Numerical Methods

Finding Roots

Finding roots / solving equations

- General solution exists for equations such as

$$ax^2 + bx + c = 0$$

The quadratic formula provides a quick answer to *all* quadratic equations.

However, *no* exact *general solution (formula)* exists for equations with powers greater than 4.

Methods For Solving Nonlinear Equations Are Iterative

- generate a sequence of points $x^{(k)}$, $k = 0, 1, 2, \ldots$ that converge to a solution; $x^{(k)}$ is called the kth *iterate*; $x^{(0)}$ is the *starting point*

- computing $x^{(k+1)}$ from $x^{(k)}$ is called one *iteration* of the algorithm

- each iteration typically requires one evaluation of f (or f and f') at $x^{(k)}$

- algorithms need a stopping criterion, $e.g.$, terminate if

$$|f(x^{(k)})| \leq \text{specified tolerance}$$

- speed of the algorithm depends on:
  - the cost of evaluating $f(x)$ (and possibly, $f'(x)$)
  - the number of iterations

Roots of Nonlinear Equations
Stop-criteria

Unrealistic stop-criteria

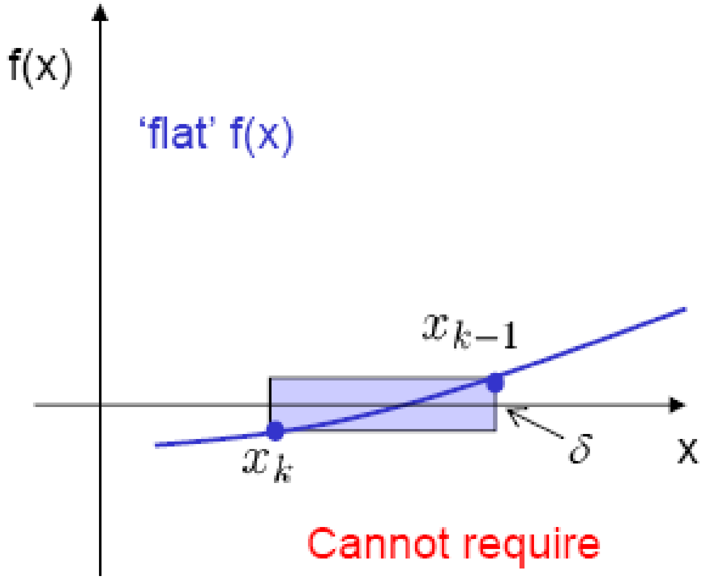
$$x_{k+1} \neq x_k$$

Realistic stop-criteria

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

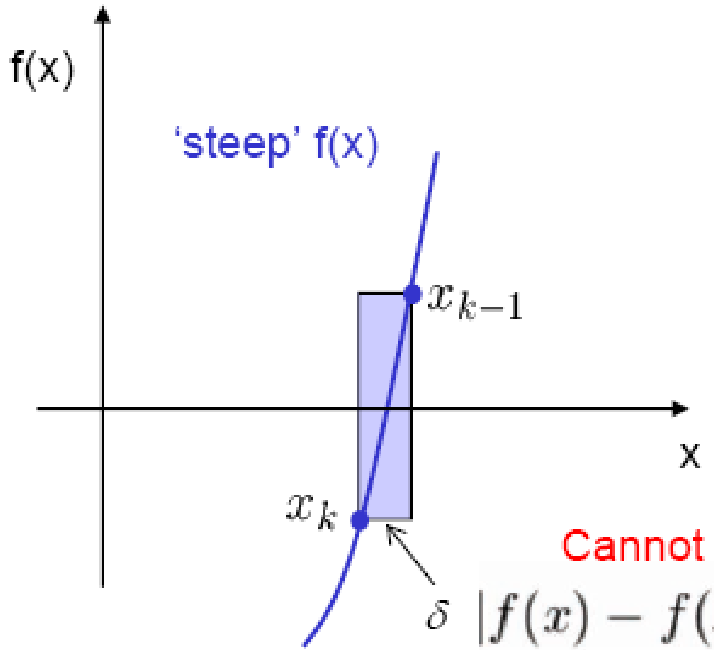
$$|f(x) - f(x_k)| \; < \; \delta$$

Use combination of the two criteria



Cannot require

$$|x_k - x_{k-1}| \; < \; \delta$$



Cannot require

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

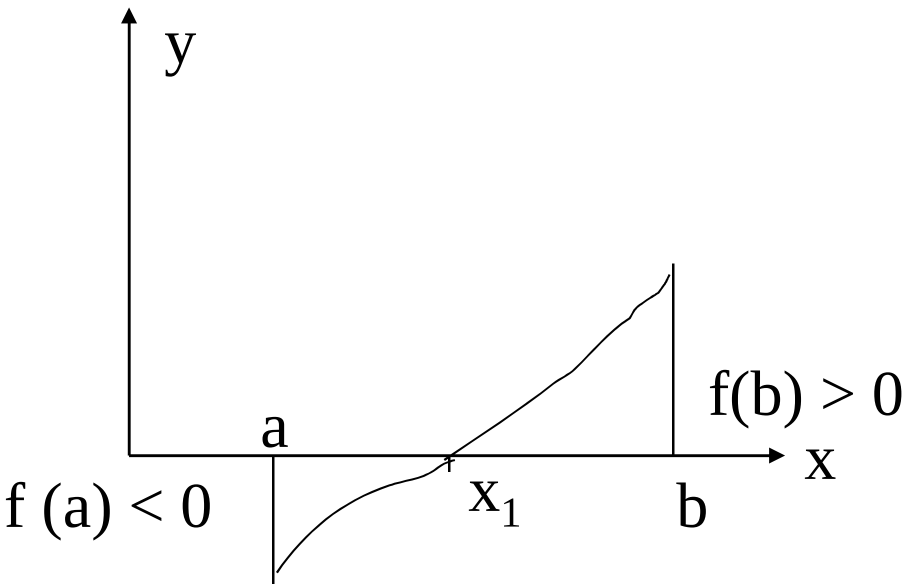
Typical stopping criteria:

- x-increment $|x_{k+1} - x_k| \leq \tau_x$
- f-value $|f(x_k)| \leq \tau_f$
- number of iterations $k \geq k_{\max}$

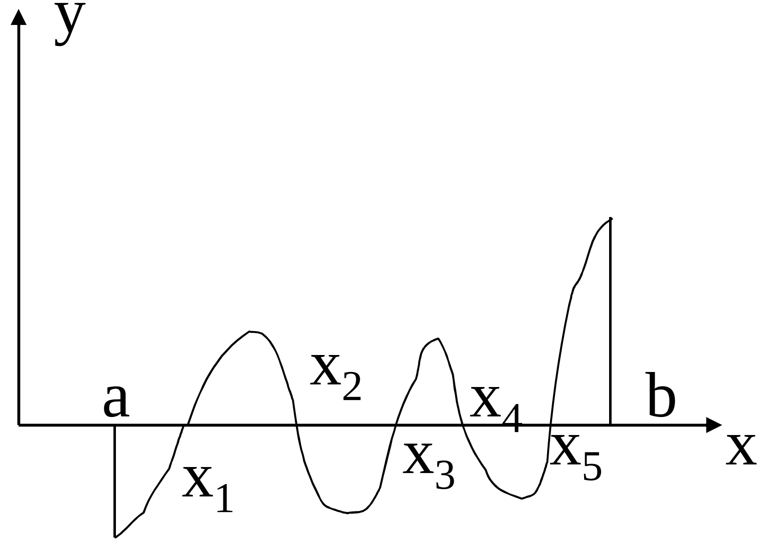
Root **Finding: f(x)=0**

Method 1: The Bisection method

Theorem: If f(x) is continuous in [a,b] and if f(a)f(b)<0, then there is atleast one root of f(x)=0 in (a,b).



Single root in (a,b)



Multiple roots in (a,b)

