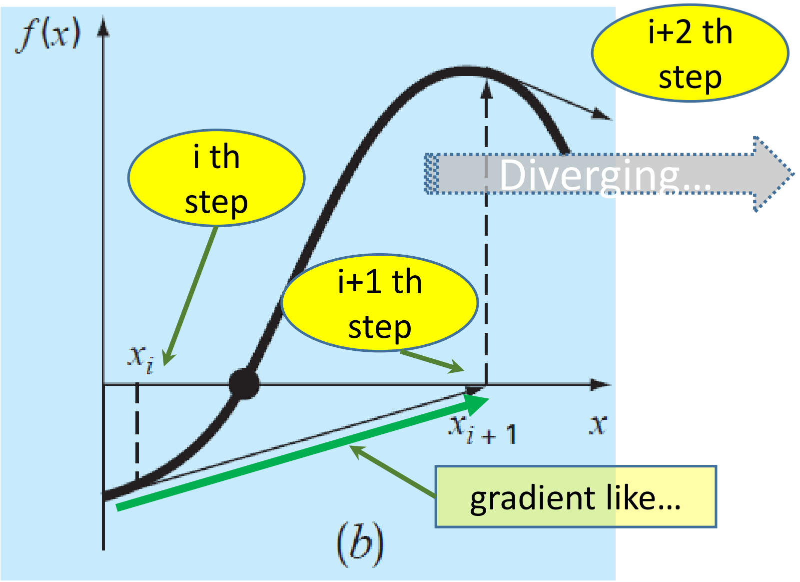
Open Methods

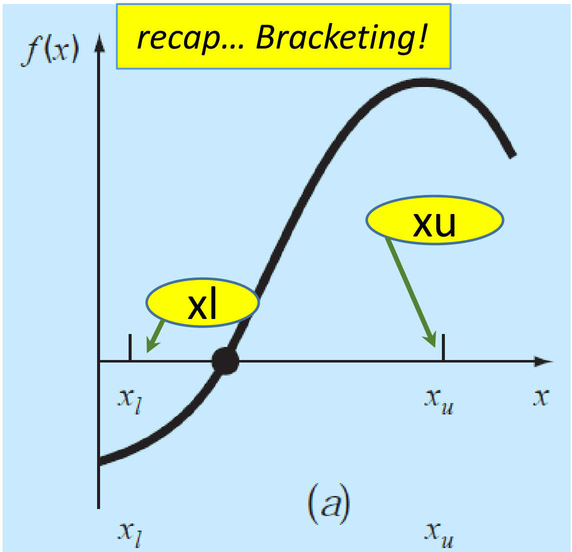
Open methods…

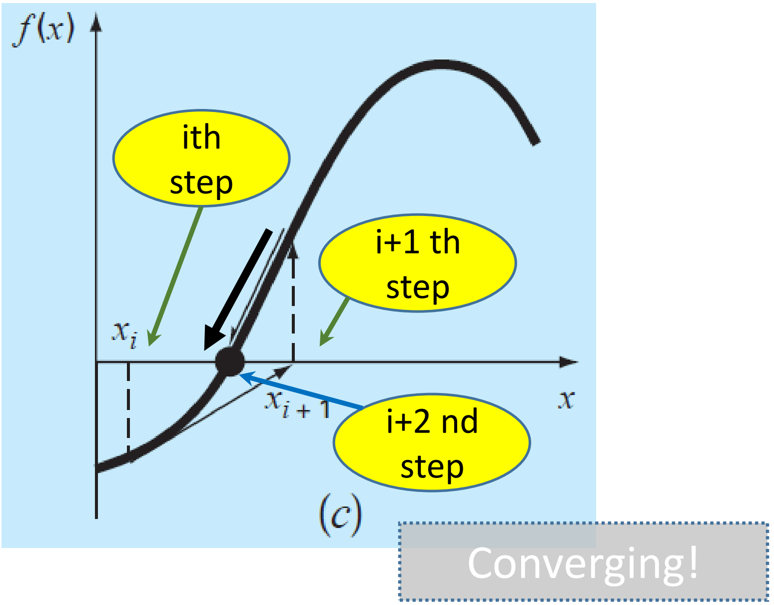


Algorithm:

```
Initial guess of the root estimate

Repeat until condition

    1. Use Local gradient of the
    curve

    2. Determine next root
    estimate
```

- May converge or may diverge
- Much ***much*** faster than bracketing – relation to Taylor series & minimizing residual error





Open Methods

- Simple Fixed point method
- The Newton-Raphson method
- The secant method
- Error convergence rates – linear and quadratic
- Comparison with bracketing algorithms

Simple Fixed Point method

- Example 1
  - $x^2 - 2x + 3 = 0$
  - $\Rightarrow x = \frac{x^2+3}{2}$
- Example 2
  - $\sin(x) = 0 \Rightarrow x = \sin(x) + x$
- In general
  - Rearrange terms such that $f(x) = 0$
  - $\Rightarrow x = g(x)$
- <u>Or Simply</u> $x = f(x) + x$

<u>Strategy…</u>

- $x_1 = g(x_0)$
- $x_2 = g(x_1)$

…

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

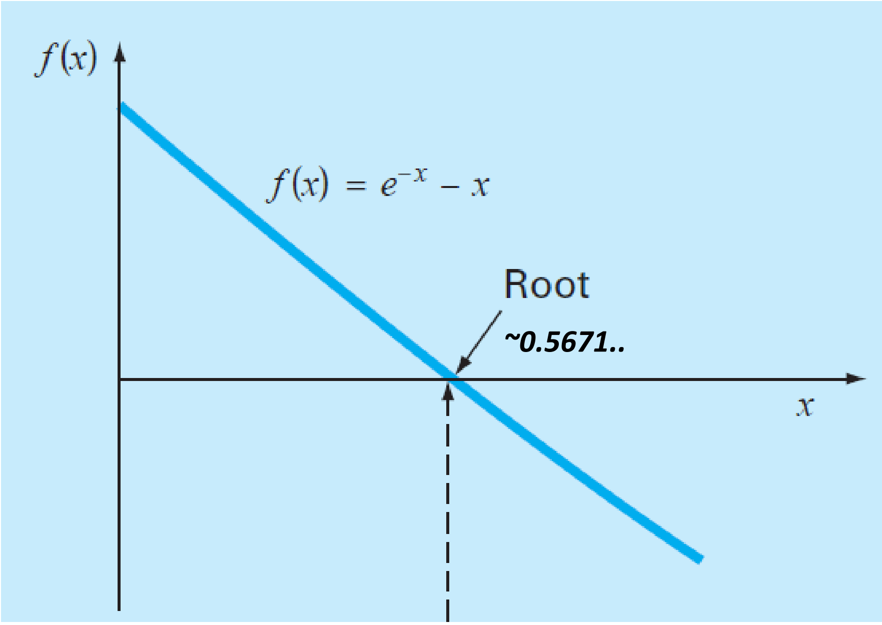
- Error estimates –

$$\varepsilon_a = \left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| * 100\%$$

