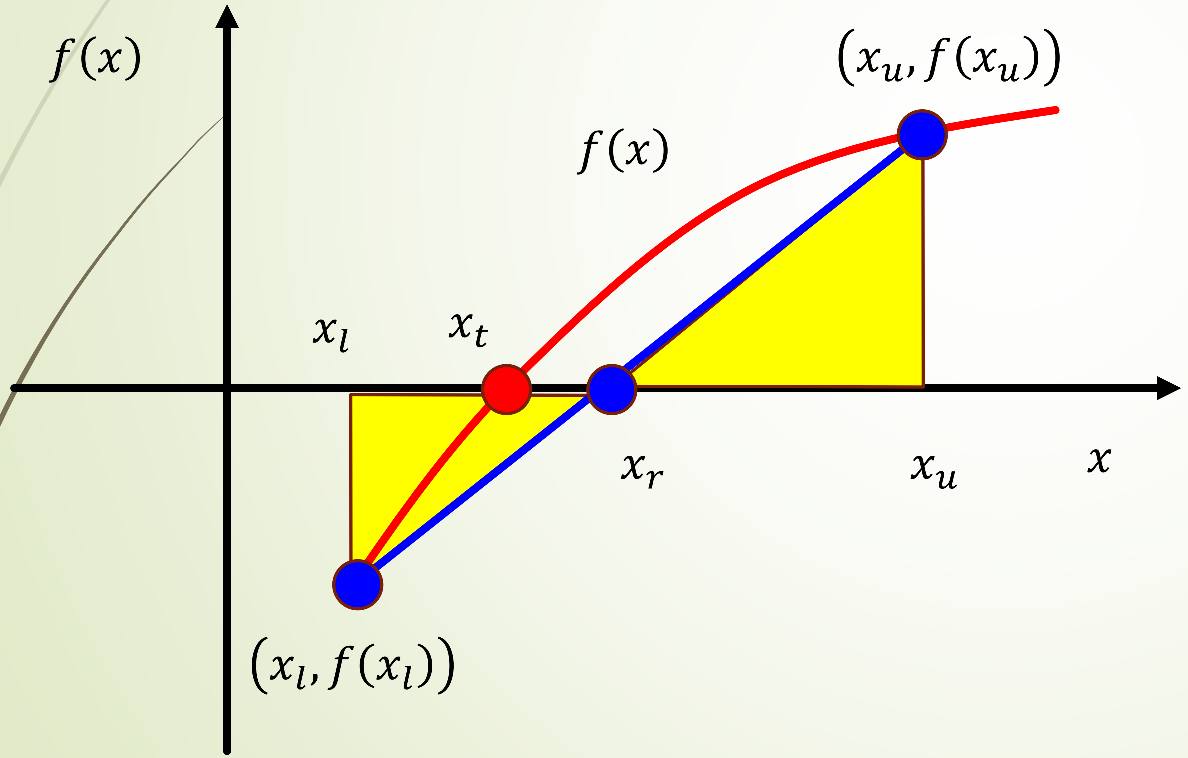
False Position (가위치법)

