CMPT-439
Numerical Computation

Fall 2020

Solving Nonlinear Equations
Bisection Method

The problem

- It is not a problem to find the exact solution of a quadratic equation

$$ax^2 + bx + c = 0 \quad \Rightarrow \quad x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- But what should we do with

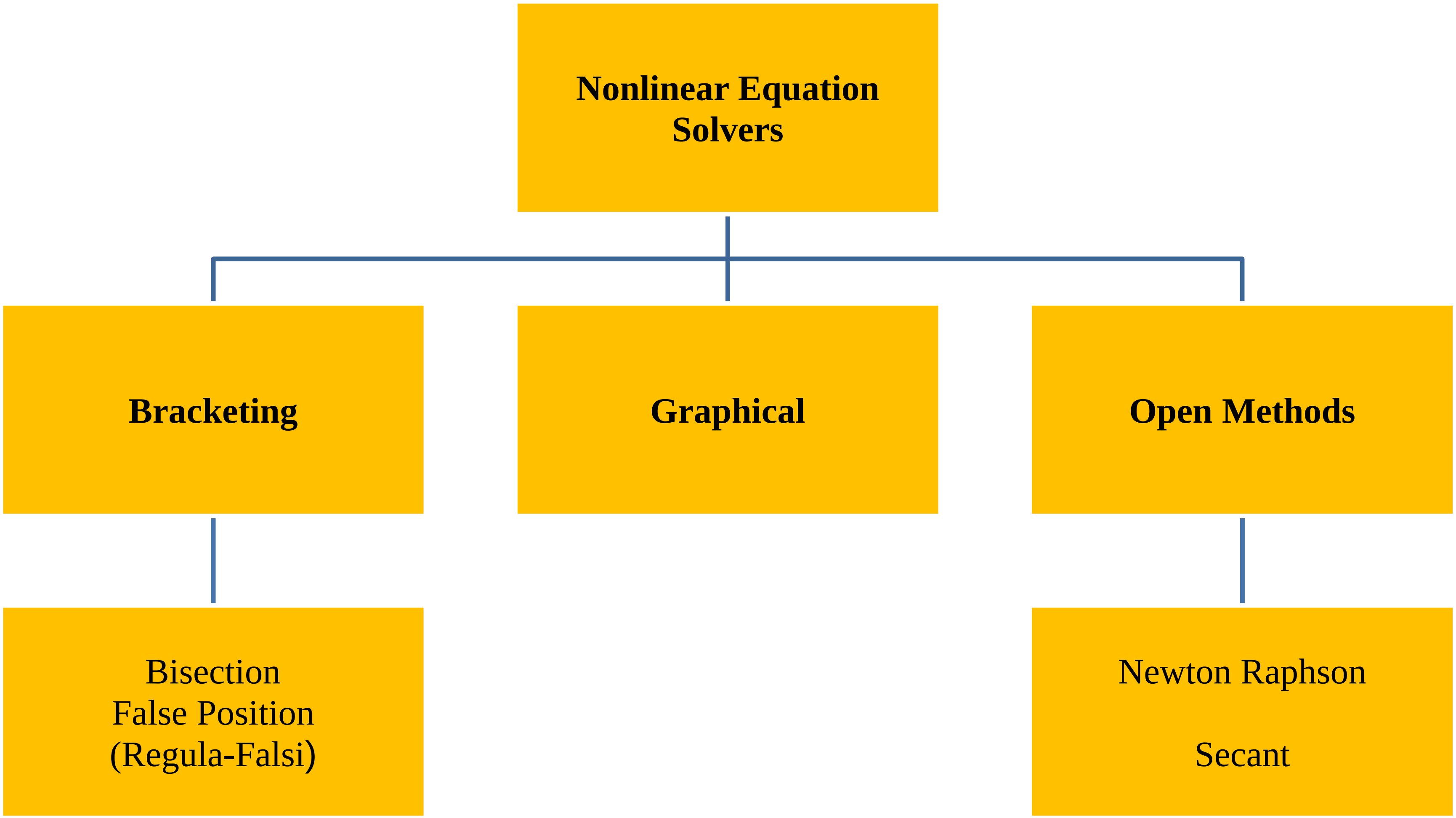
$$ax^5 + bx^4 + cx^3 + dx^2 + ex + r = 0 \quad \Rightarrow x = ?$$

$$\sin x + x = 0 \quad \Rightarrow x = ?$$

$$f(x) = 0 \quad \Rightarrow x = ?$$

The problem

- It is very difficult (and usually impossible) to find the exact roots of nonlinear equations
- Numerical methods should be applied to solve this problem



All Methods are iterative

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.

