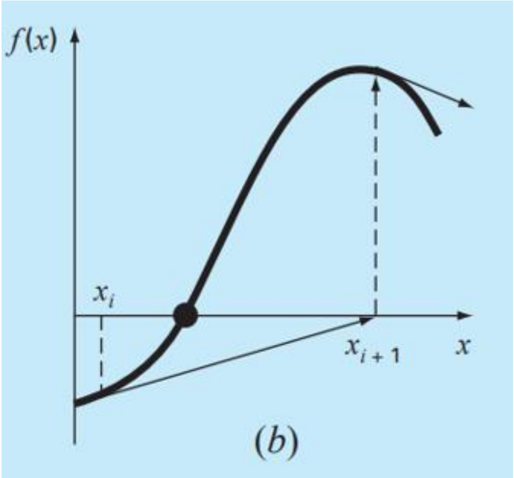
the fundamental difference between the bracketing and open methods for root location

- (a) which is the bisection method, the root is constrained within the interval prescribed by xl and xu

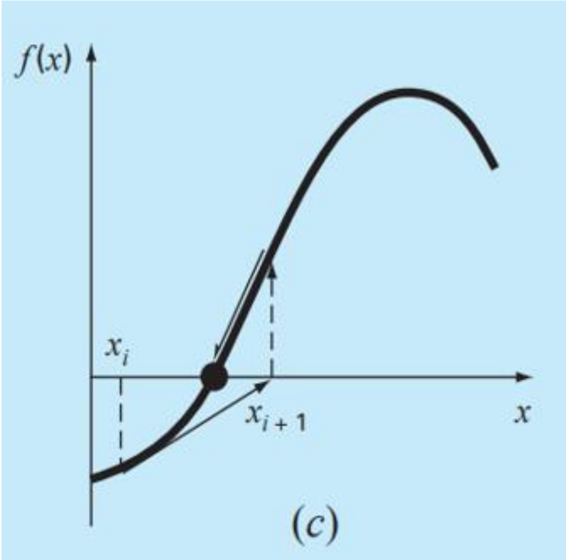
- In contrast, for the open method depicted in (b) and (c), a formula is used to project from xi to xi+1 in an iterative fashion.
- Thus, the method can **either (b) diverge** or **(c) converge rapidly**, depending on the value of the initial guess.



- As noted, **open methods** sometimes **diverge** or **move away from the true root**.



- However, when the open methods **converge**, they usually do so much **more quickly than the bracketing methods**.



SIMPLE FIXED-POINT ITERATION
one-point iteration or successive substitution

- By rearranging the function $f(x) = 0$ so that x is on the left-hand side of the equation:
  - $\boxed{x = g(x)}$
- *Example:*

$x^2 + 3 = 2x$

$\frac{x^2+3}{2} = x$?

- $x^2 - 2x + 3 = 0$
  - $\rightarrow x = \frac{x^2+3}{2}$

(algebraic manipulation)

- $\sin(x) = 0$
  - $\rightarrow x = \sin(x) + x$ (adding x to both sides of the original equation)

$x - 2 = 0$

$x = 2$

Root

SIMPLE FIXED-POINT ITERATION
one-point iteration or successive substitution

a	b	x_r

- $x = g(x)$
  - provides a formula to predict a new value of $\boldsymbol{x}$ **as a function of an <u>old</u> value of** $\boldsymbol{x}$.

  - Thus, given an initial guess at the root $\boldsymbol{x_i}$, it is used to compute a new estimate $\boldsymbol{x_{i+1}}$:

$$\boldsymbol{x_{i+1} = g(x_i)}$$

SIMPLE FIXED-POINT ITERATION
one-point iteration or successive substitution

$e^{-x} - x = 0$

- Problem Statement. Use simple fixed-point iteration to locate the root of $f(x) = e^{-x} - x$.
- Solution:

$$x = g(x)$$

$$x = e^{-x} - x + x$$

$$\boxed{x = e^{-x}}$$

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

Starting with an initial guess of $x_0 = 0$

$\frac{\text{curr} - \text{old}}{\text{curr}} \times 100$

i	x_i	ε_a (%)	ε_t (%)
0	0		100.0
1	1.000000	100.0	76.3
2	0.367879	171.8	35.1
3	0.692201	46.9	22.1
4	0.500473	38.3	11.8
5	0.606244	17.4	6.89
6	0.545396	11.2	3.83
7	0.579612	5.90	2.20
8	0.560115	3.48	1.24
9	0.571143	1.93	0.705
10	$\boxed{0.564879}$	1.11	0.399