Prof. Sang-Chul Kim, 2017-3-30

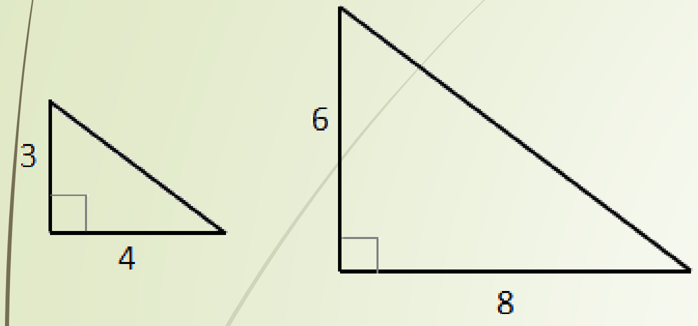
False Position

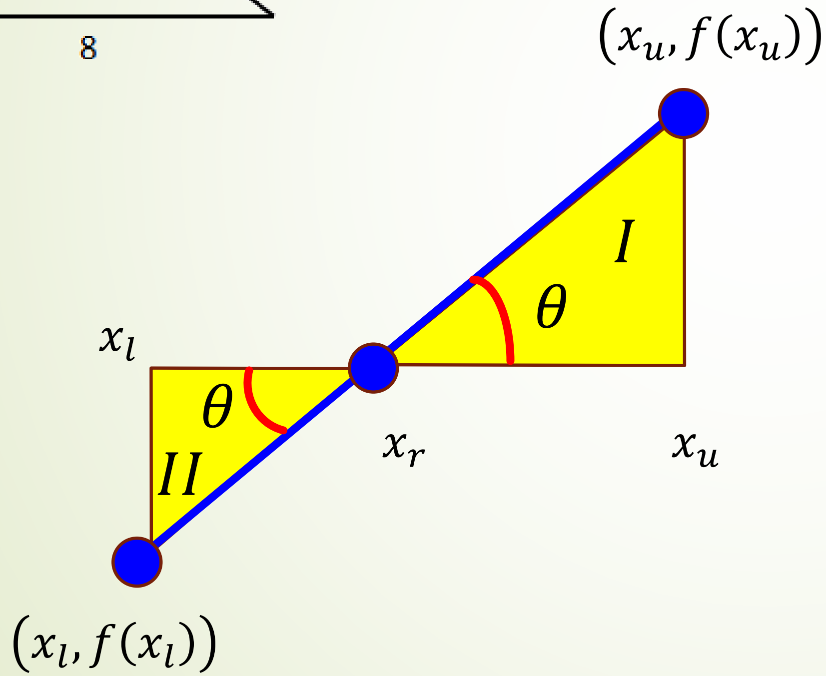


$$f(m) = \sqrt{\frac{9.8 \cdot m}{0.25}} \cdot tanh\left(\sqrt{\frac{9.8 \cdot 0.25}{m}} \cdot 4\right) - 36$$

$$f(x) = \sqrt{x \cdot c_1} \cdot tanh\left(\sqrt{\frac{1}{x}} \cdot c_2\right) - c_3$$

Similarity in Triangles (닮은꼴 삼각형)