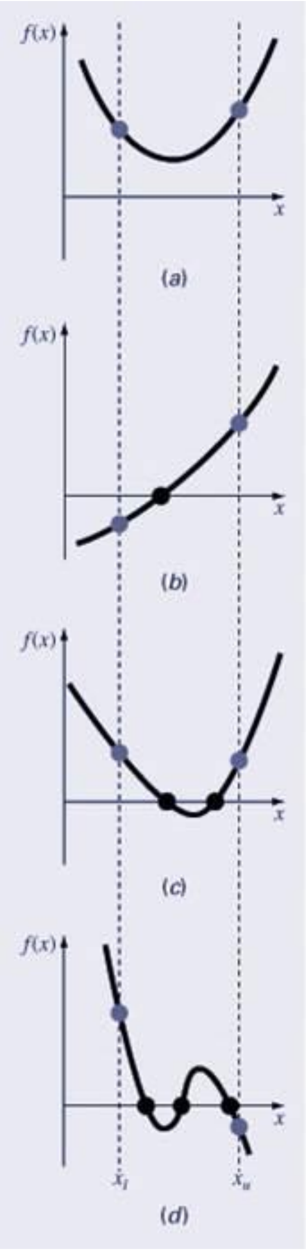
Graphing the function can also indicate where roots may be and where some root-finding methods may fail:

a) Same sign, no roots

b) Different sign, one root

c) Same sign, two roots

d) Different sign, three roots



Nonlinear Equation Solvers

- **Bracketing**
  - Bisection
  - False Position (Regula-Falsi)
- **Graphical**
- **Open Methods**
  - Newton Raphson
  - Secant

All Methods are iterative

Bracketing Methods

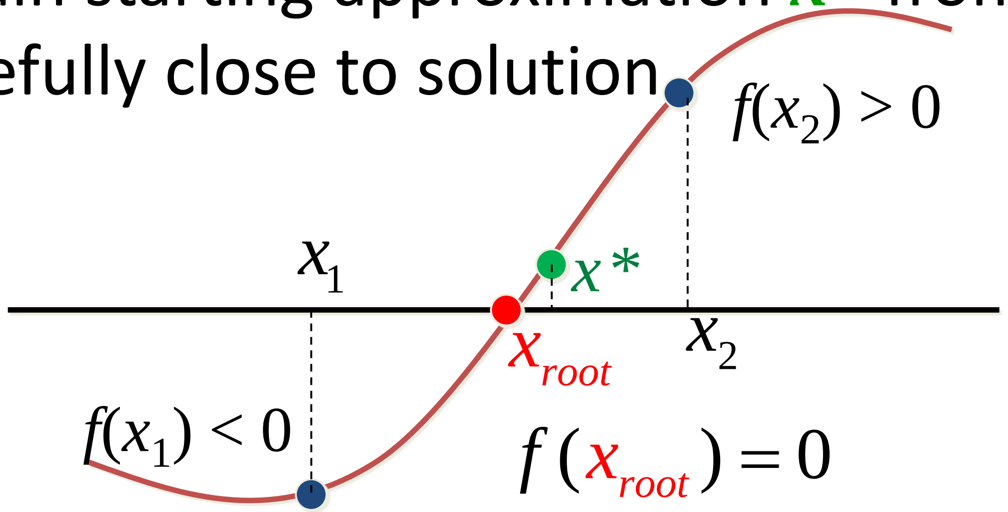
Bracketing methods are based on making two initial guesses that “bracket” the root - that is, are on either side of the root.

Brackets are formed by finding two guesses x_l and x_u where the sign of the function changes; that is, where $f(x_l)\, f(x_u) < 0$

The ***incremental search*** method tests the value of the function at evenly spaced intervals and finds brackets by identifying function sign changes between neighboring points.