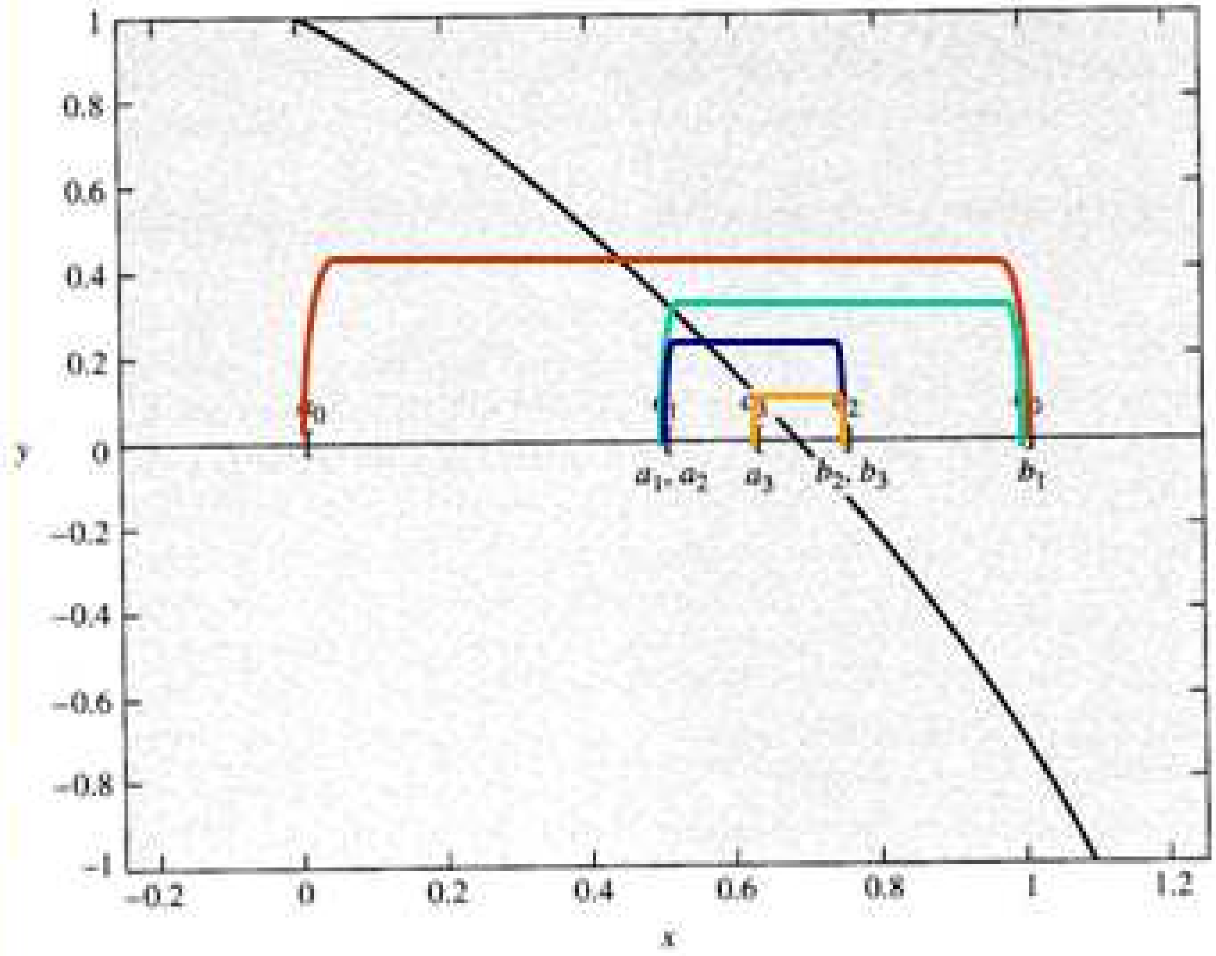


FIGURE 3.1 Bisection applied to $y = 2 - e^x$.

Bisection method

The idea for the Bisection Algorithm is to cut the interval [a,b] you are given in half (bisect it) on each iteration by computing the midpoint x_{mid}. The midpoint will replace either a or b depending on if the sign of $f(x_{mid})$ agrees with $f(a)$ or $f(b)$.

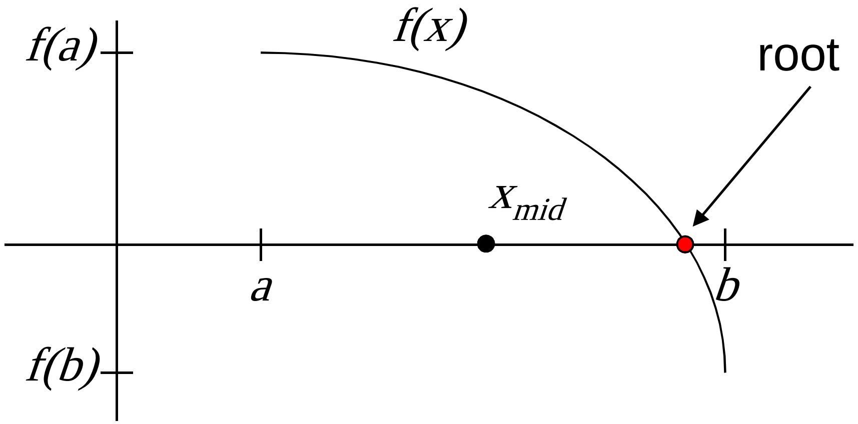
Step 1: Compute $x_{mid} = (a+b)/2$

Step 2: If $sign(f(x_{mid})) = 0$ then end algorithm

else If $sign(f(x_{mid})) = sign(f(a))$ then $a = x_{mid}$

else $b = x_{mid}$

Step 3: Return to step 1



This shows how the points a, b and x_{mid} are related.

Bisection method

- Find an interval $[x_0, x_1]$ so that $f(x_0)f(x_1) < 0$ (This may not be easy.).

- Cut the interval length into half with each iteration,by examining the sign of the function at the mid point.

$$x_2 = \frac{x_0 + x_1}{2}$$

- If $f(x_2) = 0$, x_2 is the root.
- If $f(x_2) \neq 0$ and $f(x_0)f(x_2) < 0$, root lies in $[x_0, x_2]$.
- Otherwise root lies in $[x_2, x_1]$.
- Repeat the process until the interval shrinks to a desired level.

Pseudo code (Bisection Method)

1. Input $\in > 0$, m > 0, $x_1 > x_0$ so that $f(x_0)\, f(x_1) < 0$.

   Compute $f_0 = f(x_0)$.

   k = 1 (iteration count)

2. Do

   {

   (a) Compute $f_2 = f(x_2) = f\left(\dfrac{x_0 + x_1}{2}\right)$

   (b) If $f_2 f_0 < 0$, set $x_1 = x_2$ otherwise set x_0 = x2 and $f_0 = f_2$.

   (c) Set k = k+1.

   }

3. While $|f_2| > \in$ and $k \leq m$

4. set $x = x_2$, the root.

Consider finding the root of $f(x) = x^2 - 3$. Let εstep = 0.01, εabs = 0.01 and start with the interval [1, 2].

Bisection method applied to $f(x) = x^2 - 3$.