Simple fixed point method (2/6)

- Find roots of: $f(x) = e^{-x} - x$
- Re-arranging
  - $f(x) = 0$
  - $\Rightarrow x = e^{-x}$
  - $\Rightarrow x_{i+1} = e^{-x_i}$
- Analytically solving it, true root = 0.56714329

Curve

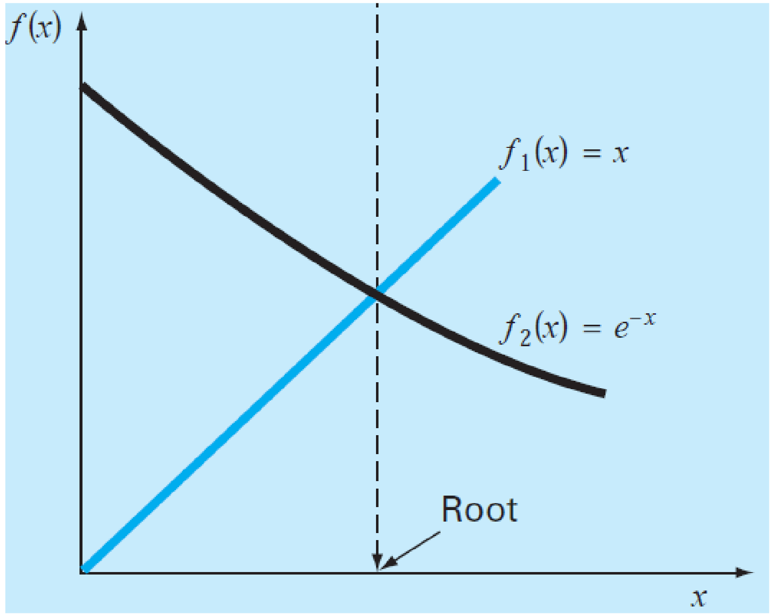


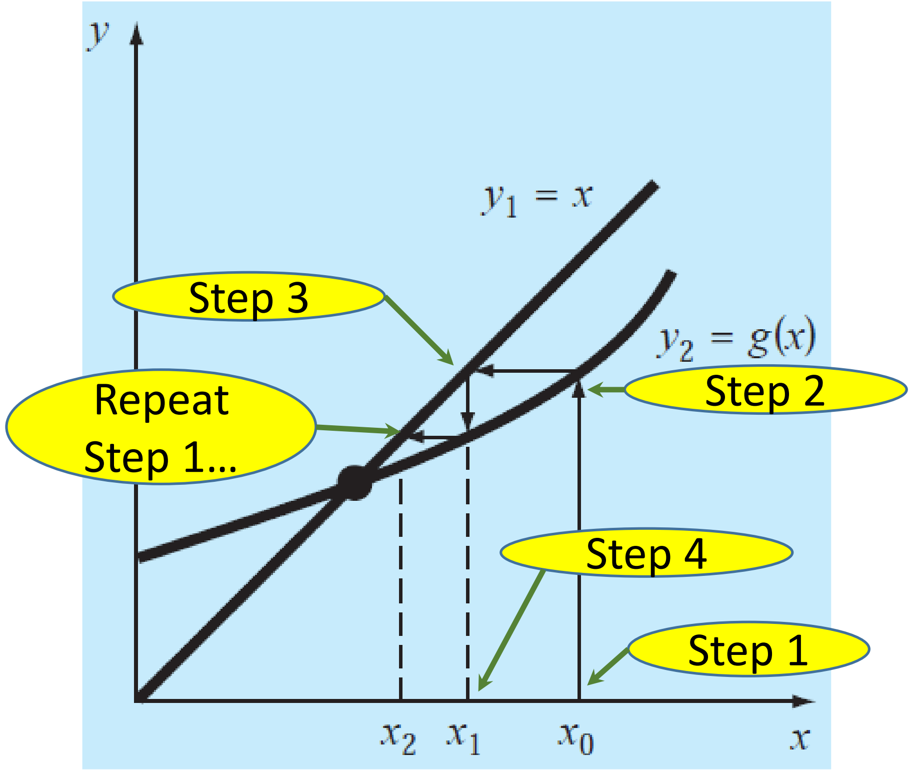
Iterations

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	0.564879	1.11	0.399

Simple fixed point method (3/6)

- $x_{i+1} = e^{-x_i}$
  - $f_1(x) = x$
  - $f_2(x) = \mathrm{e}^{-x}$
- Intersection point is the root
- Understanding the iterations
  - $x_{i+1} = g(x_i)$
  - Start with x_0 (step 1)
  - Determine $g(x_0)$ (step 2)
  - Project onto y_1 line (step 3)
  - Project onto X axis → x_1 (step 4)
  - Repeat from step 1