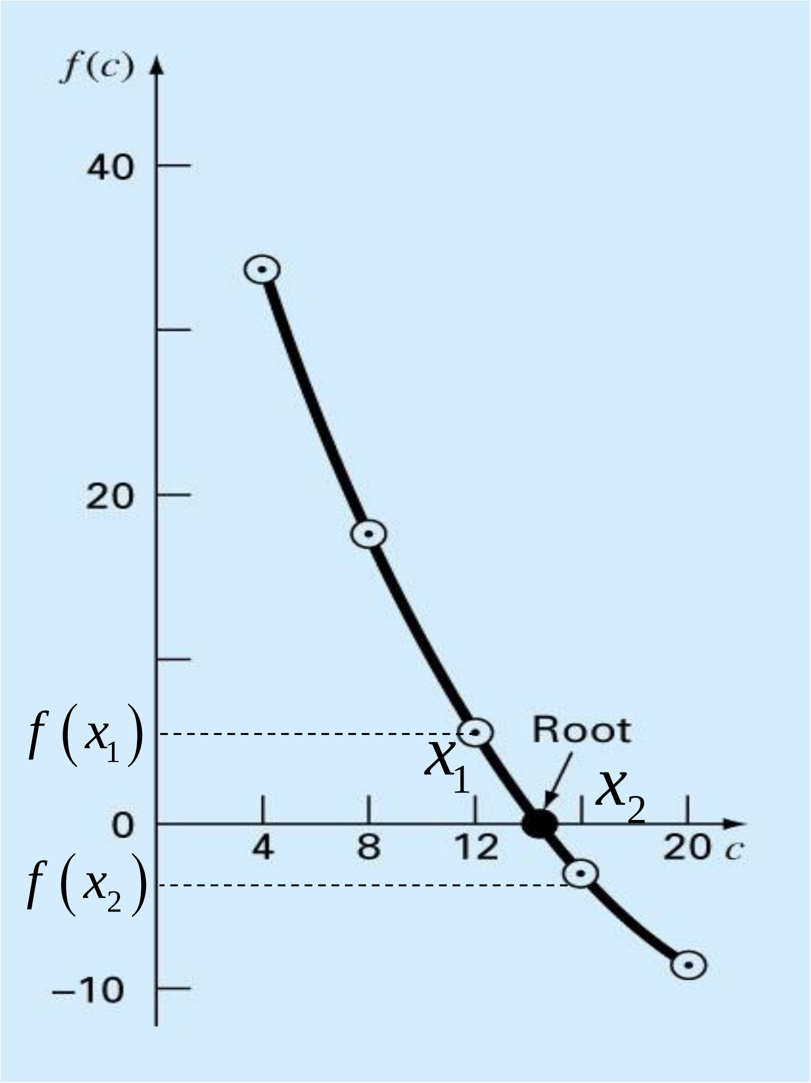
Root Finding using Bracketing

- Given some function, find location where $f(x)=0$
- Need:
  - Starting points x_1, x_2 that bracket the root
  - Obtain starting approximation x^* from x_1, x_2 , hopefully close to solution

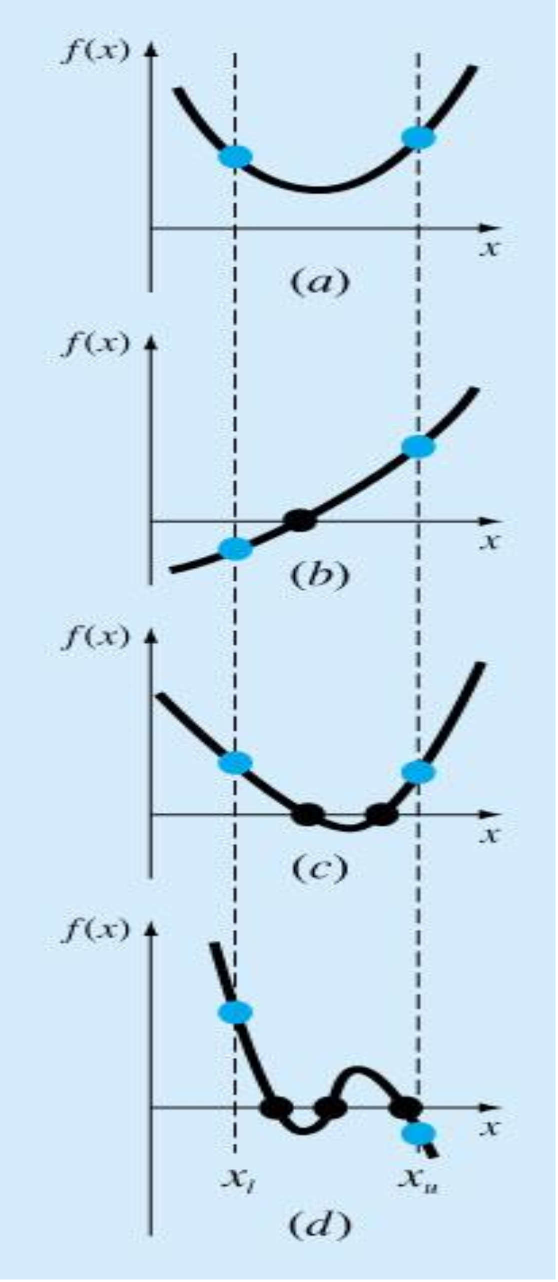


Basis of Bisection Method

- Theorem. An equation $f(x)=0$, where $f(x)$ is a real continuous function, has at least one root between x_1 and x_2 if $f(x_1)\, f(x_2) < 0$ (the function changes sign on opposite sides of the root)
- So at least one root of the equation $f(x)=0$ exists between the two points if the function $f(x)$ is real, continuous, and changes sign on the interval $\left[x_1, x_2\right]$



Some Examples:



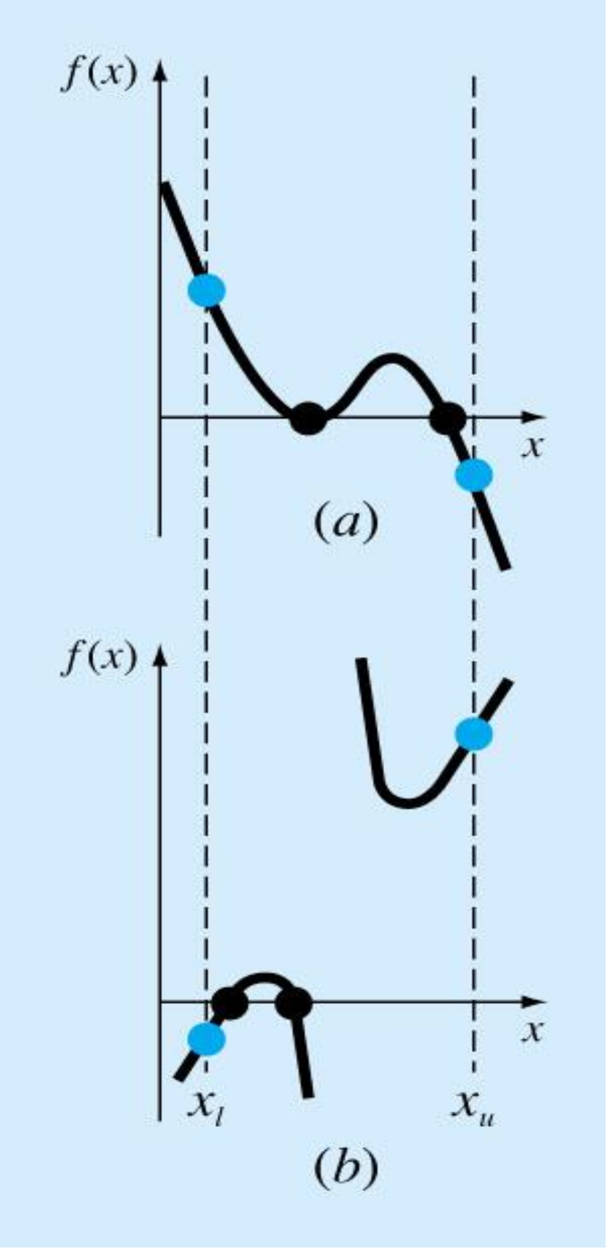
No root

Nice case (one root)