A	b	f(a)	f(b)	c = (a + b)/2	f(f)	Update	b - a
1	2	-2	1	1.5	-0.75	a = c	0.5
1.5	2	-0.75	1	1.75	0.062	b = c	0.25
1.5	1.75	-0.75	0.0625	1.625	-0.359	a = c	0.125
1.625	1.75	-0.3594	0.0625	1.6875	-0.1523	a = c	0.0625
1.6875	1.75	-0.1523	0.0625	1.7188	-0.0457	a = c	0.0313
1.7188	1.75	-0.0457	0.0625	1.7344	0.0081	b = c	0.0156
1.71988/td>	1.7344	-0.0457	0.0081	1.7266	-0.0189	a = c	0.0078

Bisection Method: Example

xo	x1	f(xo)	f(x1)	x2	f(x2)
0	4	-7	1	2	1
0	2	-7	1	1	1
0	1	-7	1	0.5	-1.625
0.5	1	-1.625	1	0.75	-0.015625
0.75	1	-0.015625	1	0.875	0.560547
0.75	0.875	-0.015625	0.560547	0.8125	0.290283
0.75	0.8125	-0.015625	0.290283	0.78125	0.141876
0.75	0.78125	-0.015625	0.141876	0.765625	0.064274

$$f(x) = (x-1)(x-2)(x-4)+1$$

Bisection applied to f(x)=exp(x)-3x^2:

tol= 1.00e-002

Iteration	Interval	
0	[0.500	1.000]
1	[0.750	1.000]
2	[0.875	1.000]
3	[0.875	0.938]
4	[0.906	0.938]
5	[0.906	0.922]
6	[0.906	0.914]

xsol= 9.1016e-001
f(xsol)= -4.4246e-004

Number of Iterations and Error Tolerance

- Length of the interval (where the root lies) after n iterations

$$e_n = \frac{x_1 - x_0}{2^{n+1}}$$

- We can fix the number of iterations so that the root lies within an interval of chosen length $\in$ (error tolerance).

$$e_n \leq \in \quad \Rightarrow \quad n \geq \left(\frac{\ln(x_1 - x_0) - \ln \in}{\ln 2} \right) - 1$$

- Use the theorem from the course to find a bound for the number of iterations needed to achieve an approximation with accuracy 10^{-3} to the solution of $x^3 - x - 1 = 0$ lying in the interval $[1, 4]$.

$$\frac{b-a}{2^n} = \frac{3}{2^n} \leq 10^{-3},$$

$$3 \cdot 10^3 \leq 2^n \Rightarrow n \geq \frac{log_{10}(3 \cdot 10^3)}{log_{10}(2)} \approx 11.55$$

For example, if we were solving $g(x) = x^2 - 3 = 0$ starting in the interval $[1, 2]$, with a tolerance of 10^{-3}, the number of iterations needed would be the largest integer satisfying

$$
\begin{aligned}
i &\geq \frac{\log\left(\frac{2-1}{10^{-3}}\right)}{\log(2)} \\
&= \frac{\log(10^3)}{\log(2)} \\
&= \frac{3}{\log(2)} \\
&= 9.9658
\end{aligned}
$$

Thus, 10 iterations would be needed.

Find a bound for the number of Bisection method iterations needed to achieve an approximation with accuracy 10^{-9} to the solution of $x^5+x = 1$ lying in the interval $[0, 1]$.Find an approximation to the root with this degree of accuracy.

Convergence criteria

We would like $f(p_n) \approx 0$ and $p_n \approx p_{n-1}$

The criteria can be

- For the ordinate: $|f(p_n)| < \epsilon$
- For the abscissa:
  - for the absolute error: $|p_n \quad p_{n-1}| < \delta$
  - for the relative error: $\dfrac{2|p_n - p_{n-1}|}{|p_n| + |p_{n-1}|} < \delta$

May also use $N = \mathrm{ceil}\dfrac{\ln(b-a) - \ln(\delta)}{\ln(2)}$

Advantages

- Always convergent
- The root bracket gets halved with each iteration -it is guaranteed to converge under its assumptions,

Drawbacks

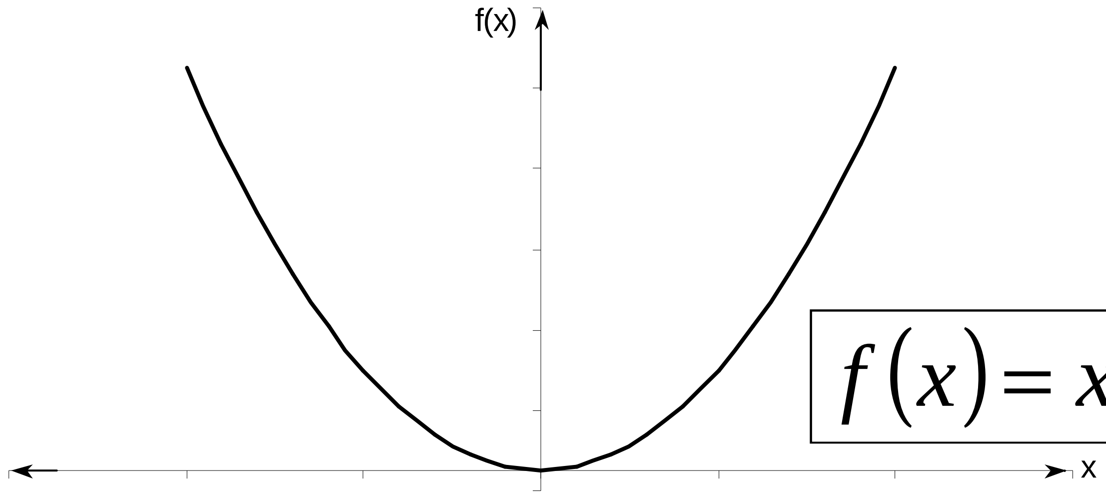
- Slow convergence

Drawbacks (continued)

