Sources for presentations:

- Textbook
- http://math.fullerton.edu/mathews/numerical.html
- http://numericalmethods.eng.usf.edu
- MIT Open Courseware “Introduction to Numerical Analysis for Engineering”

The Solution of a Nonlinear Equation f(x) = 0

Example – Square root

$$x^2 - a = 0 \Rightarrow x = \sqrt{a}$$

Example:

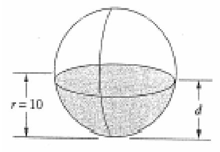


Figure 2.1 The portion of a sphere of radius r that is to be submerged to a depth d.

$$\frac{\pi(d^3 - 3d^2 r + 4r^3 \rho)}{3} = 0.$$

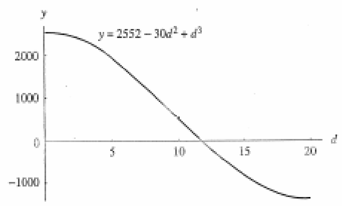


Figure 2.2 The cubic $y = 2552 - 30d^2 + d^3$.

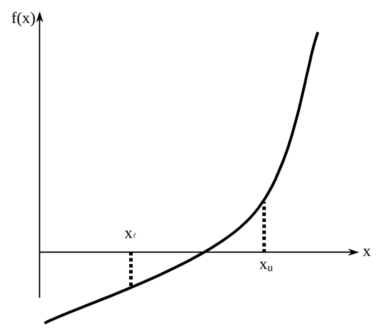
General procedure for solving nonlinear equations/ root finding/finding zero

- Plot the function
- Make an initial guess
- Iteratively refine the initial guess with a root-finding algorithm

İteration: a process is repeated until an answer is achieved.

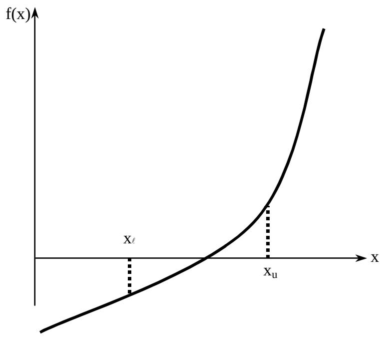
Bisection Method

An equation f(x)=0, where f(x) is a real continuous function, has at least one root between x_l and x_u if $f(x_l)\, f(x_u) < 0$.



Step 1

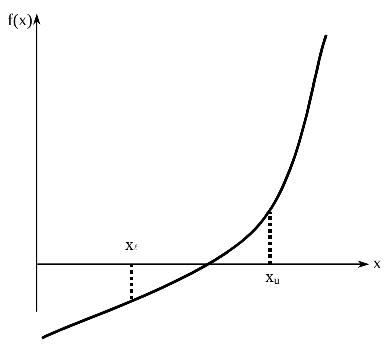
- Choose x_ℓ and x_u as two guesses for the root such that $f(x_\ell)\, f(x_u) < 0$, or in other words, f(x) changes sign between x_ℓ and x_u.



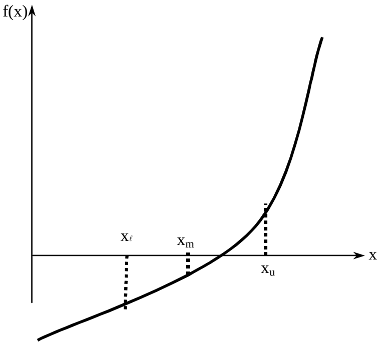
Step 2

Estimate the root, x_m of the equation f (x) = 0 as the mid-point between x_ℓ and x_u as

$$x_m = \frac{x_\ell + x_u}{2}$$



Step 3



Now check the following

If $f(x_\ell)\, f(x_m) < 0$, then the root lies between x_ℓ and x_m; then
$x_\ell = x_\ell$; $x_u = x_m$.

If $f(x_\ell)\, f(x_m) > 0$, then the root lies between x_m and x_u; then
$x_\ell = x_m$; $x_u = x_u$.

If $f(x_\ell)\, f(x_m) = 0$; then the root is x_m.
Stop the algorithm if this is true.