Oops!! (two roots!!)

Three roots(Might work for a while!!)

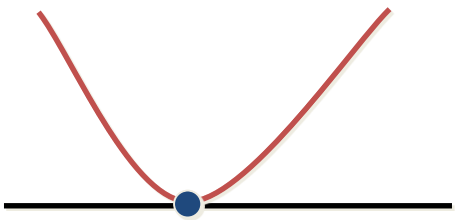
Some Examples:



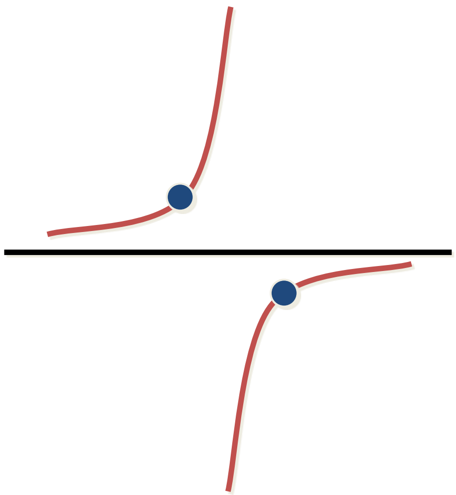
Two roots(Might work for a while!!)

Discontinuous function. Need special method

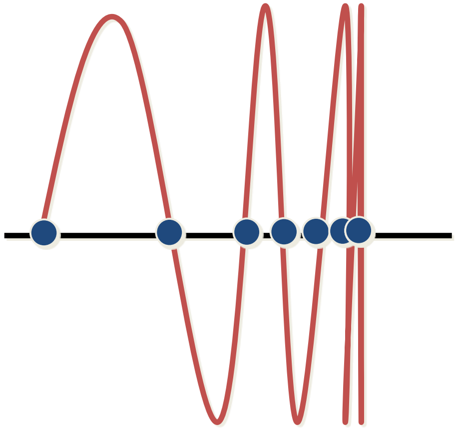
Complicated cases



Tangent point:
very difficult
to find



Singularity:
brackets don’t
surround root



Pathological case:
infinite number of
roots – e.g. sin(1/x)

Bisection Method (Interval Halving)

- Suppose $f\left(x_1\right) < 0$ and $f\left(x_2\right) > 0$
- Thus points x_1 and x_2 bracket a root. Find

  $$x_{half} = (x_1 + x_2)/2$$

  and evaluate $f(x_{half})$
- If $f\left(x_{half}\right) > 0$, set $x_2 = x_{half}$ else set $x_1 = x_{half}$
- Stop when x_1 and x_2 are close enough or when $f\left(\left(x_1 + x_2\right)/2\right)$ is close to 0.
- If the function f is continuous, this will definitely succeed in finding **some** root

Bisection Method: Algorithm

For the arbitrary equation of one variable, $f(x)=0$

1. Pick x_1 and x_2 such that they bound the root of interest, check if $f(x_1)f(x_2) < 0$

2. Estimate the root by evaluating $f\left[\left(x_1 + x_2\right)/2\right]$

3. Narrow the interval:
   - If $f\left(x_1\right) f\left[\left(x_1 + x_2\right)/2\right] < 0$, then root lies in the lower subinterval, then $x_2 = \left(x_1 + x_2\right)/2$

- If $f(x_1)\,f\big((x_1+x_2)/2\big) > 0$, root lies in the upper subinterval, then $x_1 = (x_1+x_2)/2$
- If $f(x_1)\,f\big((x_1+x_2)/2\big) = 0$, then root is $(x_1+x_2)/2$ and terminate.

4. Compare the ***absolute approximate error*** ε_a or ***absolute relative approximate error*** ε_r with the pre-determined threshold δ
5. If $\varepsilon < \delta$, then root is $(x_1+x_2)/2$ and terminate; Otherwise go to step 2 and repeat the process.

ε_a

$$\left| x_1 - \frac{x_1+x_2}{2} \right|$$

or

$$\left| x_2 - \frac{x_1+x_2}{2} \right|$$

$$\frac{\left| x_1 - \frac{x_1+x_2}{2} \right|}{\left| \frac{x_1+x_2}{2} \right|}$$

or

$$\frac{\left| x_2 - \frac{x_1+x_2}{2} \right|}{\left| \frac{x_1+x_2}{2} \right|}$$

ε_r

Variation for steps 4 and 5

- If $f(x_1)\,f\left(\left(x_1+x_2\right)/2\right)>0$, root lies in the uppersubinterval, then $x_1=\left(x_1+x_2\right)/2$
- If $f(x_1)\,f\left(\left(x_1+x_2\right)/2\right)=0$, then root is $\left(x_1+x_2\right)/2$ and terminate.