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

Drawbacks (continued)

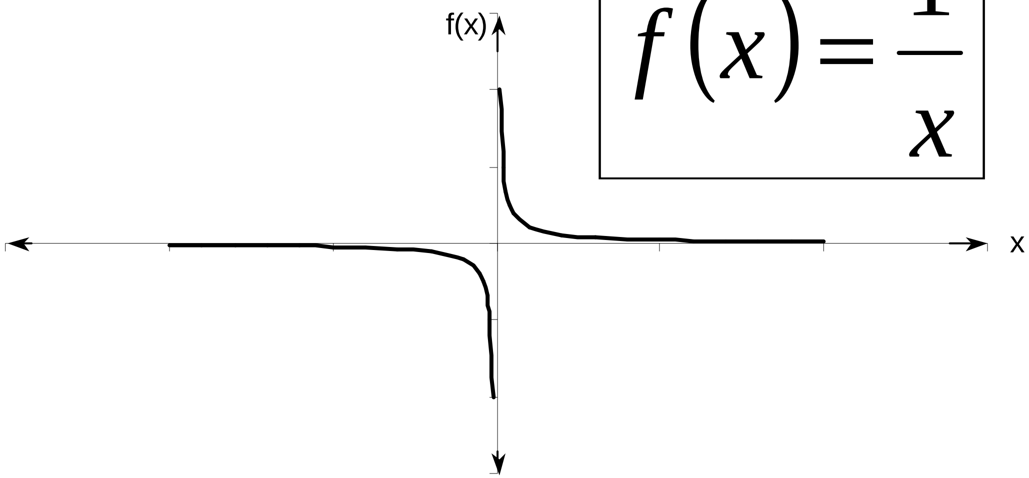
- If a function f(x) is such that it just touches the x-axis it will be unable to find the lower and upper guesses.



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

Drawbacks (continued)

- Function changes sign but root does not exist



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

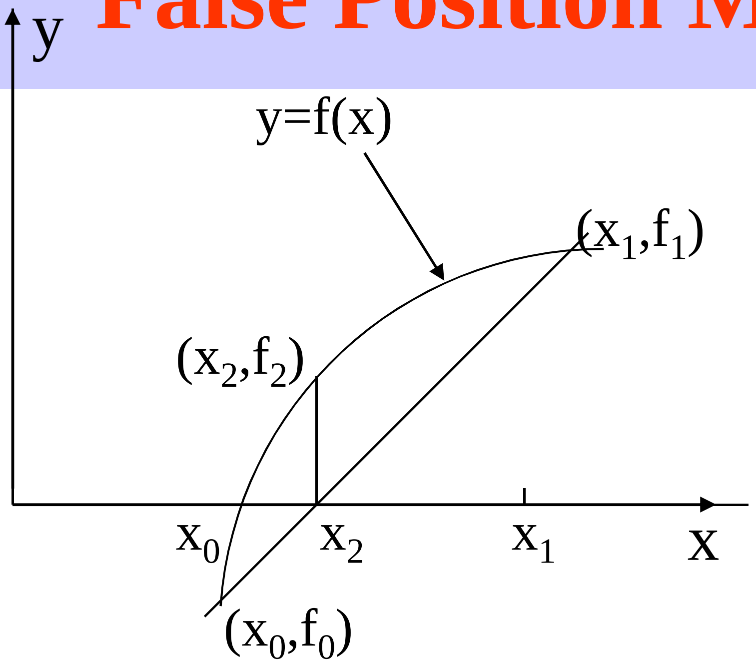
Improvement to Bisection

- *Regula Falsi*, or *Method of False Position*.
- Use the shape of the curve as a cue
- Use a straight line between *y* values to select interior point
- As curve segments become small, this closely approximates the root

False Position Method (Regula Falsi)

Instead of bisecting the interval $[x_0,x_1]$, we choose the point where the straight line through the end points meet the x-axis as x_2 and bracket the root with $[x_0,x_2]$ or $[x_2,x_1]$ depending on the sign of $f(x_2)$.