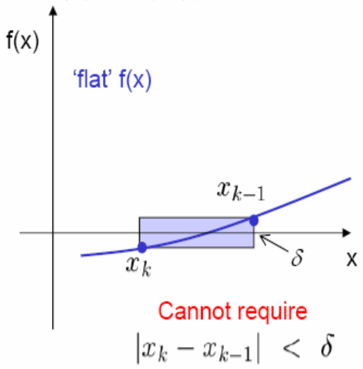
***Iteration* continues until *stopping criteria* are met.**

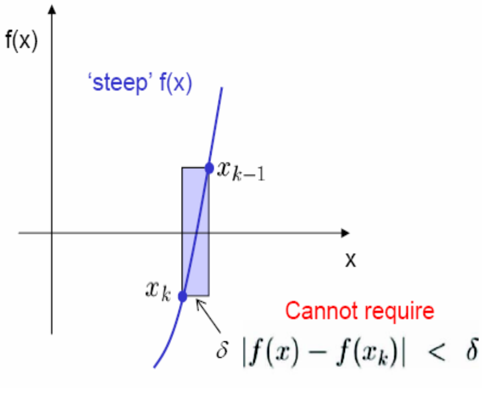
Stopping criteria:

$$|x_k - x_{k-1}| < \delta \leftarrow \text{Machine Accuracy}$$

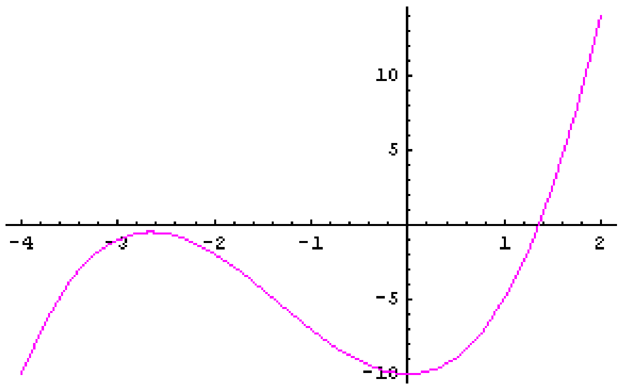
$$|f(x) - f(x_k)| < \delta$$

Use combination of the two criteria

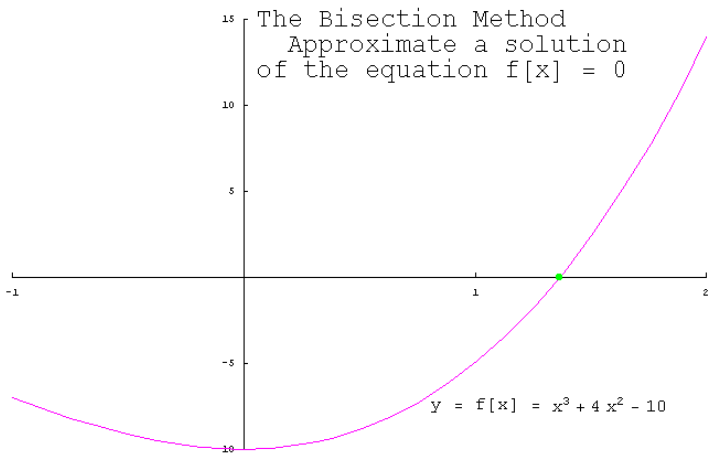




Example 1. Find all the real solutions to the cubic equation $x^3 + 4x^2 - 10 = 0$



There appears to be only one real root which lies in the interval [1,2].



Use the starting interval [a, b] = [-1, 2]

k	a_k	c_k	b_k	$f[c_k]$
0	-1.	0.5	2.	-8.875
1	0.5	1.25	2.	-1.796875
2	1.25	1.625	2.	4.853515625
3	1.25	1.4375	1.625	1.236083984375
4	1.25	1.34375	1.4375	-0.350982666015625
5	1.34375	1.390625	1.4375	0.4245948791503906
6	1.34375	1.3671875	1.390625	0.03235578536987305
7	1.34375	1.35546875	1.3671875	-0.1604211926460266
8	1.35546875	1.361328125	1.3671875	-0.06431024521589279
9	1.361328125	1.3642578125	1.3671875	-0.01604669075459242
10	1.3642578125	1.36572265625	1.3671875	0.00813717267010361
………				
29	1.36523001268506	1.365230015479028	1.365230018272996	$3.409903603923681 \times 10^{-8}$
30	1.36523001268506	1.365230014082044	1.365230015479028	$1.103008440139774 \times 10^{-8}$

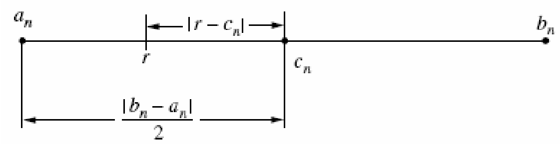
c = 1.365230014082044

Δc = $\pm 1.39698 \times 10^{-9}$

f[c] = $1.103008440139774 \times 10^{-8}$

Theorem 2.4 (Bisection Theorem). Assume that $f \in C[a, b]$ and that there exists a number $r \in [a, b]$ such that $f(r) = 0$. If $f(a)$ and $f(b)$ have opposite signs, and $\{c_n\}_{n=0}^{\infty}$ represents the sequence of midpoints generated by the bisection process of (8) and (9), then

$$|r - c_n| \le \frac{b - a}{2^{n+1}} \quad \text{for } n = 0, 1, \dots, \tag{10}$$