4. Estimate the ***true absolute error*** ε_t and compare it with the pre-determined threshold δ
5. If $\varepsilon_t<\delta$, then root is $\left(x_1+x_2\right)/2$ and terminate; Otherwise go to step 2 and repeat the process.

ε_t

$f\left(\left(x_1+x_2\right)/2\right)<\delta$

Programming Bisection

```
function [root,ea,iter]=bisect(func,xl,xu,es,maxit,varargin)
% bisect: root location zeroes
%   [root,ea,iter]=bisect(func,xl,xu,es,maxit,p1,p2,...):
%      uses bisection method to find the root of func
% input:
%   func = function handle
%   xl, xu = lower and upper guesses
%   es = desired relative error (default = 0.0001%)
%   maxit = maximum allowable iterations (default = 50)
%   p1,p2,... = additional parameters used by func
% output:
%   root = real root
%   ea = approximate relative error (%)
%   iter = number of iterations

if nargin<3,error('at least 3 input arguments required'),end
test = func(xl,varargin{:})*func(xu,varargin{:});
if test>0,error('no sign change'),end
if nargin<4||isempty(es), es=0.0001;end
if nargin<5||isempty(maxit), maxit=50;end
iter = 0; xr = xl;
while (1)
  xrold = xr;
  xr = (xl + xu)/2;
  iter = iter + 1;
  if xr ~= 0,ea = abs((xr - xrold)/xr) * 100;end
  test = func(xl,varargin{:})*func(xr,varargin{:});
  if test < 0
    xu = xr;
  elseif test > 0
    xl = xr;
  else
    ea = 0;
  end
  if ea <= es || iter >= maxit,break,end
end
root = xr;
```

Bisection: Analysis

- Very robust method
- Convergence rate:
  - Error bounded by the size of $[x_1, x_2]$ interval
  - Interval shrinks in half at each iteration!
  - Therefore, error cut in half at each iteration:

$$|\varepsilon_{n+1}| = ½\,|\varepsilon_n|$$

  - This is called "linear convergence"

Convergence of an iterative algorithm

- Convergence of an iterative algorithm means that the algorithm stops in some natural way: a solution can be found with some accuracy (usually the accuracy of the solution is determined by a tolerable error)
- Thus the algorithm converges when the error drops below some reasonable pre-determined threshold value (tolerance)

Bisection: how many iterations it takes?

- Length of the first interval $L_0 = |x_1 - x_2|$
- After 1 iteration $L_1 = L_o/2$
- After 2 iterations $L_2 = L_o/4$
- …
- After n iterations $L_n = L_o/2^n$

Bisection: how many iterations it takes?

- After n iterations $L_n = L_o / 2^n$
- Since the pre-determined threshold δ for the ***absolute approximate error*** cannot be larger than the length of the interval, then, knowing δ, it is always easy to predict the number of iterations $\boldsymbol{n}$ if absolute approximate error is used to estimate accuracy of our solution:

$$\delta = \frac{L_0}{2^n} \Rightarrow 2^n = \frac{L_0}{\delta} \Rightarrow n = \log_2 \frac{L_0}{\delta}$$

Bisection: Advantages

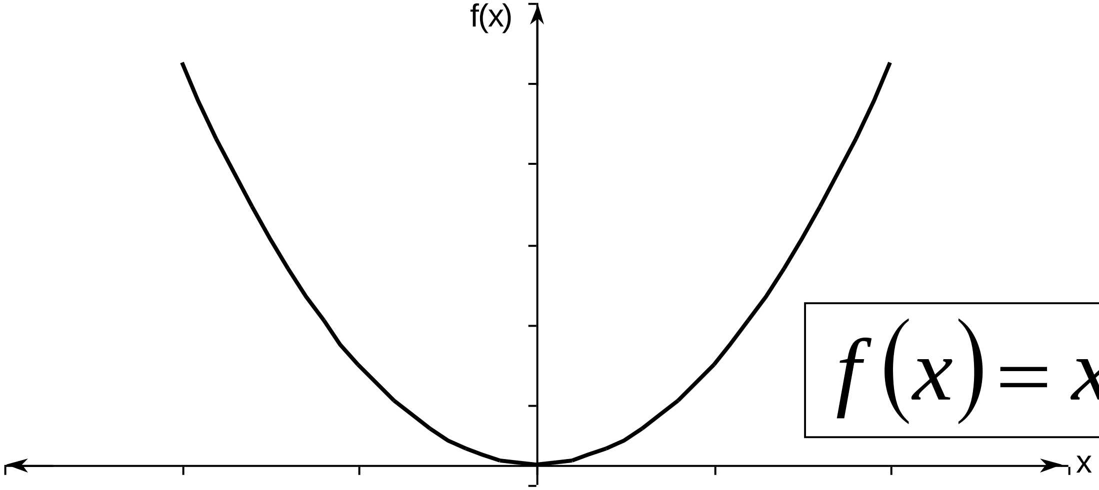
- Always converges regardless of the pre-determined threshold value if a root was initially bracketed correctly
- The root bracket gets halved with each iteration - guaranteed