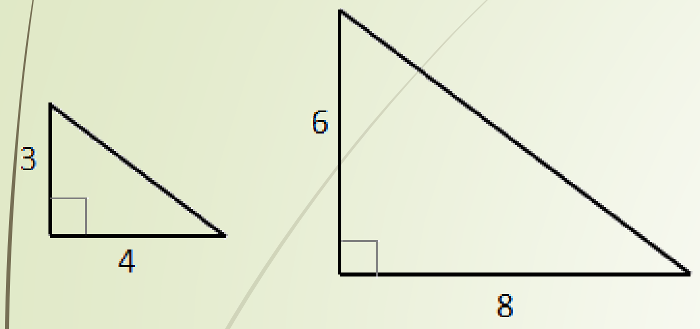


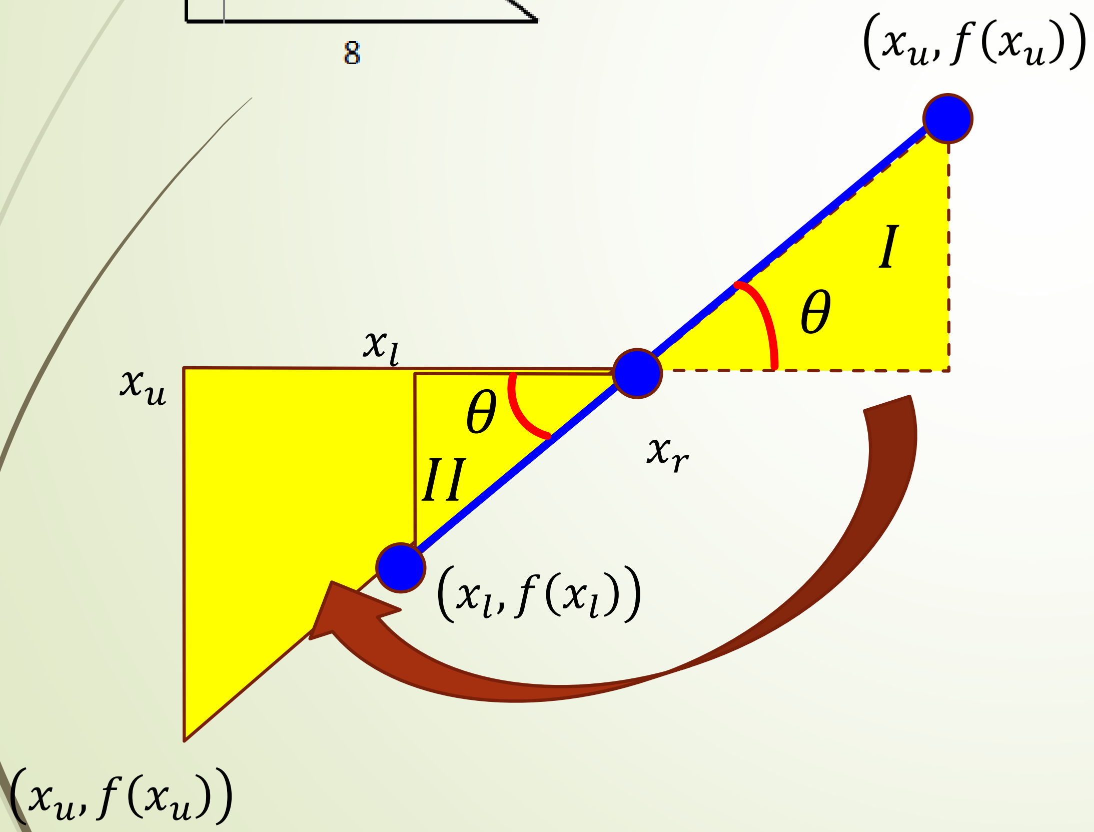


- Triangles I and II are Similar
- Triangle I's $\frac{Height}{Base}$ = Triangle II's $\frac{Height}{Base}$

$$\frac{f(x_u) - 0}{x_u - x_r} = \frac{f(x_l) - 0}{x_r - x_l}$$

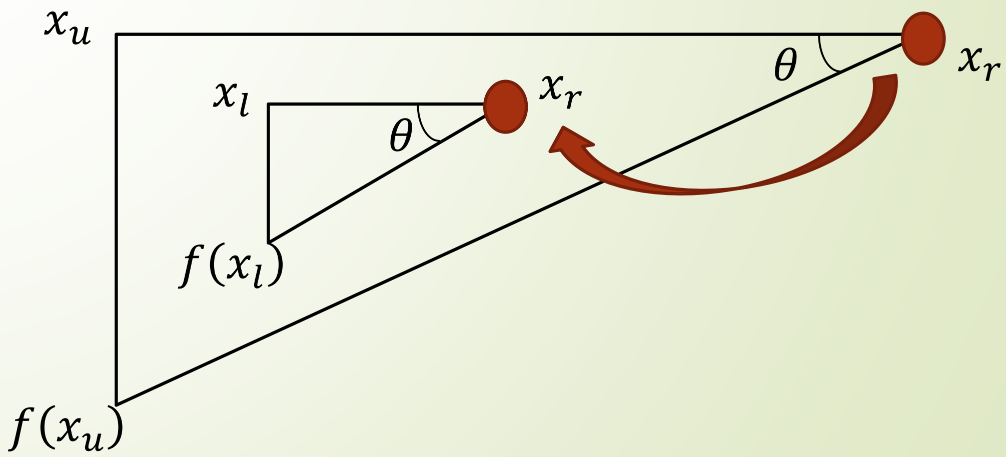
Similarity in Triangles (닮은꼴 삼각형)