Observe that the successive interval widths form the pattern

$$b_1 - a_1 = \frac{b_0 - a_0}{2^1},$$

$$b_2 - a_2 = \frac{b_1 - a_1}{2} = \frac{b_0 - a_0}{2^2}.$$

It is left as an exercise for the reader to use mathematical induction and show that

$$b_n - a_n = \frac{b_0 - a_0}{2^n}. \tag{13}$$

A virtue of the bisection method is that formula (10) provides a predetermined estimate for the accuracy of the computed solution. In Example 2.7 the width of the starting interval was $b_0 - a_0 = 2$. Suppose that Table 2.1 were continued to the thirty-first iterate; then, by (10), the error bound would be $|E_{31}| \le (2-0)/2^{32} \approx 4.656613 \times 10^{-10}$. Hence c_{31} would be an approximation to r with nine decimal places of accuracy. The number N of repeated bisections needed to guarantee that the Nth midpoint c_N is an approximation to a zero and has an error less than the preassigned value δ is

$$N = \operatorname{int}\left(\frac{\ln(b-a) - \ln(\delta)}{\ln(2)}\right). \tag{15}$$

where $\left| r - c_n \right| = \delta$

Ex. If we want to reduce the error to less than 0.1% of the original interval we need N=9 iterations.

Convergence

- Solution to f(x) = 0 involves a series of approxiamations.

If $x_0, x_1, \ldots x_n \to x^e \ , \quad n \to \infty$

then the numerical method *converges* (*is convergent*)
Otherwise diverges (*is divergent*)

- We are interested in

– Conditions of convergence
– Speed of convergence

Ex. Bisection method is always convergent.

p_n

Order of Convergence

$$\lim_{n\to\infty} \frac{|p - p_{n+1}|}{|p - p_n|^R} = \lim_{n\to\infty} \frac{|E_{n+1}|}{|E_n|^R} = A$$

R : order of convergence (R>0)
A : asymptotic error constant (A≠0)
p : root
E: error

R=1 linear convergence
R=2 quadratic convergence

Ex. bisection method
has a linear convergenc(R=1,A=0.5):

$$e_n = \frac{b_n - a_n}{2}, e_{n+1} = \frac{b_{n+1} - a_{n+1}}{2} = \frac{b_n - a_n}{4}$$

$$\frac{e_{n+1}}{e_n} = \frac{1}{2}$$

Matlab Code for bisection method

```
function [c,err,yc]=bisect(f,a,b,delta)

%Input - f is the function input as a string 'f'
%       - a and b are the left and right endpoints
%       - delta is the tolerance
%Output - c is the zero
%        - yc= f(c)
%         - err is the error estimate for c

ya=feval(f,a);
yb=feval(f,b);
if ya*yb > 0,break,end
max1=1+round((log(b-a)-log(delta))/log(2));
for k=1:max1
   c=(a+b)/2;
   yc=feval(f,c);
   if yc==0
      a=c;
      b=c;
   elseif yb*yc>0
      b=c;
      yb=yc;
   else
      a=c;
      ya=yc;
   end
   if b-a < delta, break,end
end

c=(a+b)/2;
err=abs(b-a);
yc=feval(f,c);
```

Advantages

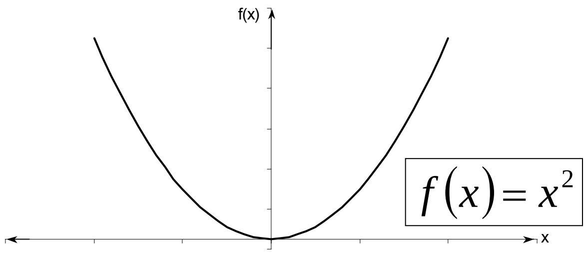
- Always convergent
- The root bracket gets halved with each iteration - guaranteed.

Drawbacks

•Slow convergence
•If one of the initial guesses is close to the root, the convergence is slower

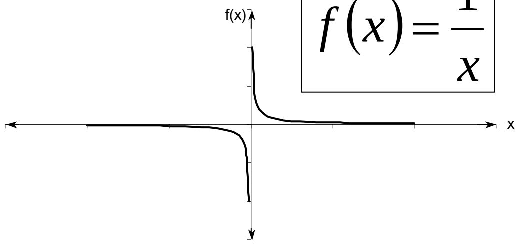
Drawbacks (continued)

- If a function f(x) is such that it just touches the x-axis it will be unable to find the lower and upper guesses.



