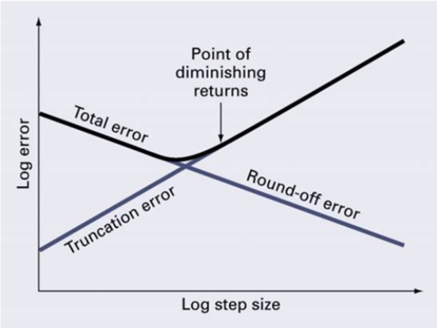
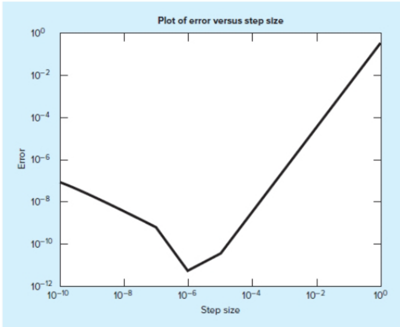
Round-off errors and truncation errors

we used a centered difference approximation to estimate the first derivative of the following function at x = 0.5,

$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$

Perform the same computation starting with h = 1. Then progressively divide the step size by a factor of 10 to demonstrate how roundoff becomes dominant as the step size is reduced.





Round-off errors and truncation errors

Let's take the example of second-order FD of the first derivative,

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{f'''}{6}h^2$$

And since we have round-off errors the values of the function f are saved in computer as $\tilde{f}$ with a truncation of e, and we have,

$$f(x_{i+1}) = \tilde{f}(x_{i+1}) + \tilde{e}_{i+1}$$
$$f(x_{i-1}) = \tilde{f}(x_{i-1}) + \tilde{e}_{i-1}$$

Thus,

$$f'(x_i) = \frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h} + \frac{\tilde{e}_{i+1} - \tilde{e}_{i-1}}{2h} - \frac{f'''}{6}h^2$$

For a maximum absolute value of ε for $(\tilde{e}_{i+1} - \tilde{e}_{i-1})/2$, and a maximum absolute value of M for $-f'''$ we have,

$$Total\ Error = \left| f'(x_i) - \underbrace{\frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h}}_{\text{discretized } f(x)} \right| < \frac{\varepsilon}{h} + \frac{Mh^2}{6}$$

The optimum value of h is found to be, $h = \sqrt[3]{\frac{3\varepsilon}{M}}$

$f(\underbrace{x_{i+1}}_{\text{index}})$

1. Find 3rd derivative

$f'(x) = -0.4x^3 - 0.45x^2 - x - 0.25$

$f''(x) = -1.2x^2 - 0.9x - 1$

$f^3(x) = -2.4x - 0.9$

2. Find M

$M = |f^3(0.5)| \rightarrow |-2.4(0.5) - 0.9| = 2.1$

$\varepsilon = 0.5 \times 10^{-16}$

3. Find h

$h = \sqrt[3]{\frac{3\varepsilon}{M}}$

$\rightarrow \sqrt[3]{\frac{3(0.5 \times 10^{-16})}{2.1}} = \boxed{4.3 \times 10^{-16}}$

4. Write in Matlab

```
dfdx = @(x) -0.4x^3 -0.45x^2 - x - 0.25;
fx = @(x) -0.1x^4 -0.15x^3 -0.25x +1.2;
x_zero = 0.5;
true_val= dfdx(0.5);
delta_h = 1;


for i =1:10

    disc_f = (fx(x_zero + delta_h) - fx(x_zero -delta_h)) / (2 * delta_h);

    error(i) = abs(true_val - disc_f);
        saves error into array
    delta_h = delta_h /10;

    h_save(i)= delta_h;
        saves delta_h into array
end
```

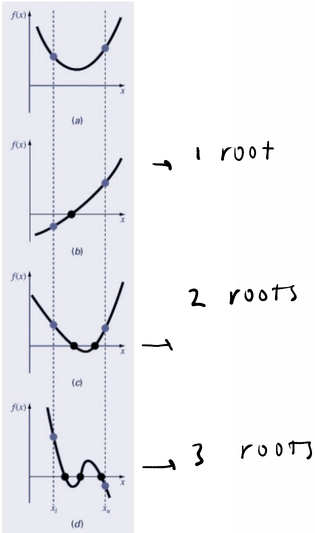
plot (h-save , error)