$x_0 = 0$

$x_1 = e^{-0} = 1$

$x_2 = e^{-1} = 0.367879$

$x_3 = e^{-}$

$e^{-0.564879} - 0.564 \overset{?}{=} 0$

Thus, each iteration brings the estimate closer to the true value of the root: 0.56714329.

Convergence

- Notice that the true percent relative error for each iteration of the previous example is roughly proportional (by a factor of about 0.5 to 0.6) to the error from the previous iteration.
- This property, called linear convergence, is characteristic of fixed-point iteration.

i	x_i	ε_a (%) $= \frac{cur - prev}{curr}$	ε_t (%) $= \frac{true - es}{true}$
0	0		100.0
1	1.000000	100.0	76.3
2	0.367879	171.8	35.1
3	0.692201	46.9	22.1
4	0.500473	38.3	11.8
5	0.606244	17.4	6.89
6	0.545396	11.2	3.83
7	0.579612	5.90	2.20
8	0.560115	3.48	1.24
9	0.571143	1.93	0.705
10	0.564879	1.11	0.399

Graphical Representation of a function
-to visualize its structure and behavior

- $f(x) = 0$ can be represented by two ways in order to find its root
  1. Draw $f(x)$
     - The x values corresponding to the intersections of this function with the x-axis represents the roots of $f(x) = 0$. (I.e., at $y = 0$)

  2. Separate $f(x)$ into two component parts, as in $f_1(x) = f_2(x)$.
     - Then the two equations $y_1 = f_1(x)\ and\ y_2 = f_2(x)$ can be plotted separately.
     - The x values corresponding to the intersections of these functions represent the roots of $f(x) = 0$.
- **The concepts of convergence and divergence can be depicted graphically using one of them.**
- **See the figures on the next slide**

$e^{-x} - x \Rightarrow$

Problem Statement.

Let the equation $f(x) = e^{-x} - x = 0$, finds graphically the roots of $f(x)$ by drawing $f(x)$ or by separating it into two parts

Solution:

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

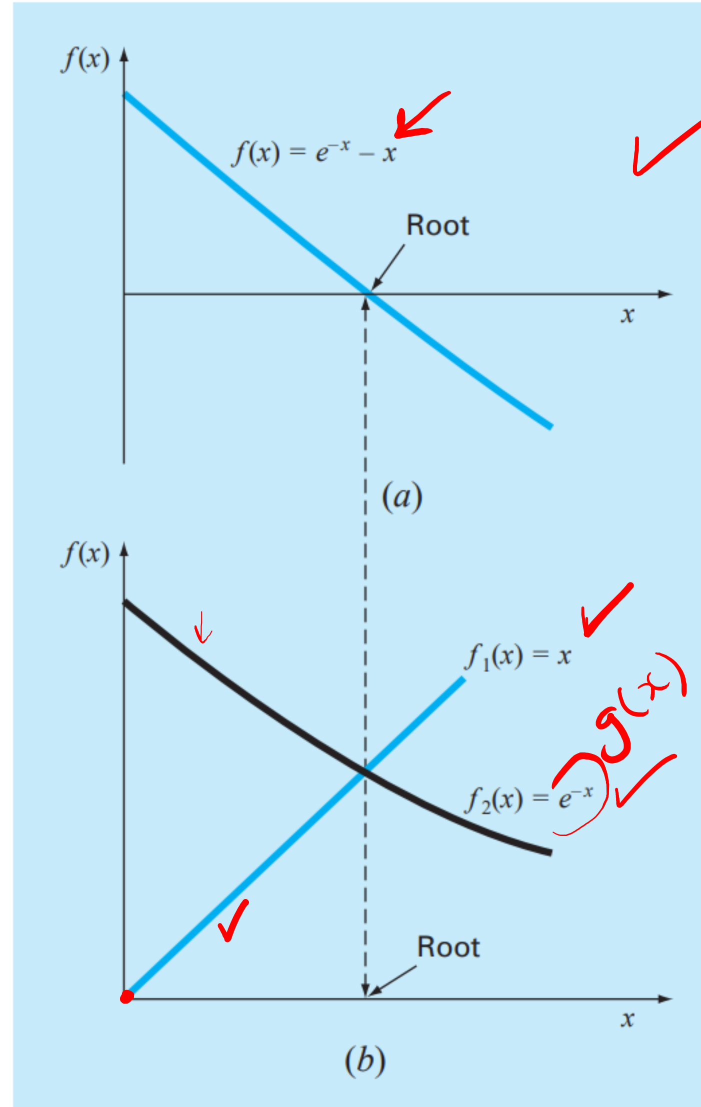
2. $y_1 = x$ and $y_2 = e^{-x}$. (Figure b)