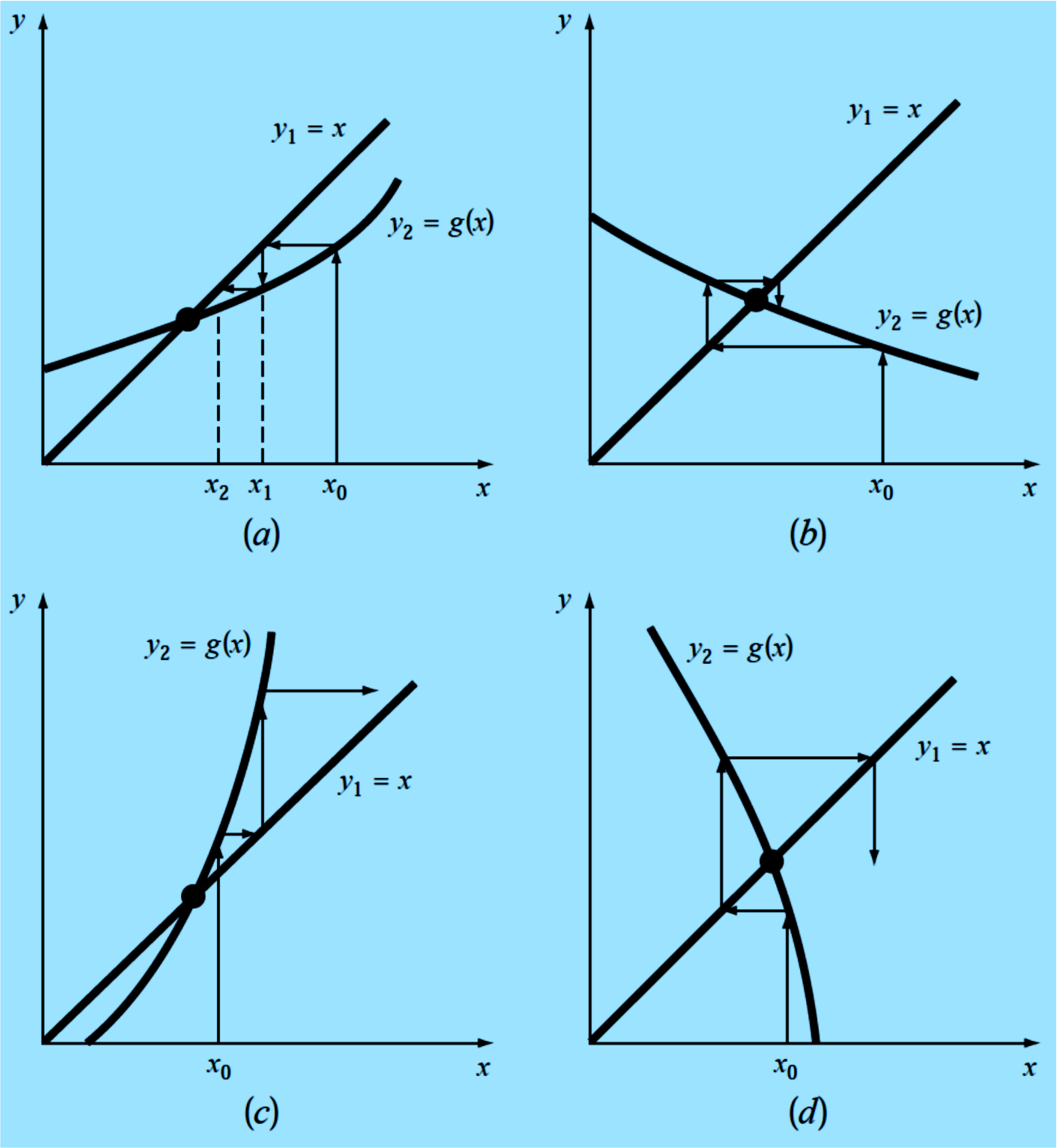


Simple fixed point method (4/6)

- Simple fixed point method can
  - Converge (figures ***a*** and ***b***)

  or
  - Diverge (figures ***c*** and ***d***)

- Depends on the initial guess
- Depends on the ***<u>nature</u>*** of the curve
  - Informally, observe that
    - Inclination of the curve **<** the diagonal line → Converges
    - Inclination of the curve **>** the diagonal line → Diverges



Simple fixed point method (5/6)

1. $x_{i+1} = g(x_i)$
2. At the root $x_r = g(x_r)$
3. [1] – [2] →

$$x_r - x_{i+1} = g(x_r) - g(x_i)$$

4. *Mean value theorem*

$$g'(\varepsilon) = \frac{g(b) - g(a)}{b - a}$$

To explain on board

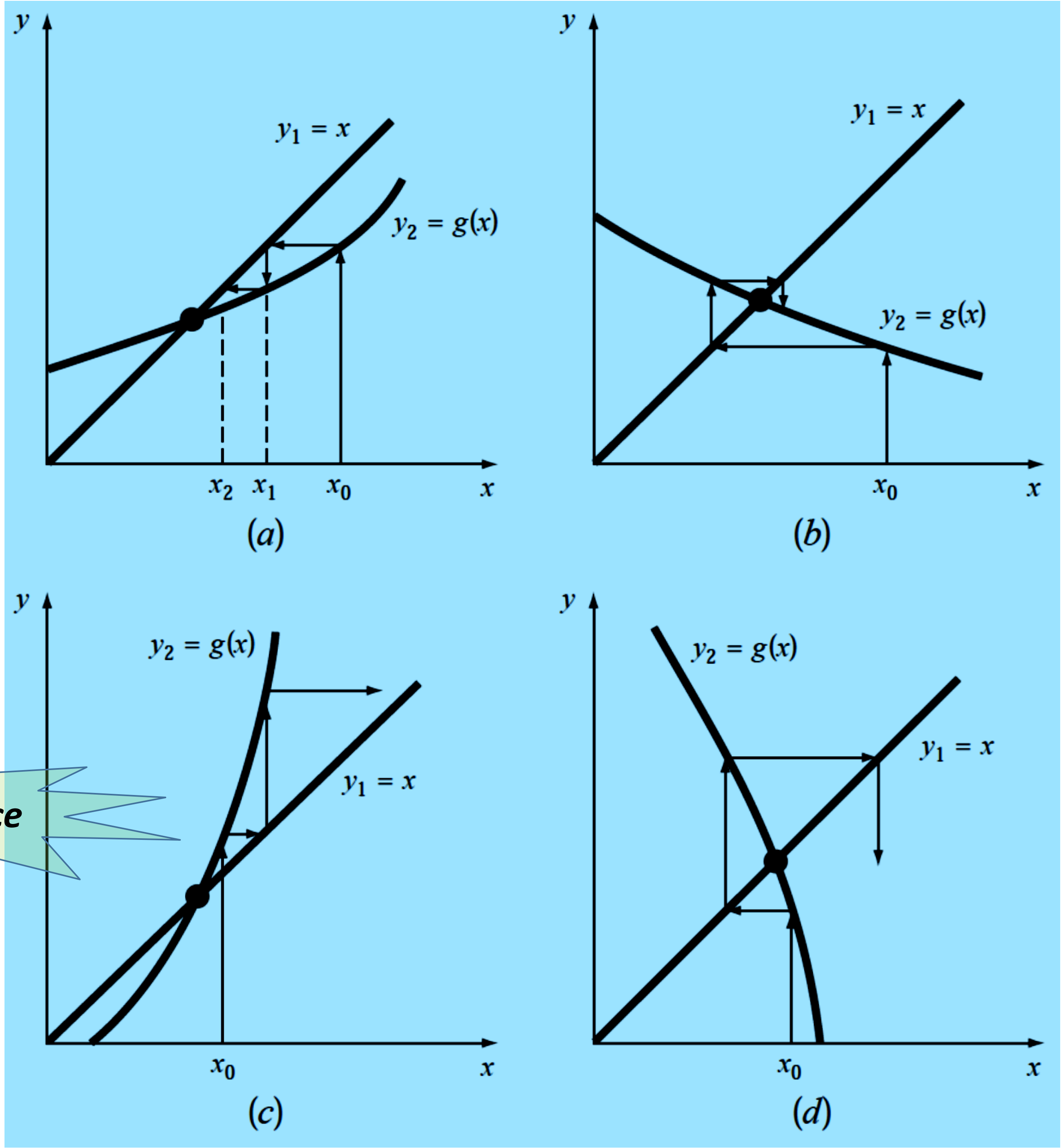
5. $g(x_r) - g(x_i) = (x_r - x_i)g'(\varepsilon)$
6. [5] – [3] → $x_r - x_{i+1} = (x_r - x_i)g'(\varepsilon)$
7. Let *true* error at i^th iteration be