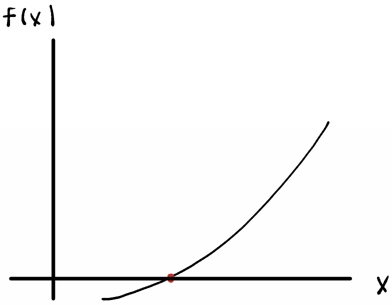
Braketing Methods

Graphical Methods

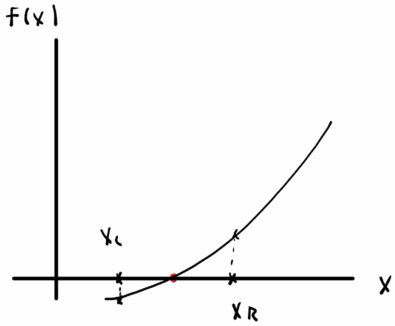
- A simple method for obtaining the estimate of the root of the equation *f(x)*=0 is to make a plot of the function and observe where it crosses the *x*-axis.

- Graphing the function can also indicate where roots may be and where some root-finding methods may fail:
  a) Same sign, no roots
  b) Different sign, one root
  c) Same sign, two roots
  d) Different sign, three roots



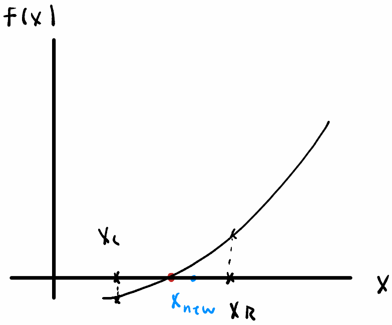


1. choose point on left + right side

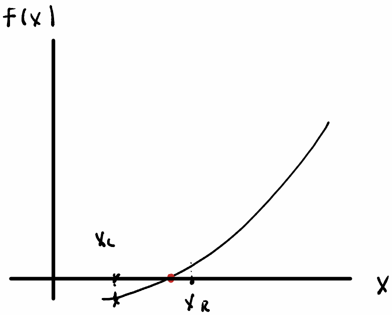


2. Find x_{new}

$$x_{new} = \frac{x_L + x_R}{2}$$



3. Assign x_{new} to L or R



if ($f(x_{new}) * f(x_R)$) > 0

$x_R = x_{new}$

else

$x_L = x_{new}$

4. Repeat until root is found

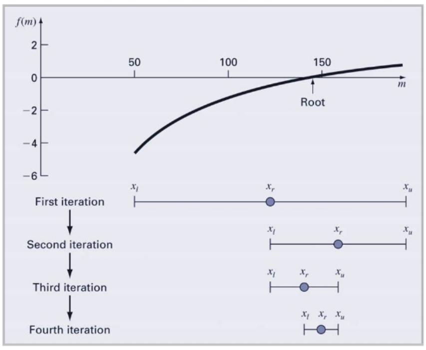
Bisection

The ***bisection method*** is a variation of the incremental search method in which the interval is always divided in half.

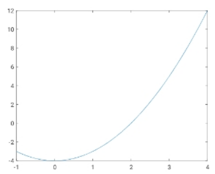
If a function changes sign over an interval, the function value at the midpoint is evaluated.

The location of the root is then determined as lying within the subinterval where the sign change occurs.

The absolute error is reduced by a factor of 2 for each iteration.



Bisection: Example



$$E^n = \frac{\Delta x^0}{2^n}$$

$$n = \frac{\ln\left(\frac{\Delta x^0}{E^n}\right)}{\ln(2)}$$

- $f(x) = x^2 - 4$
- Step 1: Determine the initial conditions:

$$x_l = 0 \rightarrow f(x_l) = -4$$
$$x_u = 3 \rightarrow f(x_u) = 5$$
$$f(x_l)f(x_u) < 0$$

- Step 2: Halve the bracket and determine the value of $f(x)$ at $x_r = 0.5(x_l + x_u)$