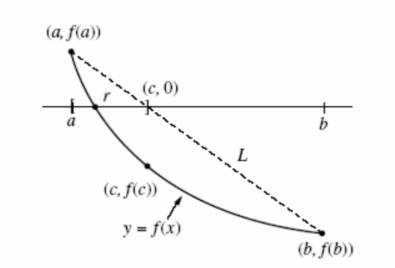
Drawbacks (continued)

- Function changes sign but root does not exist



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

Method of False Position (Regula Falsi)

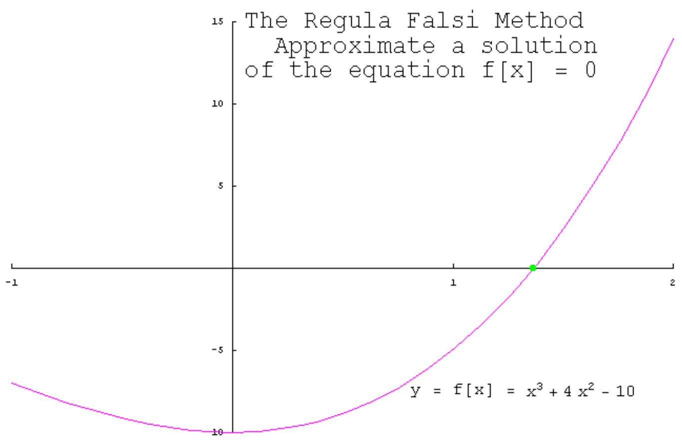


$$m = \frac{f(b) - f(a)}{b - a}, \quad m = \frac{0 - f(b)}{c - b}, \Rightarrow \frac{f(b) - f(a)}{b - a} = \frac{0 - f(b)}{c - b},$$

$$\Downarrow$$

$$c = b - \frac{f(b)(b - a)}{f(b) - f(a)}. \quad *$$

- **Start with an initial interval *braketing* the root**
- **Calculate *c* (*)**
- If $f(a)$ and $f(c)$ have opposite signs, a zero lies in $[a, c]$.
  If $f(c)$ and $f(b)$ have opposite signs, a zero lies in $[c, b]$.
  If $f(c) = 0$, then the zero is c.
- **Continue iteration until stopping criteria are satisfied**

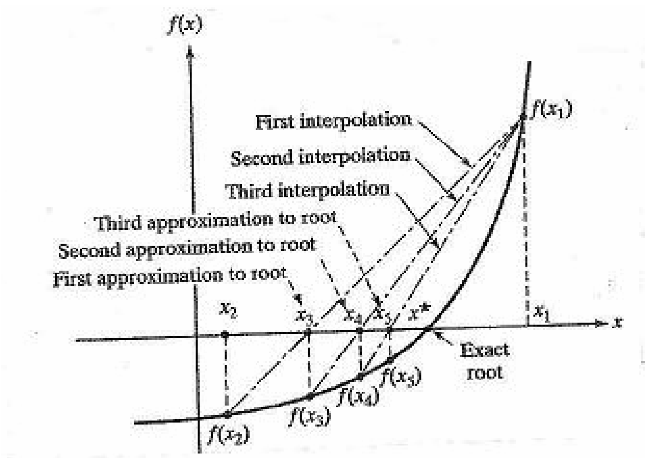


For the same stopping criteria False position converges in 15 steps (bisection 30 steps)
False position's convergence is faster than linear

c = 1.365230010769655

f[c] = $-4.366872907723973 \times 10^{-8}$

Drawback



Stationary end point for the false position method

Fixed Point Iteration

- Rewrite f(x) = 0 as x=g(x) so that finding the root of f(x) = 0 becomes equivalent to finding the fixed point of g(x).
- Start with initial guess p_0
- Iterate according to

$$p_1 = g(p_0)$$

$$p_2 = g(p_1)$$

$$\vdots$$

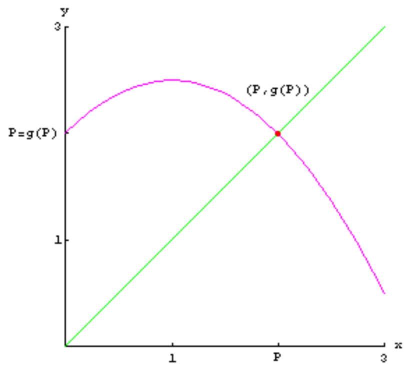
$$p_k = g(p_{k-1})$$

$$p_{k+1} = g(p_k)$$

Note 1: no initial interval – open method

Note 2: Check if the number of iterations has exceeded the maximum number of iterations (additional stopping criteria)

Fixed point *p* is the intersection of g(x) and y=x



Convergence conditions for the fixed-point iterations:

If

1. g(x) is continuous and g(x) maps [a,b] into [a,b],
2. g'(x) is continuous on [a,b], and
3. there is a number K<1 such that $|g'(x)| \leq K$ for all x in [a,b],

then

- x=g(x) has excatly one solution (say x*) in [a,b], and
- the fixed-point iteration converges to x*, for any initial guess in [a,b].