False Position Method



Straight line through (x_0,f_0) ,(x_1,f_1) :

$$y = f_1 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_1)$$

New end point x_2 :

$$x_2 = x_1 - \left(\frac{x_1 - x_0}{f_1 - f_0}\right) f_1$$

False Position Method (Pseudo Code)

1. Choose $\in > 0$ (tolerance on $|f(x)|$)
   $m > 0$ (maximum number of iterations)
   $k = 1$ (iteration count)
   x_0, x_1 (so that $f_0, f_1 < 0$)
2. {
   a. Compute $$x_2 = x_1 - \left(\frac{x_1 - x_0}{f_1 - f_0}\right) f_1$$
   $f_2 = f(x_2)$

   b. If $f_0 f_2 < 0$ set $x_1 = x_2$, $f_0 = f_2$

   c. $k = k+1$

   }

3. While ($|f_2| \geq \in$) and ($k \leq m$)
4. $x = x_2$, the root.

: bisection method

Solve the equation

$$\sin x = 0$$

using the initial interval $a = 2$ and $b = 4$.

n	a_n	b_n	c_n	$\lvert f(c_n)\rvert$
0	2.000000	4.000000	3.000000	1.411200e-01
1	3.000000	4.000000	3.500000	3.507832e-01
2	3.000000	3.500000	3.250000	1.081951e-01
3	3.000000	3.250000	3.125000	1.659189e-02
4	3.125000	3.250000	3.187500	4.589122e-02
5	3.125000	3.187500	3.156250	1.465682e-02
6	3.125000	3.156250	3.140625	9.676534e-04
7	3.140625	3.156250	3.148438	6.844793e-03
8	3.140625	3.148438	3.144531	2.938592e-03
9	3.140625	3.144531	3.142578	9.854713e-04
10	3.140625	3.142578	3.141602	8.908910e-06
11	3.140625	3.141602	3.141113	4.793723e-04
12	3.141113	3.141602	3.141357	2.352317e-04
13	3.141357	3.141602	3.141479	1.131614e-04
14	3.141479	3.141602	3.141541	5.212625e-05
15	3.141541	3.141602	3.141571	2.160867e-05
16	3.141571	3.141602	3.141586	6.349879e-06
17	3.141586	3.141602	3.141594	1.279516e-06
18	3.141586	3.141594	3.141590	2.535182e-06
19	3.141590	3.141594	3.141592	6.278330e-07

Experimentally, 18 iterations are required to compute π with 6 significant digits.

False Position Method

Solve the equation

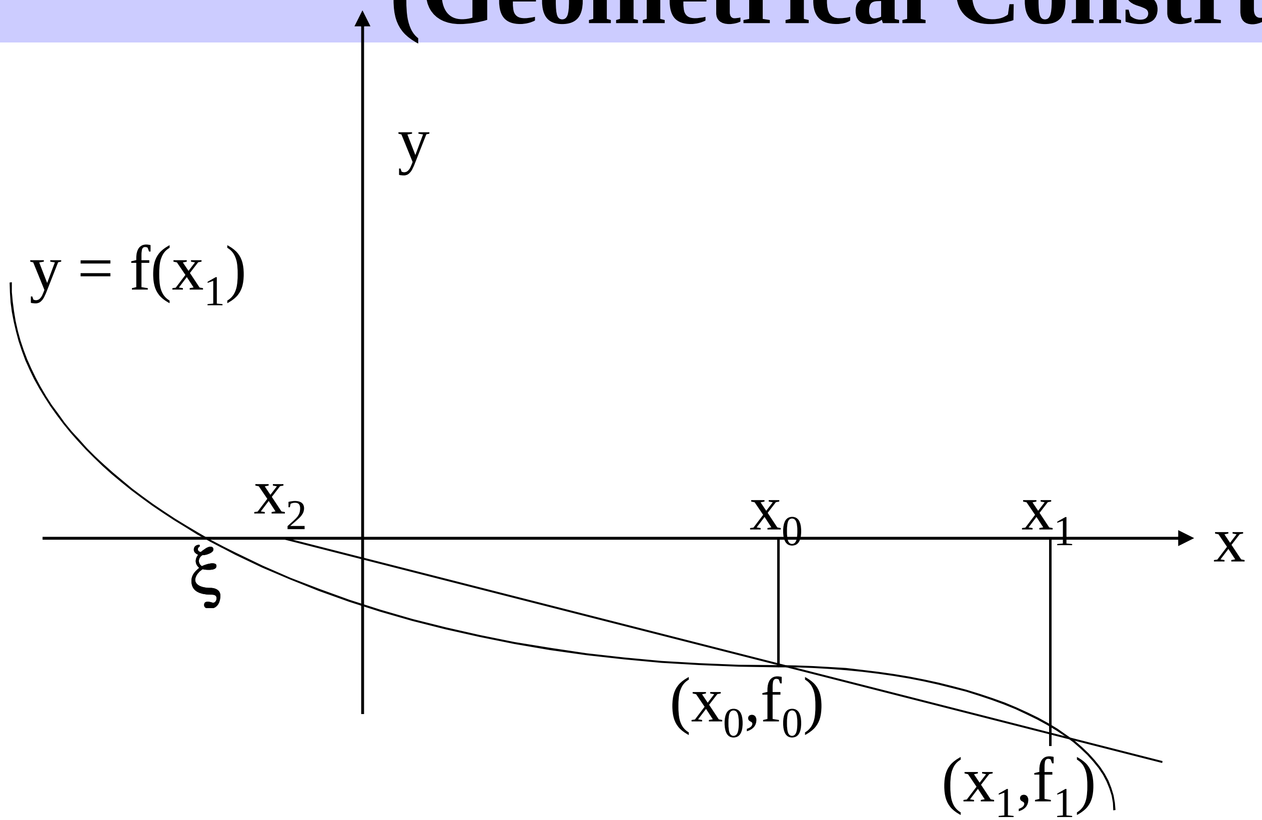
$$\sin x = 0$$

using the initial interval $a = 2$ and $b = 4$.