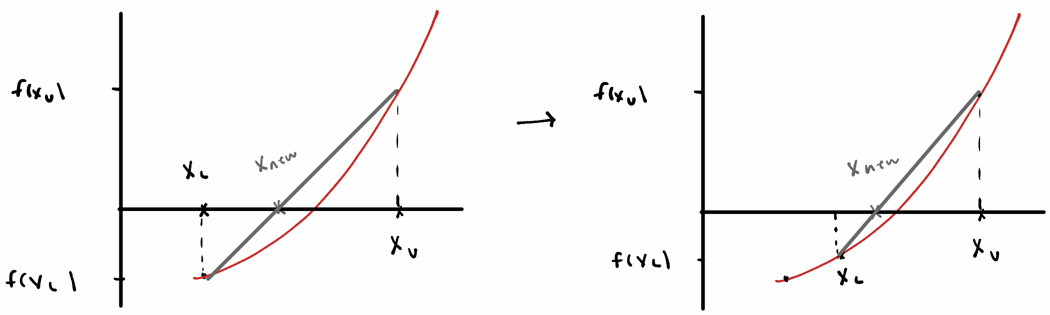
$$x_r = 0.5(x_l + x_u) = 1.5 \rightarrow f(x_r = 1.5) = -1.75$$

- Step 3: Based on the sign of $f(x_r)$, Replace x_r with x_l or x_u
  - If $f(x_r)\ f(x_u) > 0$, then the root lies between x_r and x_l, then $x_l = x_l$ and $x_u = x_r$
  - If $f(x_r)\ f(x_l) > 0$, then the root lies between x_r and x_u, then $x_u = x_u$ and $x_l = x_r$

$$f(1.5)f(3) < 0 \rightarrow x_l = 1.5$$
error=|f(1.5)|=|1.75|

- Step 4: Go to Step 2 until error is sufficiently small

Faster way to find root



$$\frac{f(x_u) - f(x_L)}{x_u - x_L} = \frac{f(x_u) - \cancel{f(x_{new})}^{0}}{x_u - x_{new}}$$

$$\frac{f(x_u) - f(x_L)}{x_u - x_L}(x_u - x_{new}) = f(x_u)$$

$$x_u - x_{new} = \frac{f(x_u)(x_u - x_L)}{f(x_u) - f(x_L)}$$

$$x_{new} = x_u - \frac{f(x_u)(x_u - x_L)}{f(x_u) - f(x_L)}$$

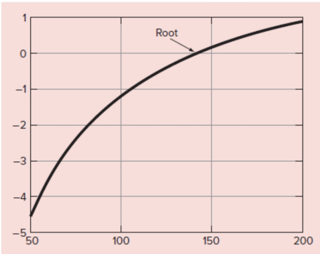
$$x_{new} = x_u - \frac{f(x_u)(x_u - x_L)}{f(x_u) - f(x_L)}$$

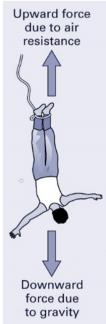
false position

False Position and Bisection Example

Use the bisection and false position approach to determine the mass of the bungee jumper with a drag coefficient of 0.25 kg/m to have a velocity of 36 m/s after 4 s of free fall. Note: The acceleration of gravity is 9.81 m/s^2.

$$v(t) = \sqrt{\frac{gm}{c_d}}\tanh\left(\sqrt{\frac{gc_d}{m}}\,t\right)$$





Bisection

1. Matlab: Plot function

```
Cd = 0.25;
g = 9.8;
v = 36;
t = 4;
func = @(m) sqrt(g*m /Cd) *tanh(sqrt(g*cd/m)*t)-v;
x = 50:1:200;

for i = 1:151
    y(i) = func(x(i));
end

plot(x,y);
```

2. Matlab: Find Root

```
x_left = 100;
x_right = 200;
iter = 0;

                                          your error
while (abs(x-right - x_left) > 10^-4)
    iter = iter + 1;
    x_new = (x_left + x_right)/2;
    if func(x_new) * func(x_left)>0
        x_left = x_new;
    else
        x_right = x_new;
    end
end
```

False position

$$x_{new} = x_U - \frac{f(x_U)(x_U - x_L)}{f(x_U) - f(x_L)}$$

```
while (abs(x-right - x-left) > 10^-4)
    iter = iter + 1;
    x-new = x-right - (func(x-right) * (x-right - x-left)) / func(x-right)
                                                          - func(x-left);
    if func(x-new) * func(x-left) > 0
        x_left = x-new;
    else
        x-right = x-new;
    end
end
```