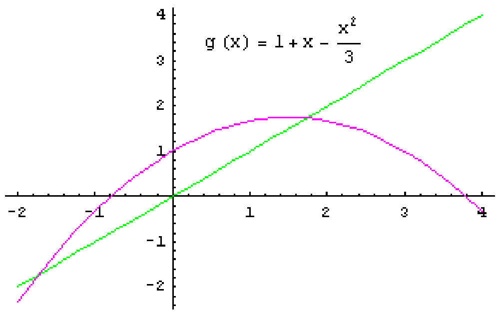
Note 1: if 1 and 2 hold, but $|g'(x)| > 1$ then fixed-point iteration diverges
Note 2: since g(x) is continuous in [a,b] we can also use
$|g'(x^*)| \leq K < 1$ and $|g'(x^*)| > 1$.

Example 2.4. Consider the iteration $p_{n+1} = g(p_n)$ when the function $g(x) = 1 + x - x^2/4$ is used. The fixed points can be found by solving the equation $x = g(x)$. The two solutions (fixed points of g) are $x = -2$ and $x = 2$. The derivative of the function is $g'(x) = 1 - x/2$, and there are only two cases to consider.

Case (i):	$P = -2$	*Case (ii):*	$P = 2$
Start with	$p_0 = -2.05$	Start with	$p_0 = 1.6$
then get	$p_1 = -2.100625$	then get	$p_1 = 1.96$
	$p_2 = -2.20378135$		$p_2 = 1.9996$
	$p_3 = -2.41794441$		$p_3 = 1.99999996$
	$\vdots$		$\vdots$
	$\lim_{n\to\infty} p_n = -\infty.$		$\lim_{n\to\infty} p_n = 2.$

Since $|g'(x)| > \frac{3}{2}$ on $[-3, -1]$, by Theorem 2.3, the sequence will not converge to $P = -2$.

Since $|g'(x)| < \frac{1}{2}$ on $[1, 3]$, by Theorem 2.3, the sequence will converge to $P = 2$.



$p_0 = 3.000000000000000$
$p_1 = 1.000000000000000$
$p_2 = 1.666666666666670$
$p_3 = 1.740740740740740$
$p_4 = 1.730681298582530$
$p_5 = 1.732262046161430$
$p_6 = 1.732018113970970$
$p_7 = 1.732055864929790$

$|g'[\sqrt{3}]| = 0.154701$ $\quad |g'[p]| < 1$

$p_0 = -2.000000000000000$
$p_1 = -2.333333333333330$
$p_2 = -3.148148148148150$
$p_3 = -5.451760402377680$
$p_4 = -14.358990897355400$
$p_5 = -82.085864094134300$

$|g'[-\sqrt{3}]| = 2.1547$ $\quad |g'[p]| > 1$

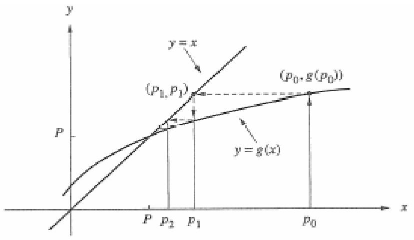


Figure 2.4 (a) Monotone convergence when $0 < g'(P) < 1$.

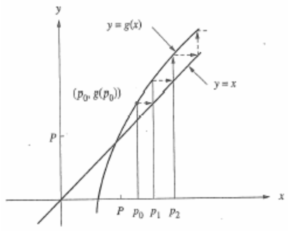


Figure 2.5 (a) Monotone divergence when $1 < g'(P)$.

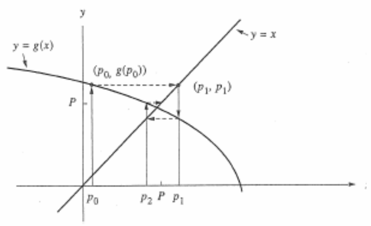


Figure 2.4 (b) Oscillating convergence when $-1 < g'(P) < 0$.