$$\mathrm{E}_{t,i} = x_r - x_i$$

4. [6] → $\mathrm{E}_{t,i+1} = g'(\varepsilon) * E_{t,i}$

Linear Convergence

5. Observations
   - Diverges if $|g'(\varepsilon)| > 1$
   - Converges if $|g'(\varepsilon)| < 1$ and further
     - Flip flops/*Oscillates* about the root if $g'(\varepsilon) < 0$
     - Monotonically converges if $g'(\varepsilon) > 0$



Simple fixed point method (6/6)

- Note that only single starting point is provided
- g() is a function which is invoked in the pseudo code
- Exercise:
  - Trace the pseudo code for 4 iterations
  - x0 = 0
  - $g(x) = x^2 - 2x + 3 = 0$

```
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 = |(xr − xrold) / xr| · 100
    END IF
    IF ea < es OR iter ≥ imax EXIT
  END DO
  Fixpt = xr
END Fixpt
```

The Newton-Raphson method

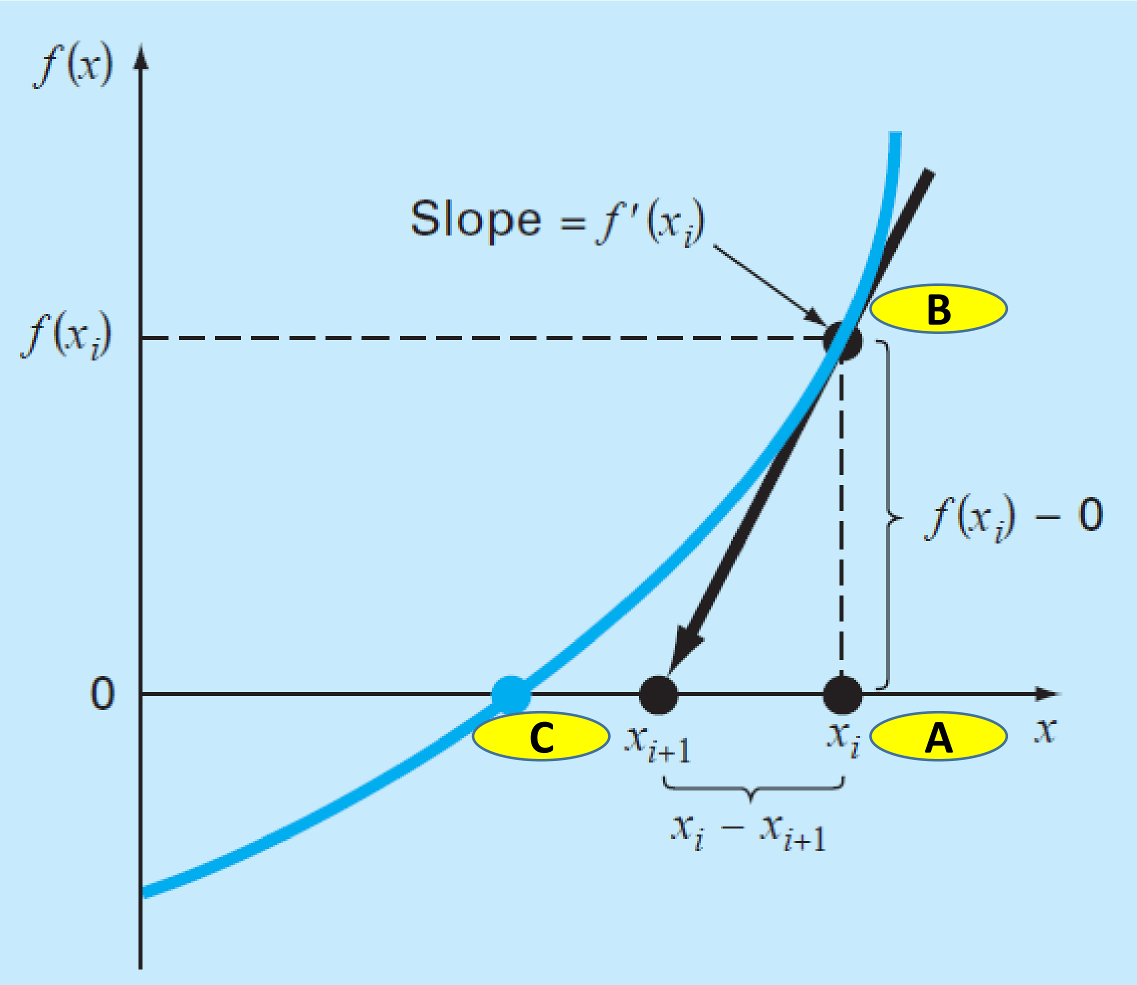
- The mechanism of selection of next root estimate changes
- Consider triangle ABC
- First derivative at B = Slope