x	$f(x)$
0.0	1.0000
0.2	0.6187
0.4	0.2703
0.6	0.0512-
0.8	0.3507-
1.0	0.6321-

$y_1 = x$

x	y_1	y_2
0.0	0.0	1.000
0.2	0.2	0.819
0.4	0.4	0.670
0.6	0.6	0.549
0.8	0.8	0.449
1.0	1.0	0.368

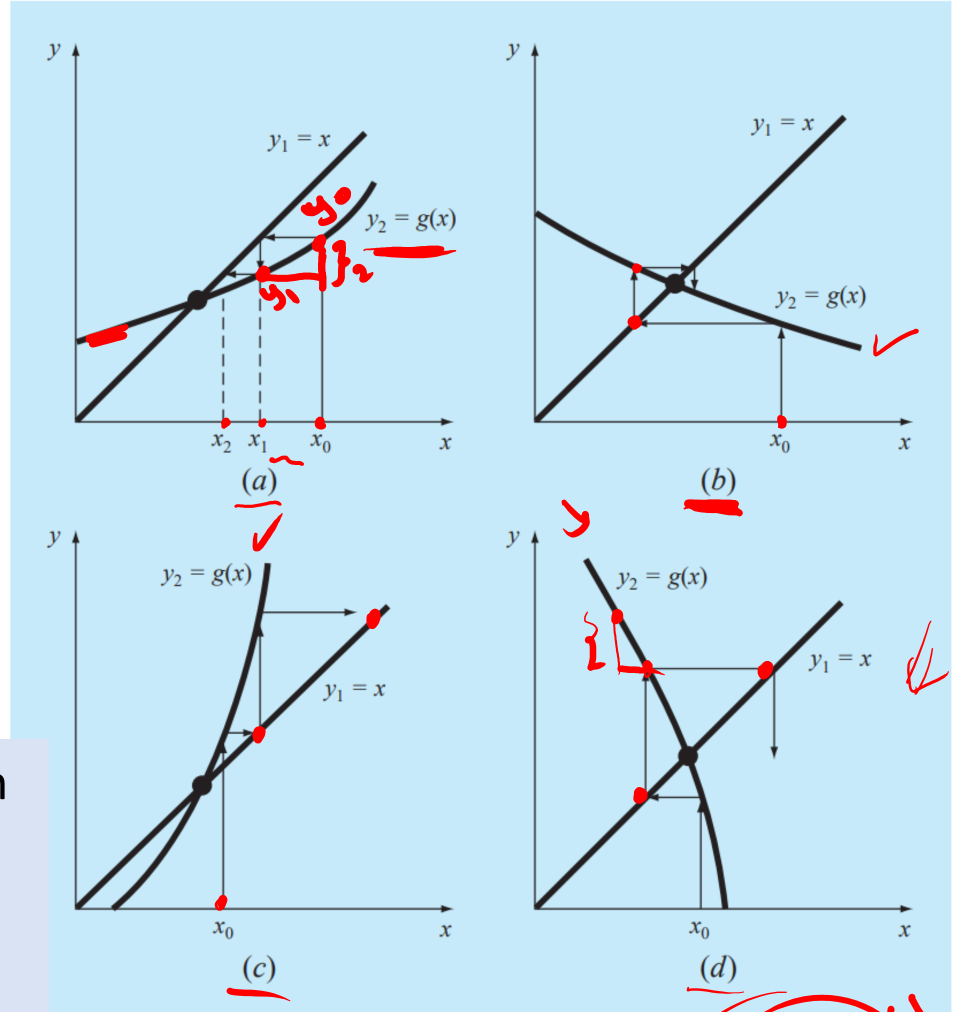
x e^{-x}



$y_1 = x$
1

Convergence and Divergence of Simple Fixed Pont Iteration

- (a) and (b) convergence and
- (c) and (d) divergence of simple fixed-point iteration.
- Graphs (a) and (c) are called monotone patterns,
- whereas (b) and (d) are called oscillating or spiral patterns.



it should be clear that fixed-point iteration converges if, in the region of interest, $|g'(x)| < 1$.
In other words, convergence occurs if the magnitude of the slope of g(x) is less than the slope of the line f(x) = x .
This observation can be demonstrated theoretically. (see book)

Slope of y_1=1

$\frac{y_2 - y_1}{x_2 - x_1} < 1$

$y_2 - y_1 < x_2 - x_1$

Fixed Point Algorithm

```
FUNCTION Fixpt(x0, es, imax, iter, ea)
  xr = x0
  iter = 0
  DO
    xrold = xr
    xr = g(xrold)
    iter = iter + 1
    IF xr ≠ 0 THEN
```

$$ea = \left| \frac{xr - xrold}{xr} \right| \cdot 100$$

```
    END IF
  IF ea < es OR iter ≥ imax EXIT
  END DO
  Fixpt = xr
END Fixpt
```