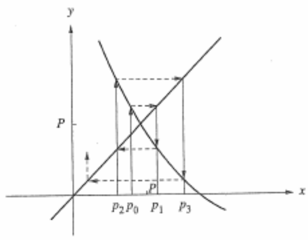
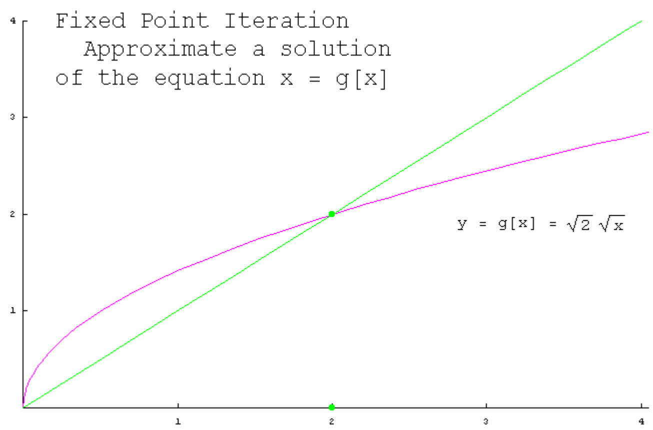
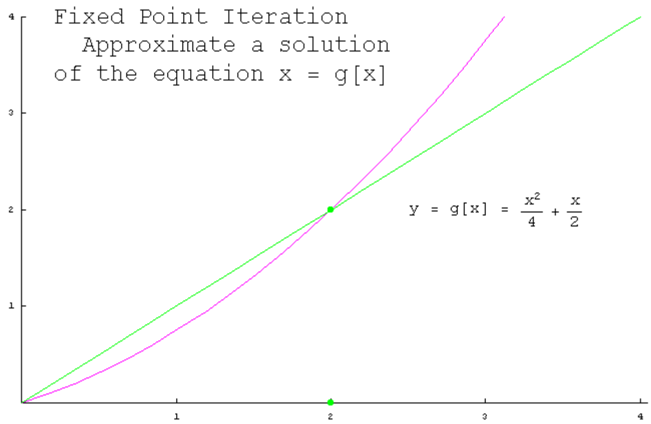


Figure 2.5 (b) Divergent oscillation when $g'(P) < -1$.

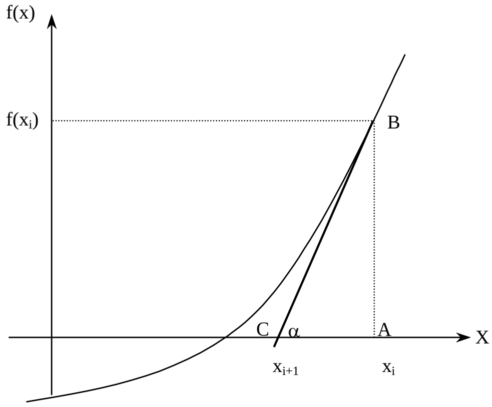
Convergence: Monotone Increasing



Divergence: Monotone Increasing



Newton-Raphson Method

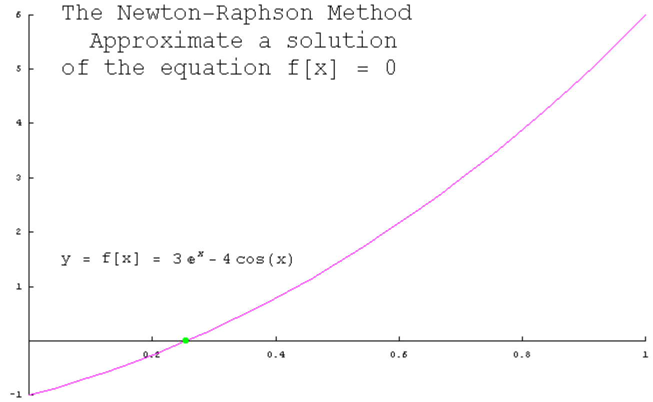


$$\tan(\alpha) = \frac{AB}{AC}$$

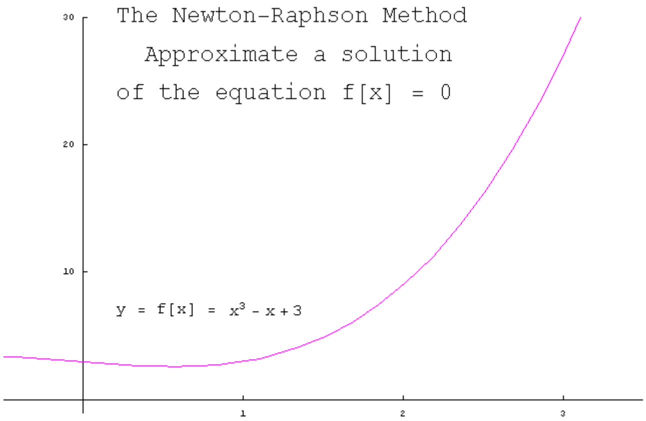
$$f'(x_i) = \frac{f(x_i)}{x_i - x_{i+1}}$$

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

Fast convergence



Cycling divergence



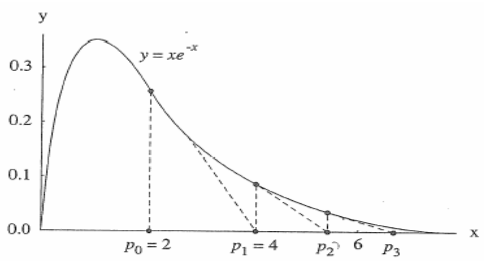


Figure 2.15 (a) Newton-Raphson iteration for $f(x) = xe^{-x}$ can produce a divergent sequence.