→

- $f'(x_i) = \frac{f(x)-0}{x_i - x_{i+1}}$

→

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



Newton-Raphson… 2/7

The Newton-Raphson formula can be realized using Taylor series as well

- First order approximation

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

- We need to determine x_{i+1} where $f(x_{i+1})$ cuts X axis i.e. $f(x_{i+1}) = 0$

→

$$0 = f(x_i) + f'(x_i)(x_{i+1} - x_i)$$

→

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

Newton-Raphson… 3/7

- Consider the same function as earlier
- $f(x) = e^{-x} - x$

→

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

We know

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

→

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

Starting with $x_0 = 0$

Iterations

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	$< 10^{-8}$

Informally, observe that

- Converges rapidly
- Accuracy *doubles* after every iteration i.e. number of digits after the decimal

Newton-Raphson… 4/7

Understanding the rapid convergence

- Taylor series up to second order terms

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(\xi)}{2!}(x_{i+1} - x_i)^2$$

where $\xi \in [x_i, x_{i+1}]$

- Recall that

$0 = f(x_i) + f'(x_i)(x_{i+1} - x_i)$ [Equation 1]

- Let x_r be the root → $x_{i+1} = x_r$

→

- $0 = f(x_i) + f'(x_i)(x_r - x_i) + \frac{f''(\xi)}{2!}(x_r - x_i)^2$

[Equation 2]

[2] – [1]

→

$0 = f'(x_i)(x_r - x_{i+1}) + \frac{f''(\xi)}{2!}(x_r - x_i)^2$ [Equation 3]

Let *true* error at i^th iteration be defined as $\mathrm{E}_{t,i} = x_r - x_i$

Equation [3] →

$$0 = f'(x_i) * E_{t,i+1} + \frac{f''(\xi)}{2!} * E^2(t,i)$$

At the root, $\xi \approx x_r$

Quadratic Convergence

$$\Rightarrow E_{t,i+1} = \frac{-f''(x_r)}{2f'(x_r)} E^2(t,i)$$

Newton-Raphson… 5/7

Consider the function: $f(x) = x^{10} - 1$

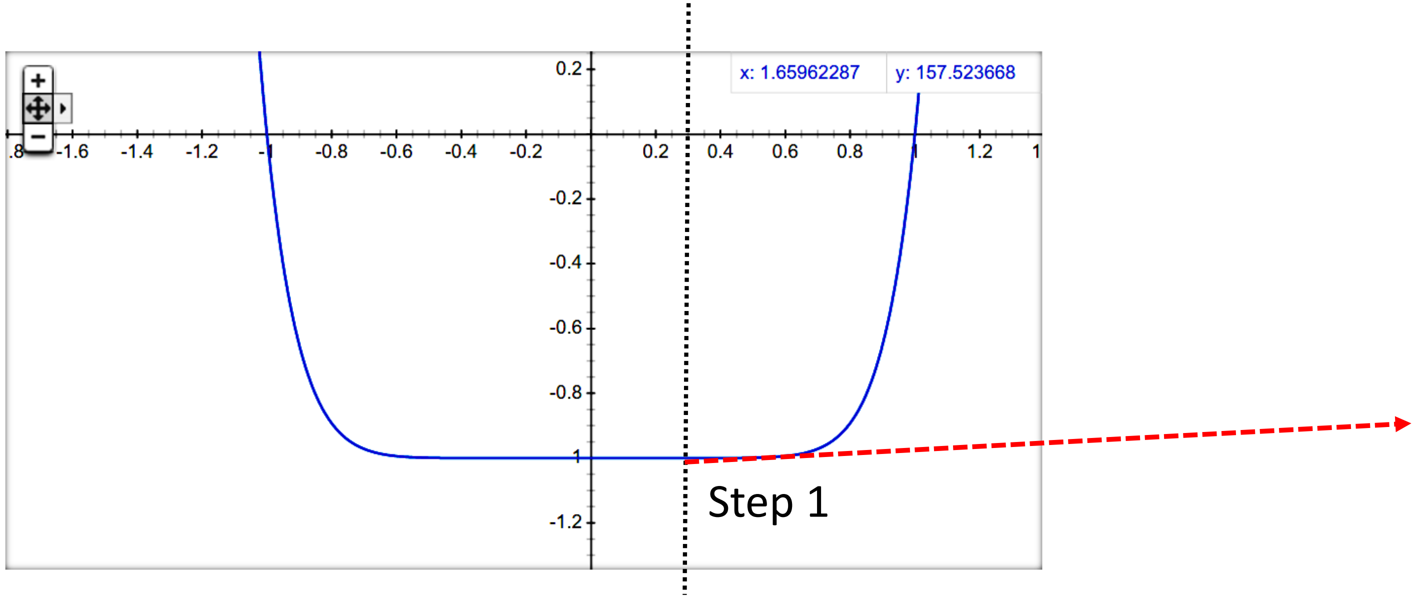
- Slow convergence
- Other problems

Criterion

- $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$
- $|x_{i+1} - x_i| < \delta$
- $\rightarrow |f(x_i)| < \delta |f'(x_i)|$
- $\rightarrow f'(x_i)$ is Large

Iterations

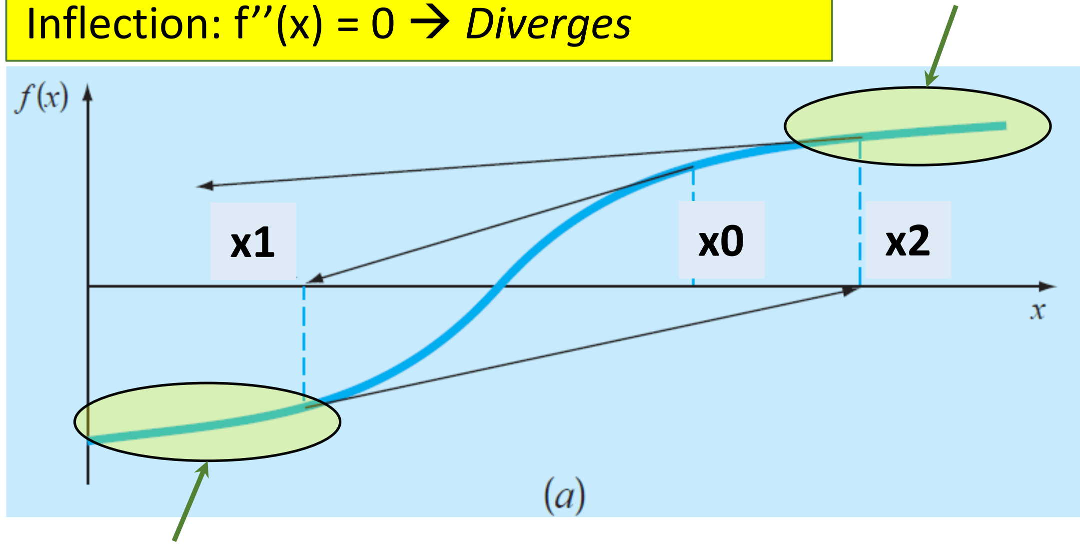
Iteration	x
0	0.5
1	51.65
2	46.485
3	41.8365
4	37.65285
5	33.887565
.	
.	
.	
∞	1.0000000