max iterations

iterations

estimated error

acceptable error

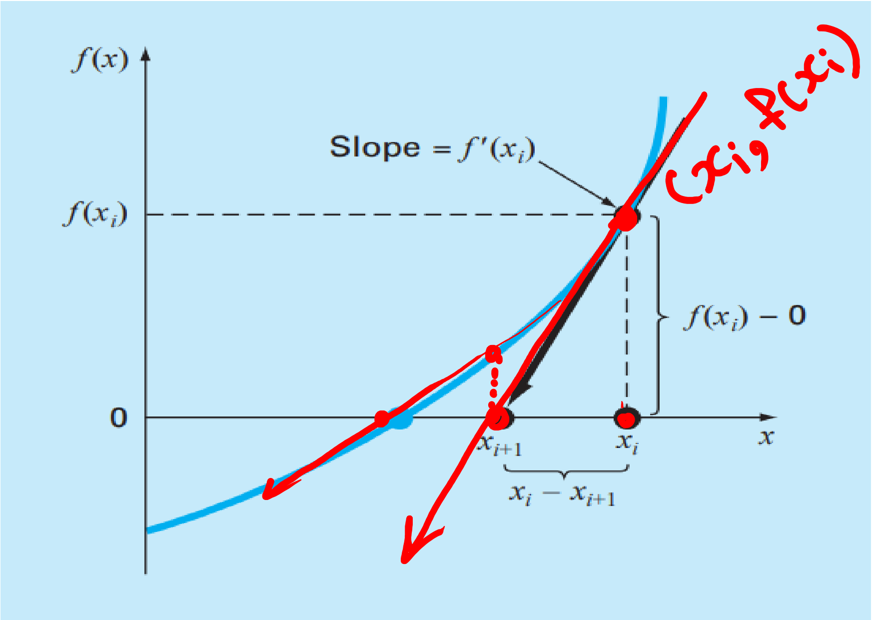
bisection ✗ bracketing method

Fixed-Point ✓ open method

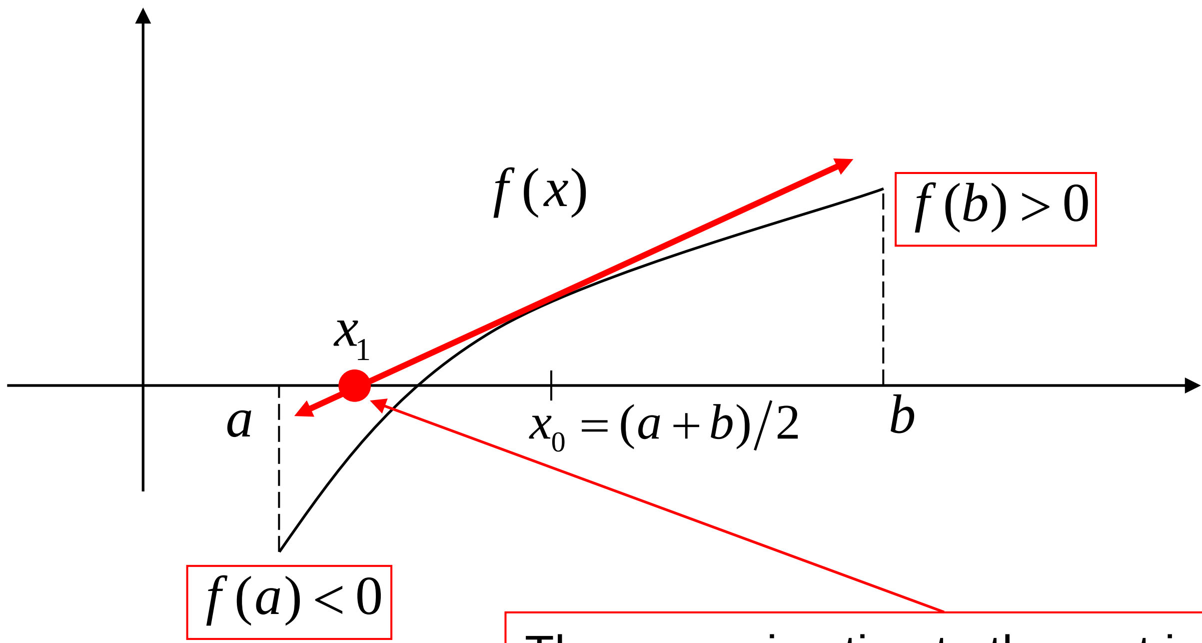
THE NEWTON-RAPHSON METHOD

- THE NEWTON-RAPHSON METHOD Perhaps the most widely used of all root-locating formulas.

- If the **initial guess at the root is x_i**, a tangent can be extended from the point [xi, f(xi)].
  - **The point where this tangent crosses the x axis usually represents an improved estimate of the root.**



Newton-Raphson Method



The approximation to the root is the intersection of the tangent line to the curve drawn at the midpoint of the interval

Now we have to calculate the value of x_1

The equation of the tangent line to the curve of the function $f(x)$ at the point x_0 is

$$f'(x_0) = \frac{y - f(x_0)}{x - x_0}$$

But when the tangent line intersects the x-axis, we have $y = 0$, $x = x_1$

Then $$\frac{-f(x_0)}{x_1 - x_0} = f'(x_0) \longrightarrow \frac{-f(x_0)}{f'(x_0)} = x_1 - x_0$$

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

Similarly, we can calculate