Bisection: Drawbacks

- Slow convergence (usually more iterations are needed that for other methods to be considered
- If one of the initial guesses is too close to the root, the convergence is slower

Bisection: Drawbacks

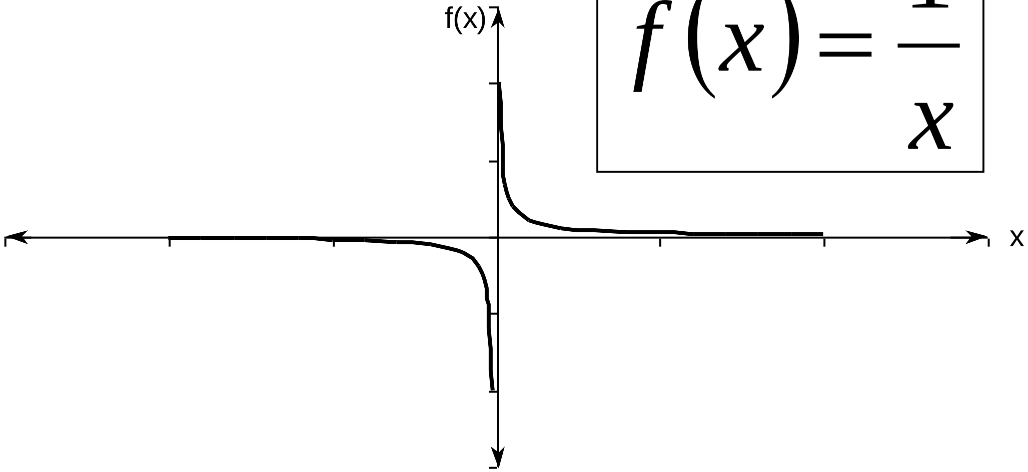
- If a function $f(x)$ is such that it just touches the x-axis, it will not be possible to find the lower and upper guesses.



$$f(x) = x^2$$

Bisection: Drawbacks

- Function changes sign but no root exists



$$f(x) = \frac{1}{x}$$

Nonlinear Equation Solvers

- **Bracketing**
  - Bisection
  - **False Position (Regula-Falsi)**
- **Graphical**
- **Open Methods**
  - Newton Raphson
  - **Secant**