- Triangles I and II are Similar
- Triangle I's $\frac{Height}{Base}$ = Triangle II's $\frac{Height}{Base}$

$$\frac{f(x_u) - 0}{x_r - x_u} = \frac{f(x_l) - 0}{x_r - x_l}$$



Calculate x_r

$$\frac{f(x_u) - 0}{x_r - x_u} = \frac{f(x_l) - 0}{x_r - x_l}$$

$$f(x_l) \cdot (x_r - x_u) = f(x_u) \cdot (x_r - x_l)$$

$$f(x_l) \cdot x_r - f(x_l) \cdot x_u = f(x_u) \cdot x_r - f(x_u) \cdot x_l$$

Calculate x_r

$$f(x_l)\cdot x_r - f(x_l)\cdot x_u = f(x_u)\cdot x_r - f(x_u)\cdot x_l$$

$$f(x_l)\cdot x_r - f(x_u)\cdot x_r = f(x_l)\cdot x_u - f(x_u)\cdot x_l$$

$$\big(f(x_l) - f(x_u)\big)\cdot x_r = x_u\cdot f(x_l) - x_l\cdot f(x_u)$$

Calculate x_r

$$\left(f(x_l) - f(x_u)\right) \cdot x_r = x_u \cdot f(x_l) - x_l \cdot f(x_u)$$

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

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

Calculate x_r

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

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

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

Calculate x_r

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

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

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

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

False Position Coding

```
import numpy as np

def false_position(func, xl, xu):
    maxit=100
    es=1.0e-4

    test=func(xl)*func(xu)

    if test > 0:
        print('No sign change')
        return [], [], [], []

    iter=0
    xr=xl

    ea=100
```

False Position Coding

```
while (1):
    xrold=xr
    #xr=np.float((xl+xu)/2)
    xr=np.float(xu-func(xu)*(xl-xu)/(func(xl)-func(xu)))

    iter=iter+1

    if xr != 0:
        ea=np.float(np.abs(np.float(xr)-np.float(xrold))/np.float(xr))*100

    test=func(xl)*func(xr)
```

False Position Coding

```
if test > 0:
        xl=xr
    elif test < 0:
        xu=xr
    else:
        ea=0

    if np.int(ea<es) | np.int(iter >= maxit):
        break

root=xr
fx=func(xr)

return root, fx, ea, iter
```

False Position Coding for Results

```
fm=lambda m: np.sqrt(9.81*m/0.25)*np.tanh(np.sqrt(9.81*0.25/m)*4)-36
root, fx, ea, iter=false_position(fm, 40, 200)

print('root = ', root , '(False Position)')
print('f(root) = ', fx, '(must be zero, False Position)')
print('estimated error= ', ea, '(must be zero error, False Position)')
print('iterated number to find root =', iter , '(False Position)')
```

Results of False Position

- root = 142.73783844758216 (False Position)
- f(root) = 4.20034974979e-06 (must be zero, False Position)
- estimated error= 7.781013797744088e-05 (must be zero error, False Position)
- iterated number to find root = 29 (False Position)

Results of Bisection

False position has less error and more iteration than Bisection

- root = 142.73765563964844 (Bisection)
- root = 142.73783844758216 (False Position)

- f(root) = 4.60891335763e-07 (must be zero, Bisection)
- f(root) = 4.20034974979e-06 (must be zero, False Position)

- estimated error= 5.3450468252827136e-05 (must be zero error, Bisection)
- estimated error= 7.781013797744088e-05 (must be zero error, False Position)

- iterated number to find root = 21 (Bisection)
- iterated number to find root = 29 (False Position)