n	a_n	b_n	c_n	$\lvert f(c_n)\rvert$
0	2.000000	4.000000	3.091528	5.004366e-02
1	3.091528	4.000000	3.147875	6.282262e-03
2	3.091528	3.147875	3.141590	2.295634e-06
3	3.141590	3.147875	3.141593	1.509491e-11
4	3.141590	3.141593	3.141593	1.224647e-16
5	3.141593	3.141593	3.141593	1.224647e-16

Experimentally, 3 iterations are required to compute π with 6 significant digits.

The Secant Method (Geometrical Construction)



- Two initial points x_0, x_1 are chosen
- The next approximation x_2 is the point where the straight line joining (x_0,f_0) and (x_1,f_1) meet the x-axis
- Take (x_1,x_2) and repeat.

The secant Method (Pseudo Code)

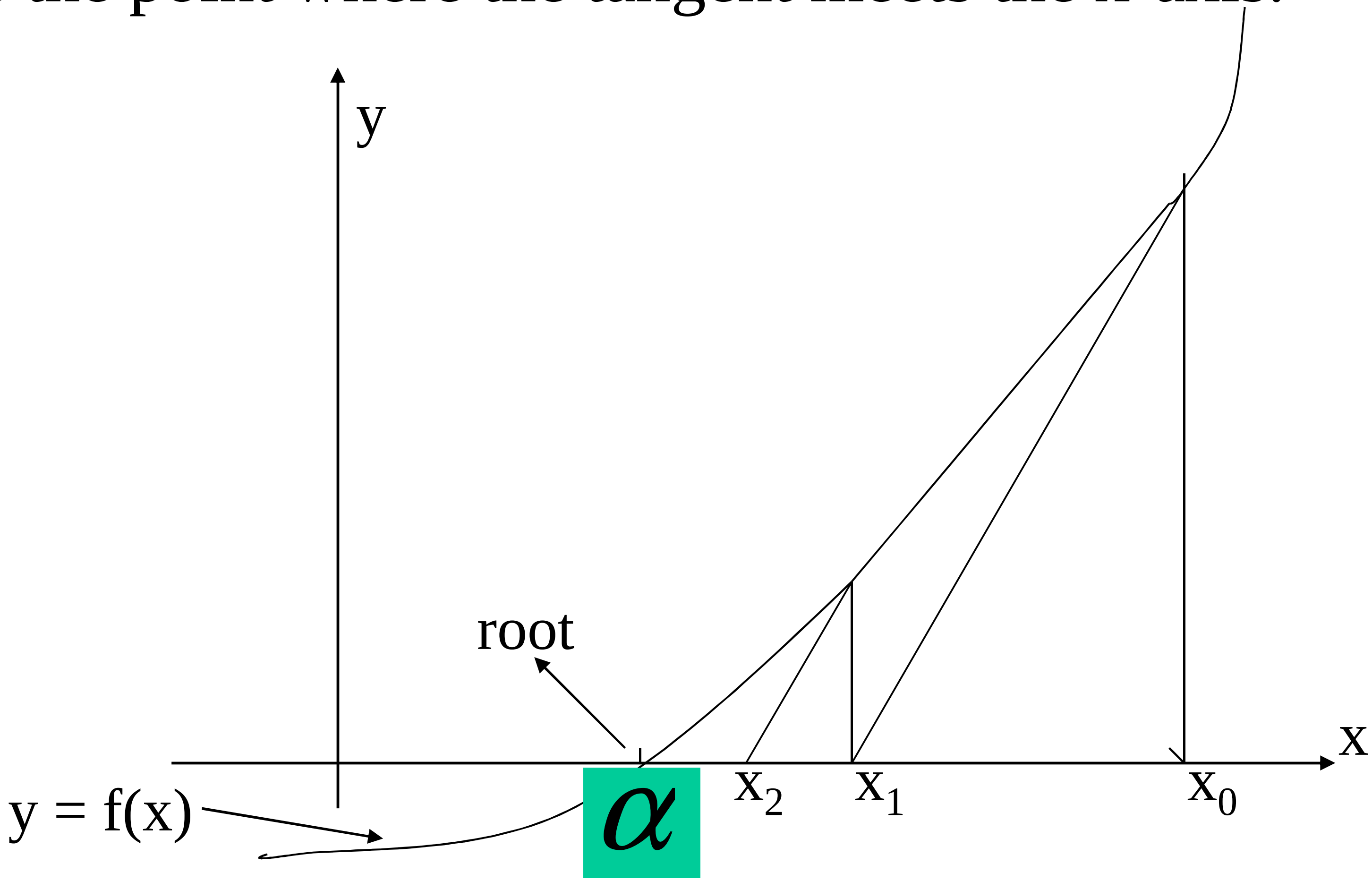
1. Choose $\in > 0$ (function tolerance $|f(x)| \leq \in$)
   $m > 0$ (Maximum number of iterations)
   x_0 , x_1 (Two initial points near the root)
   $f_0 = f(x_0)$
   $f_1 = f(x_1)$
   $k = 1$ (iteration count)
2. Do {
   $$x_2 = x_1 - \left(\frac{x_1 - x_0}{f_1 - f_0} \right) f_1$$
   $x_0 = x_1$
   $f_0 = f_1$
   $x_1 = x_2$
   $f_1 = f(x_2)$
   $k = k+1$ }
3. While ($|f_1| \geq \in$) and ($m \leq k$)