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

Generally, we have

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- Use the Newton-Raphson method to estimate the root of $f(x) = e^{-x} - x$, employing an initial guess of $x_0 = 0.$

- $\boxed{f'(x) = -e^{-x} - 1}$
- $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{e^{-x} - x}{-e^{-x} - 1}$,
- Starting with an initial guess of x_n=0

$$\frac{d}{dx} e^{x} = e^{x} \times 1$$

$$\frac{d}{dx} e^{-x} = e^{-x} \times -1 = -e^{-x}$$

$$x = 0 - \frac{e^{-0} - 0}{-e^{-0} - 1} = 0 - \frac{1}{-2} = 0.5$$

i	x_i	ε_a	ε_t (%)
0	0		100
1	0.500000000	$\frac{0.5 - 0}{0.5}$	11.8
2	0.566311003		0.147
3	0.567143165		0.0000220
4	0.567143290		$< 10^{-8}$

$$X_1 = 0 - \frac{e^{-0} - 0}{-e^{-0} - 1} = 0 - \frac{1}{-1-1} = 0 - \frac{1}{-2} = 0 + \frac{1}{2} = 0.5$$

Convergence

- Thus the error should be roughly proportional to the square of the previous error ➔ **Quadradic convergence**.
- In other words, the number of significant figures of accuracy approximately doubles with each iteration. This behavior is examined in the following example.