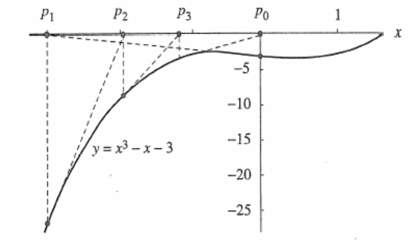


Figure 2.15 (b) Newton-Raphson iteration for $f(x) = x^3 - x - 3$ can produce a cyclic sequence.

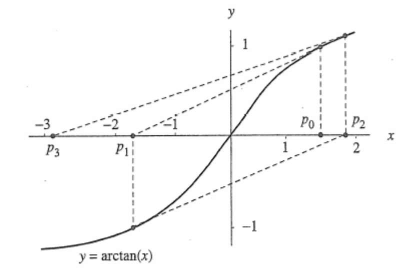
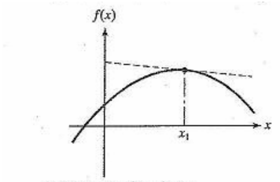


Figure 2.15 (c) Newton-Raphson iteration for $f(x) =$ arctan(x) can produce a divergent oscillating sequence.



(b) Derivative (slope) at x_1 close to zero.

Convergence rate of Newton-Raphson

- quadratic at a simple root $$|E_{k+1}| \approx \frac{|f''(p)|}{2|f'(p)|}(|E_k|)^2$$
- linear at a multiple root $$|E_{k+1}| \approx \frac{m-1}{m}|E_k|$$

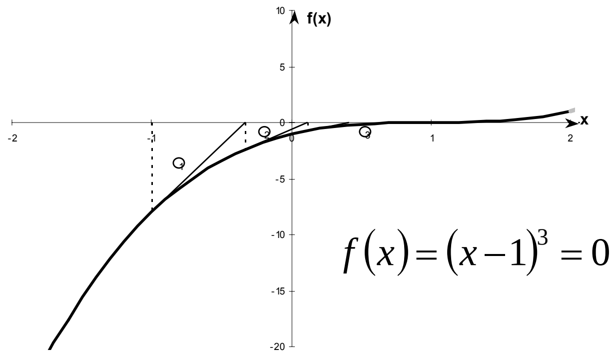
Definition : f(x) has a root of order m at x=p if

$$f(p) = 0,\ f'(p) = 0,\ f''(p) = 0,\ \ldots,\ f^{(m-1)}(p) = 0,\ f^{(m)}(p) \neq 0$$

Advantages

- Converges fast, if it converges
- Requires only one guess

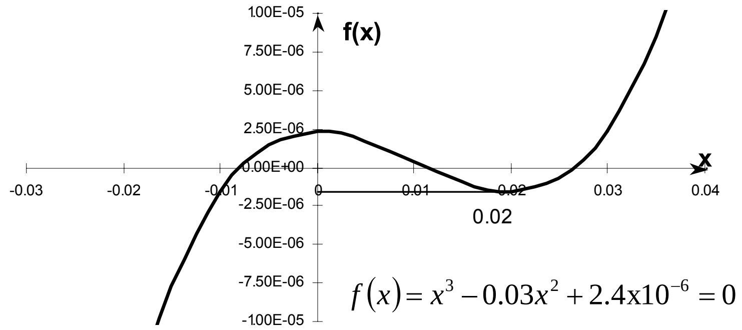
Drawbacks



$$f(x) = (x-1)^3 = 0$$

Inflection Point

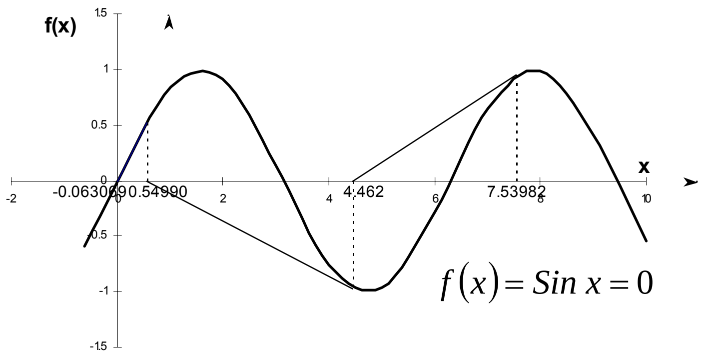
Drawbacks (continued)



$$f\left(x\right)=x^3-0.03x^2+2.4\text{x}10^{-6}=0$$

Division by zero

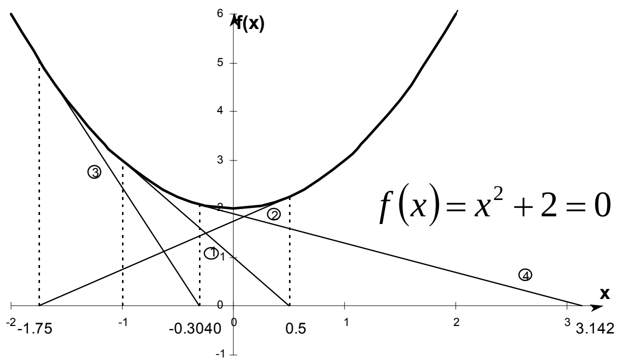
Drawbacks (continued)