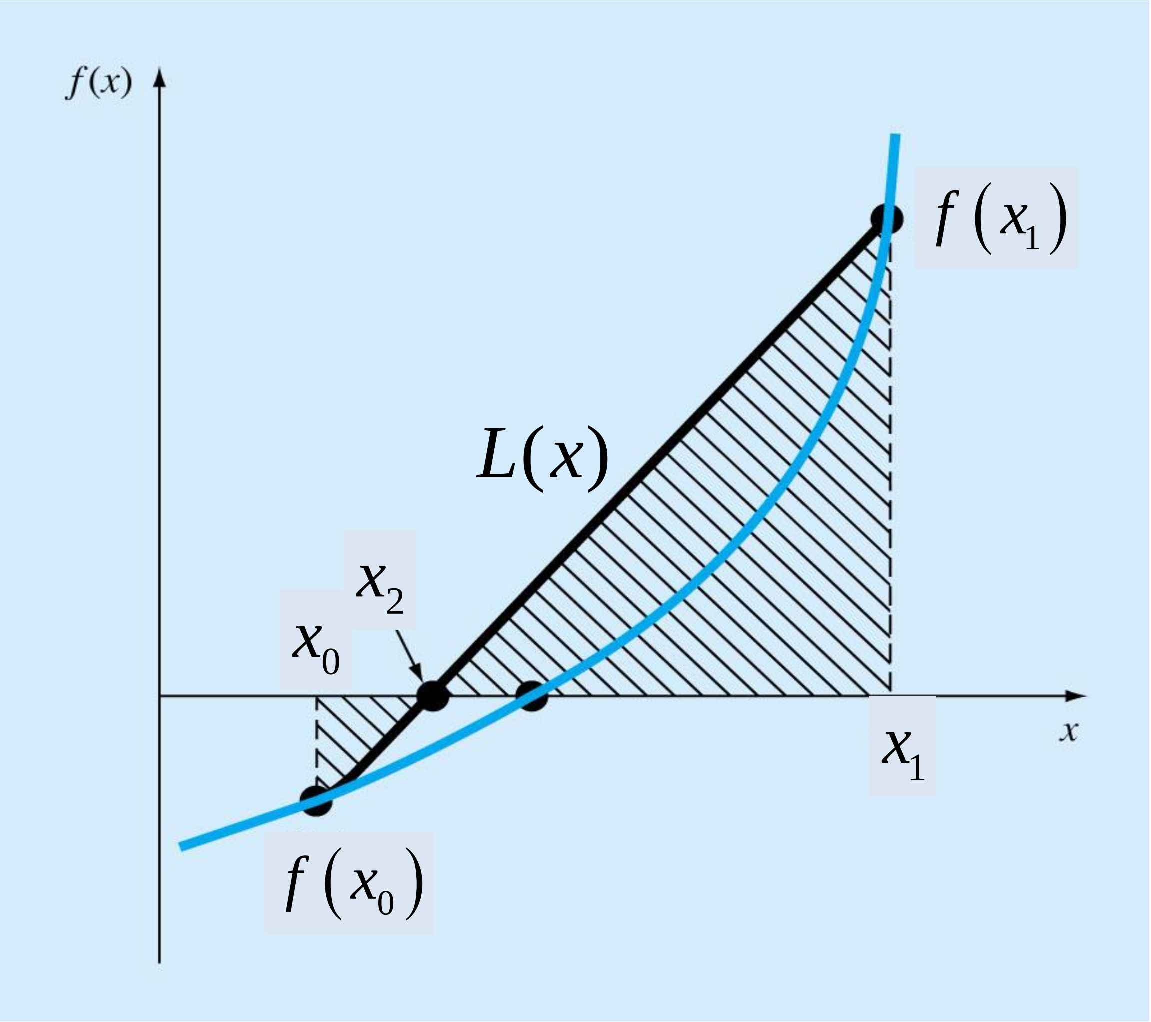
All Methods are iterative

The False Position Method

- The method of false position takes care of bracketing a root using x_o, x_1 chosen such that $f(x_o)$, $f(x_1)$ are of opposite sign (in the same way as for the bisection method)
- Unlike the bisection method, this method determines its step taking not the midpoint between the two x-values, but taking the intersection point of a line between the pair of x-values and the x-axis

The False-Position Method
(Regula-Falsi)

- If a real root is bounded by x_0 and x_1 of $f(x)=0$, then we can approximate the solution by doing a linear interpolation between the points $[x_0, f(x_0)]$ and $[x_1, f(x_1)]$ to find the x_2 value such that $L(x_2)=0$, where $L(x)$ is the linear approximation of $f(x)$

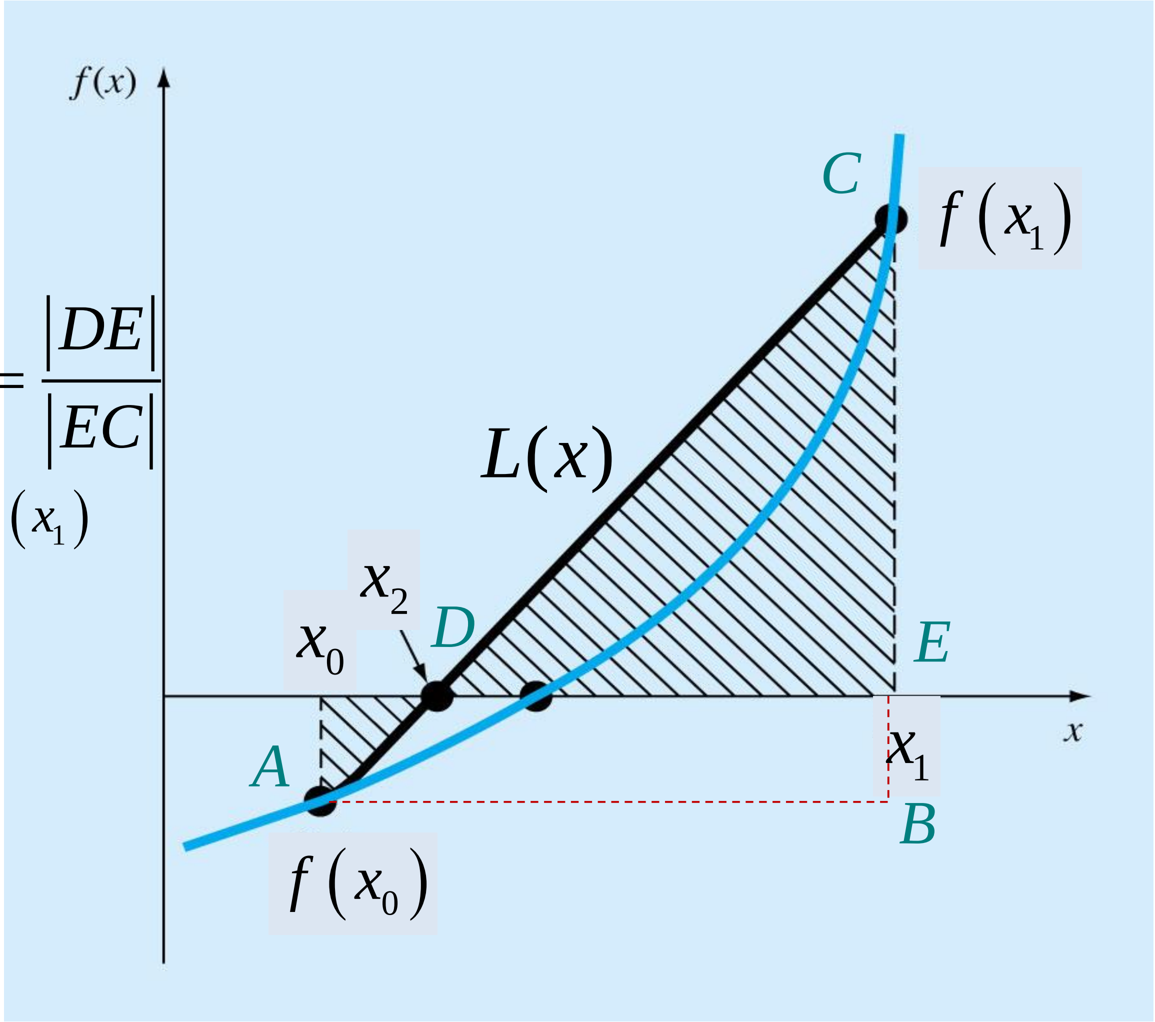


The False-Position Method
(Regula-Falsi)