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

Pitfalls of the Newton-Raphson Method

- Although the Newton-Raphson method is often very efficient, there are situations where it performs poorly.
- A special case—multiple roots—will be addressed later in this chapter. However, even when dealing with simple roots, difficulties can also arise, as in the following example:

Example of a Slowly Converging Function with Newton-Raphson

Problem Statement.

- Determine the positive root of $f(x) = x^{10} - 1$ using the NewtonRaphson method and an initial guess of $x = 0.5$.
- The Newton-Raphson formula for this case is

$$x_{i+1} = x_i - \frac{x_i^{10} - 1}{10x_i^9}$$

Iteration	x
0	0.5
1	51.65
2	46.485
3	41.8365
4	37.65285
5	33.887565
.	
.	
.	
∞	1.0000000

Thus, after the first poor prediction, the technique is converging on the true root of 1, but at a very slow rate.

Advantages & Drawbacks of Newton Raphson Method

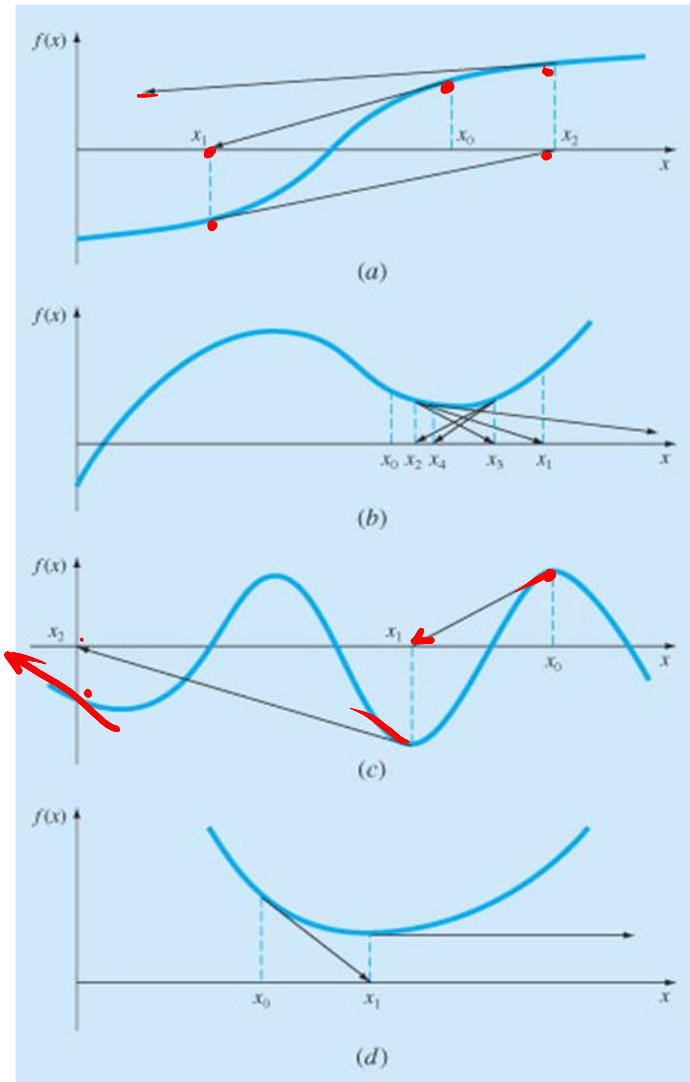
Advantages

- Requires only one guess point
- Converges fast (if it converges !!)

Drawbacks

- The evaluation of the function derivative may be difficult.
- Its convergence depends on:
  - ❑the nature of the function
  - ❑the accuracy of the initial guess

As in the following fig:



The lack of a general convergence criterion also suggests that good computer software should be designed to recognize slow convergence or divergence.