$$f\left(x\right)=Sin\ x=0$$

Root Jumping

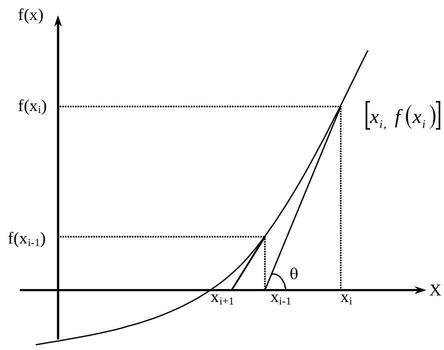
Drawbacks (continued)



$$f(x) = x^2 + 2 = 0$$

Oscillations near Local Maxima or Minima

Secant Method



Newton's Method

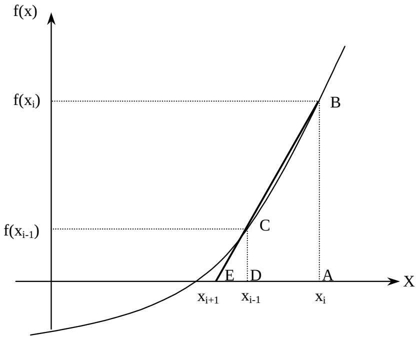
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Approximate the derivative

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

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

Secant Method



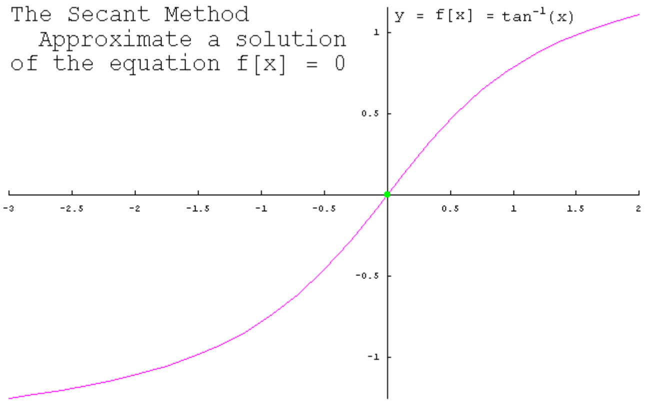
Similar Triangles

$$\frac{AB}{AE} = \frac{DC}{DE}$$

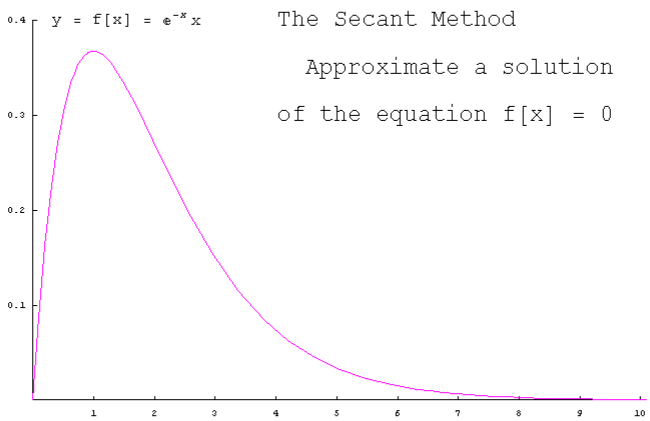
$$\frac{f(x_i)}{x_i - x_{i+1}} = \frac{f(x_{i-1})}{x_{i-1} - x_{i+1}}$$

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

Oscillating convergence



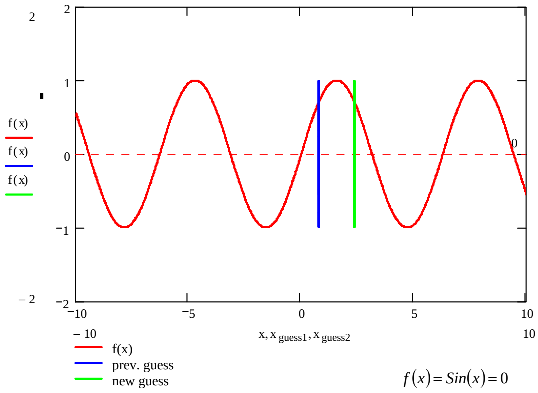
Divergence to infinity



Advantages

- Converges fast, if it converges (R=1.618)
- Requires two guesses that do not need to bracket the root

Drawbacks



$f(x) = Sin(x) = 0$

Division by zero