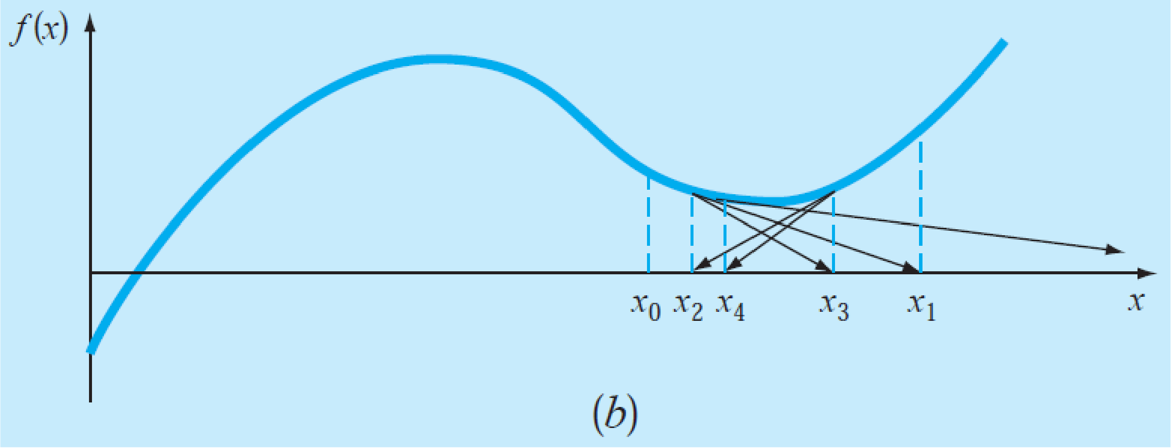
51.65

Newton-Raphson… 6/7

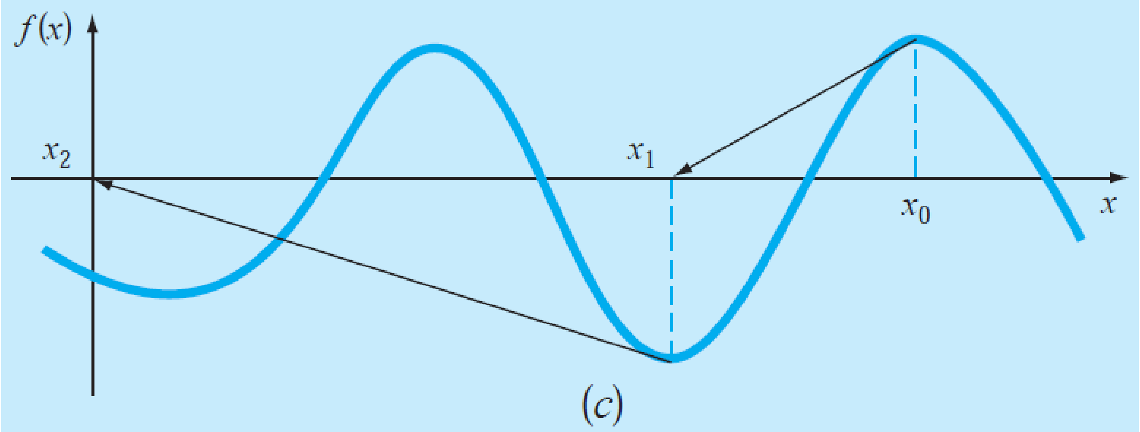
Inflection: f’’(x) = 0 → *Diverges*



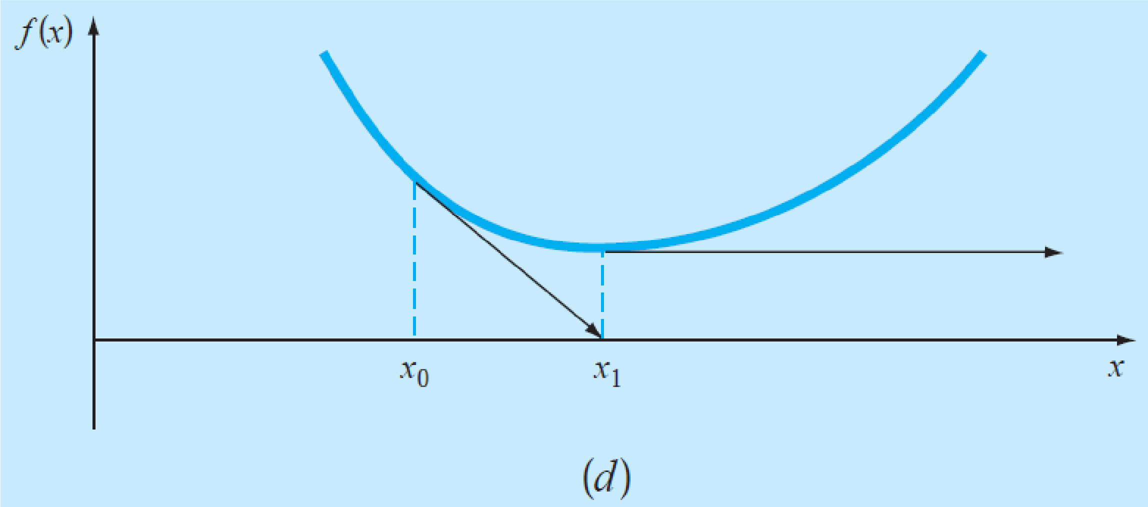
Maxima or Minima → *Oscillates*



Multiple roots → *Jumps away from close by root*



Divide by zero → *New point never touches X axis*



Newton-Raphson… 7/7

- Write pseudo code for the Newton-Raphson method

- Issue – We need f’() function

- Bounds
    - Number of iterations
    - Oscillation checks $\varepsilon_a \approx 0$
    - Slow convergence
    - Divergence checks
    - f’(x) = 0 check