Newton Raphson Algorithm

```
FUNCTION Fixpt(x0, es, imax, iter, ea)
  xr = x0
  iter = 0
  DO
    xrold = xr
    Xr=xold-f(x)/f'(x)
    iter = iter + 1
    IF xr ≠ 0 THEN
```

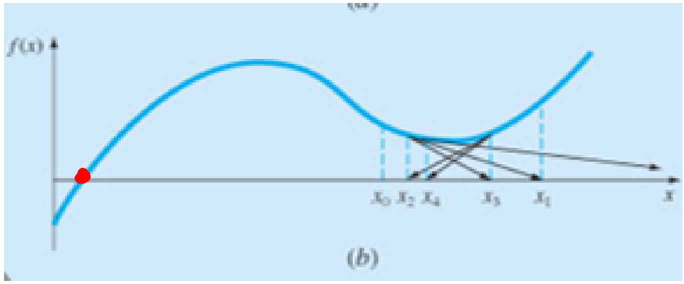
$$ea = \left| \frac{xr - xrold}{xr} \right| \cdot 100$$

```
    END IF
    IF ea < es OR iter ≥ imax EXIT
  END DO
  Fixpt = xr
END Fixpt
```

Additionally, the NewtonRaphson program would be improved by incorporating several additional features:

1-A plotting routine should be included in the program.

2. At the end of the computation, the final root estimate should always be substituted into the original function to compute whether the result is close to zero. This check partially guards against those cases where slow or oscillating convergence may lead to a small value of εa while the solution is still far from a root.



3. The program should always include an upper limit on the number of iterations to guard against oscillating, slowly convergent, or divergent solutions that could persist interminably.

4. The program should alert the user and take account of the possibility that $f(x)$ might equal zero at any time during the computation

Example 1

By using Newton-Raphson method solve the following equation to find a root between $x = 2$, $x = 3$ with an error less than 0.05%

$$x^3 - x - 11 = 0$$

Solution

Let $f(x) = x^3 - x - 11$

Hence $f'(x) = 3x^2 - 1$

Then we have $f(2) = -5$, $f(3) = 13$

Then we have at least one root in the interval $[2,3]$ and we will start by $x_0 = 2.5$

Now we have $$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^{\ 3} - x_n - 11}{3x^2_{\ n} - 1}$$

$$= \frac{3x^3_{\ n} - x_n - x_n^{\ 3} + x_n + 11}{3x^2_{\ n} - 1} \boxed{= \frac{2x^3_{\ n} + 11}{3x^2_{\ n} - 1}}$$

x_n	Error $= \left\lvert \frac{x_n - x_{n-1}}{x_n} \right\rvert \times 100$
$x_0 = 2.5$	$-----$
$x_1 = 2.380282$	5.0296%
$x_2 = 2.373669$	0.2785%
$x_3 = 2.37365$	0.0008%

The convergence to the solution is very fast

Example 2

In a grinding process through a specific industrial application, the equation that illustrates the amount of metallic residues R in milligrams that are expected to be produced through the process as a function of time t seconds is given by

$$R(t) = e^{-t} - t$$

By using Newton-Raphson method find an approximation for the time needed to grind all the metallic residues. Use an initial guess for time $t_0 = 0$ and find the approximation with an error less than 0.00005%

$e^{-x} - x$

Solution

We have $\boxed{R(t) = e^{-t} - t}$, $R'(t) = -e^{-t} - 1$

Then

$$t_{n+1} = t_n - \frac{R(t_n)}{R'(t_n)} \quad \boxed{= t_n - \frac{e^{-t_n} - t_n}{-e^{-t_n} - 1}}$$

t_n	Error $= \left\lvert \frac{t_n - t_{n-1}}{t_n} \right\rvert \times 100$
$t_0 = 0$	$-----$
$t_1 = 0.5$	100%
$t_2 = 0.566311$	11.7%
$t_3 = 0.56714317$	0.147%
$t_4 = 0.56714329$	0.000022%