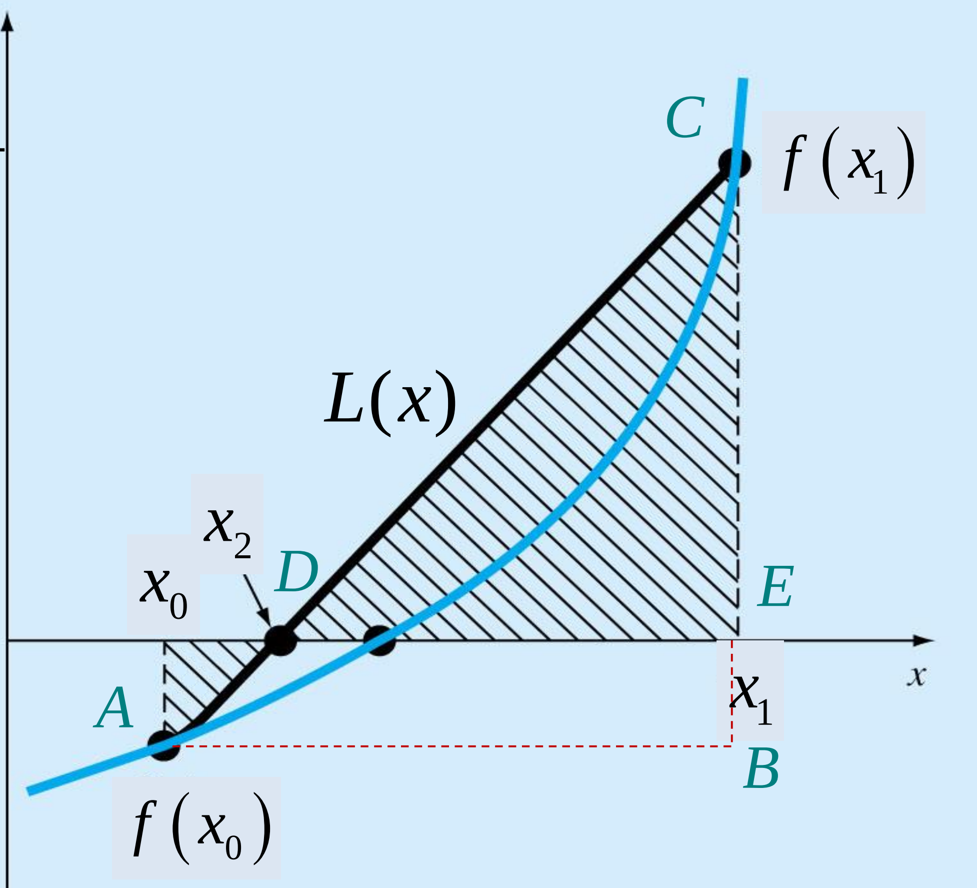
$$\triangleleft\, ABC \sim \triangleleft\, DEC \rightarrow \frac{|AB|}{|BC|} = \frac{|DE|}{|EC|}$$

$$|AB| = x_0 - x_1 \quad |BC| = f(x_0) - f(x_1)$$

$$|DE| = x_1 - x_2 \quad |EC| = f(x_1)$$



The False-Position Method
(Regula-Falsi)



$$\triangleleft\, ABC \sim \triangleleft\, DEC \rightarrow \frac{|AB|}{|BC|} = \frac{|DE|}{|EC|}$$

$$|AB| = x_0 - x_1 \quad |BC| = f(x_0) - f(x_1)$$

$$|DE| = x_1 - x_2 \quad |EC| = f(x_1)$$

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

$\Downarrow$

$\dots\ \dots\ \dots$

$\Downarrow$

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

The False Position Method : Preliminaries

- The false position method should be used to find a root of $f(x)=0$
- A root should be estimated first. It can be done by plotting a graph of $f(x)$ and narrowing an interval around a point where $f(x)=0$.
- Then the initial values x_o, x_1 should be chosen such that $f(x_o)$, $f(x_1)$ are of opposite sign

The False Position Method: Algorithm

- Determine δ - the tolerance value to ensure the appropriate true or approximate accuracy of a solution
- Determine x_0 and x_1 close enough to the projected root such that $f(x_0)f(x_1) < 0$ and set $x = x_1$
- Step 1. Set $x_2 = x_1 - f(x_1)\dfrac{(x_0 - x_1)}{f(x_0) - f(x_1)}$
- Step 2. If $f(x_0)f(x_2) < 0$, then set $x_1 = x_2$ else set $x_0 = x_2$
- Step 2 (true error). If $|f(x_2)| \le \delta$, then stop and x_2 is a root, (estimation) else go to Step 1
- Alternative Step 2. If $|x - x_2| \le \delta$, then stop and x_2 is a root, (absolute approximate error*) else $x = x_2$ and go to Step 1
- *A relative approximate error should also be used here

Bisection vs False Position

- The false position method converges faster than the bisection method

- As well as the bisection method, the false position method cannot work if a root cannot be bracketed
- Bisection does not take into account the shape of the function; this can be good or bad depending on the function!

Example: $f(x) = x^{10} - 1$