NR

FUNCTION $\boxed{\text{NR}}$*(x0, es, imax, iter, ea)*

  xr = x0

  iter = 0

  DO

    xrold = xr

    ~~*xr = g(xrold)*~~ ⇒ $\boxed{x_r = x_r - \frac{f(x_r)}{f'(x_r)}}$

    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

  $\boxed{\text{NR}}$ *= xr*

END Fixpt

The Secant method

- Newton-Raphson method

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

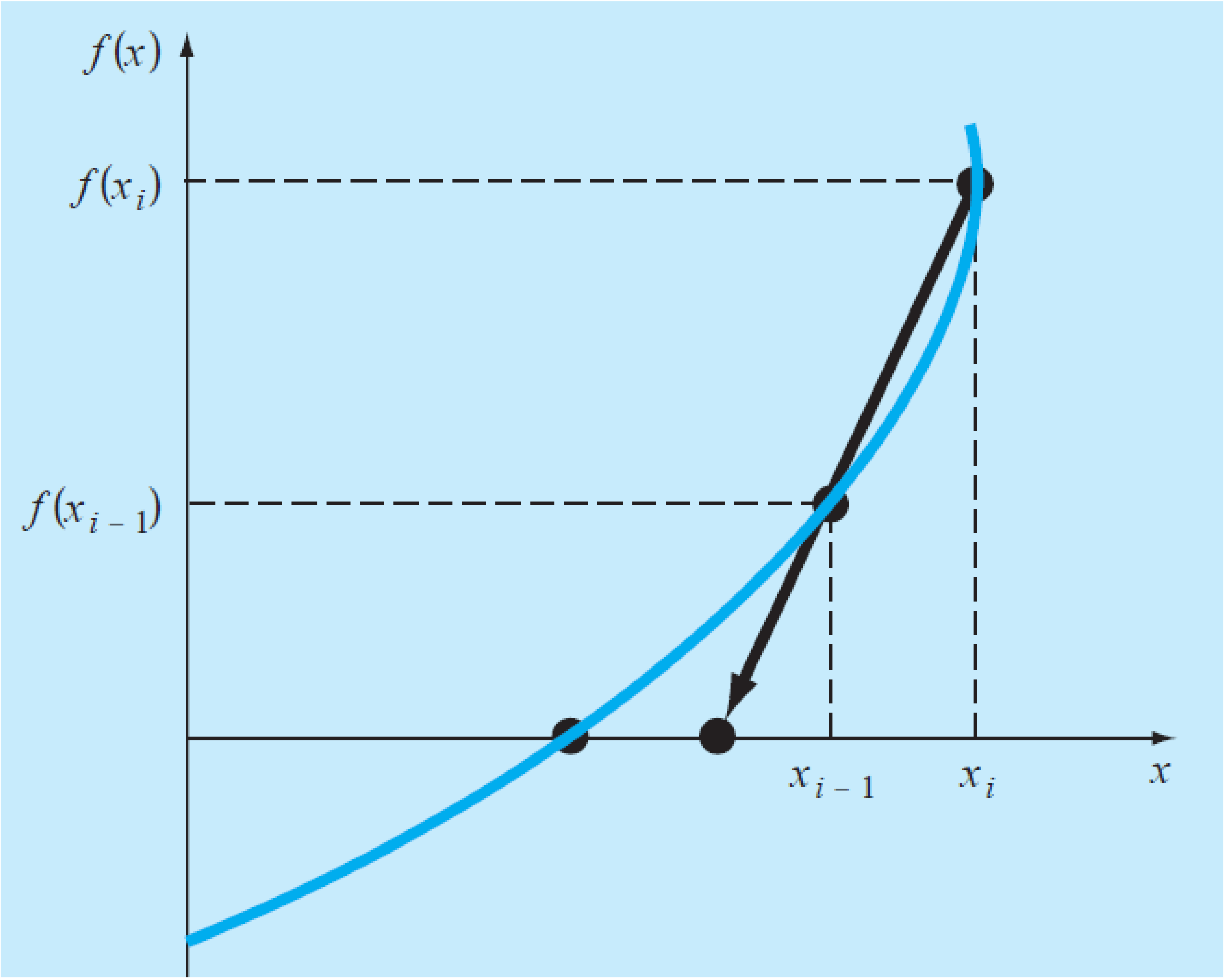
- It needs derivative – it is difficult or inconvenient to evaluate

- Using *backward finite difference*

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

→

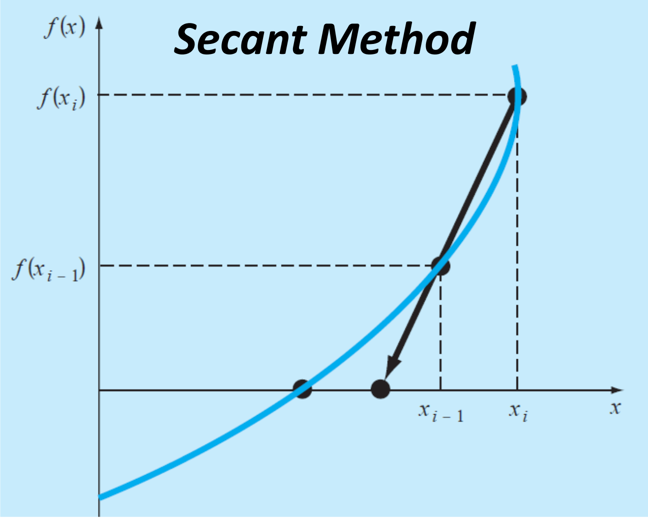
$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$$



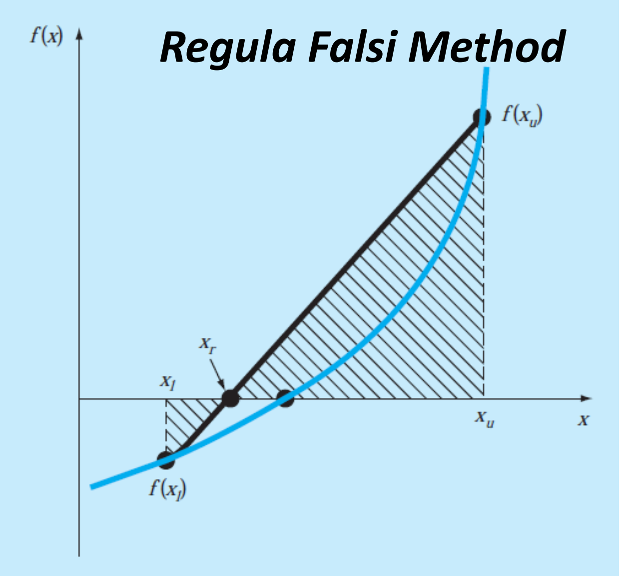
Secant Method

<u>Secant method</u> Relationship with/against <u>Regula Falsi method</u>

- Secant method update equation
- $x_{i+1} = x_i - \frac{f(x_i)(x_{i-1}-x_i)}{f(x_{i-1})-f(x_i)}$

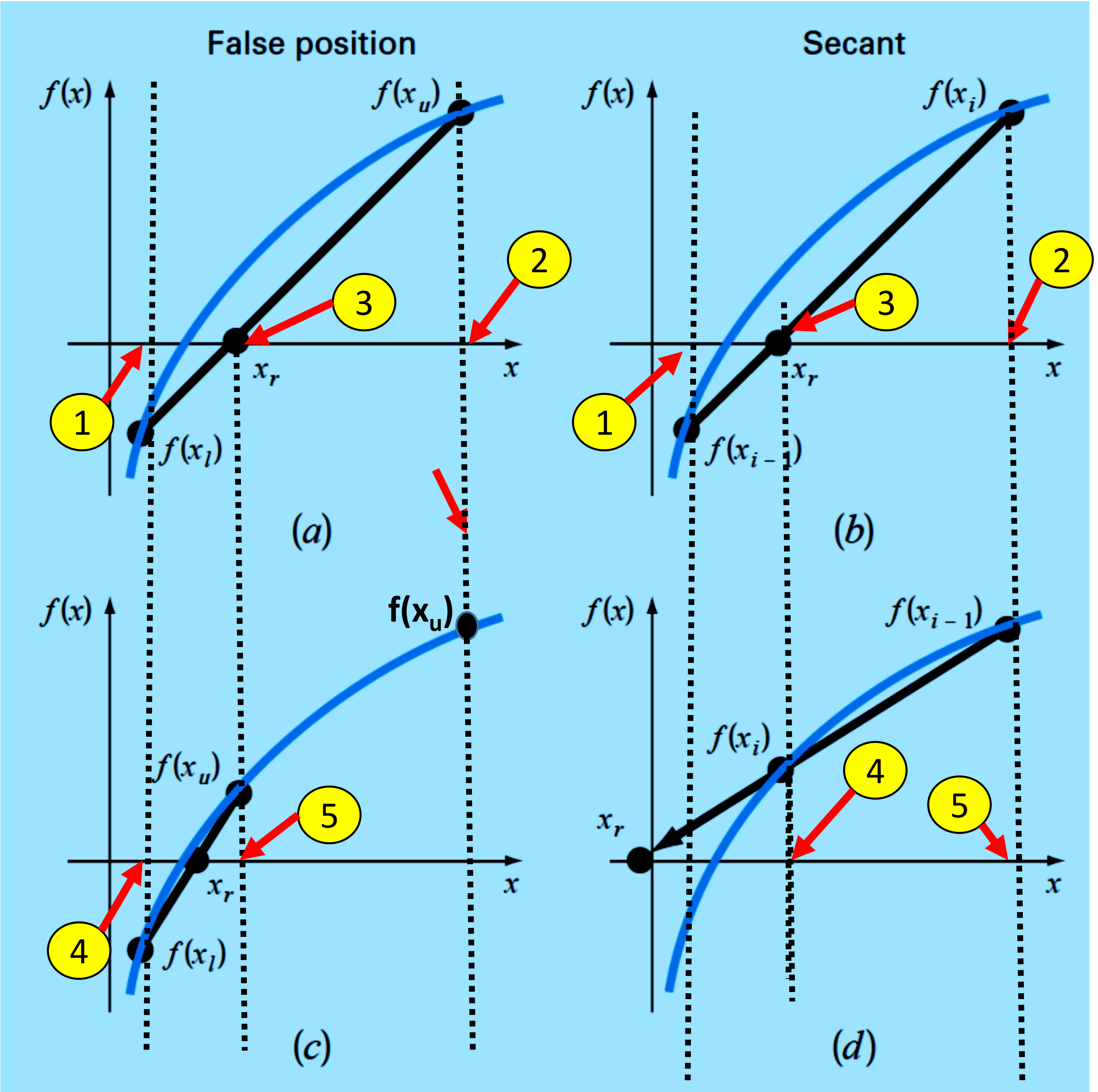


- Regula Falsi method update equation
- $x_r = x_u - \frac{f(x_u)(x_l-x_u)}{f(x_l)-f(x_u)}$



Secant Method…

- Updates to coordinates are different between secant method and the regula falsi
- In a given iteration for (a) and (b) Third coordinate determined is same
- Continuing the iterations further, coordinate assignments change



Secant Method…

- For the function
  - $f(x) = \ln(x)$
  - $x_l = x_{i-1} = 0.5$
  - $x_u = x_i = 5.0$
- Issues – we need x_{i-1} and x_i
- Introduce a δ term, but have a single variable as input
- $f(x) \cong \frac{f(x_i+\delta x_i)-f(x_i)}{\delta x_i}$
- Issue – choice of size of δ
  - Too small, will give round off errors
  - Too big, the method can diverge

Regula Falsi Method

Iteration	x_l	x_u	x_r
1	0.5	5.0	1.8546
2	0.5	1.8546	1.2163
3	0.5	1.2163	1.0585

Secant Method

Iteration	x_{i-1}	x_i	x_{i+1}
1	0.5	5.0	1.8546
2	5.0	1.8546	−0.10438

Convergence plot of different methods