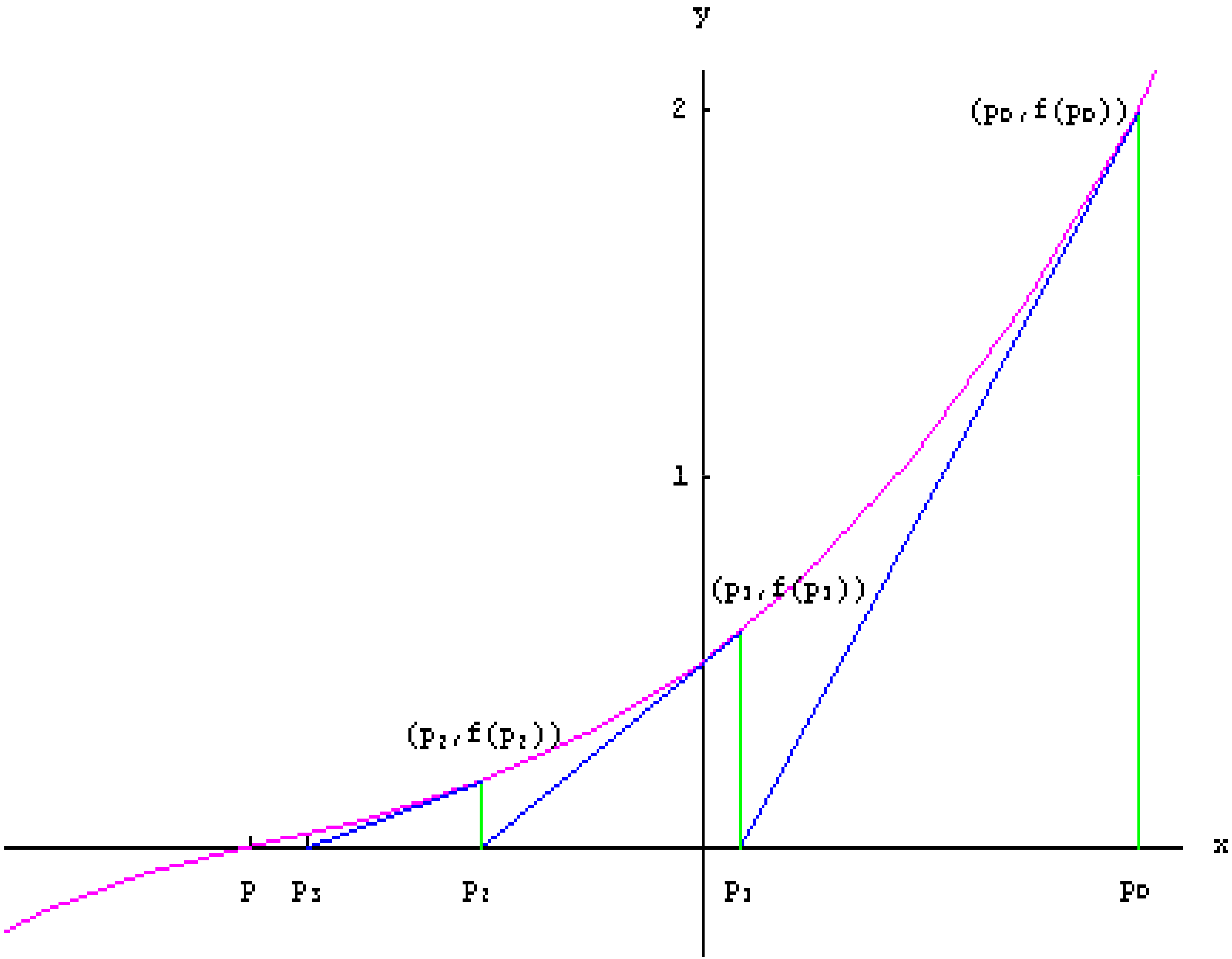
- Example
- As an example of the secant method, suppose we wish to find a root of the function
- $f(x) = \cos(x) + 2\sin(x) + x^2$.
- A closed form solution for x does not exist so we must use a numerical technique. We will use $x0 = 0$ and $x1 = -0.1$ as our initial approximations. We will let the two values $\varepsilon step = 0.001$ and $\varepsilon abs = 0.001$ and we will halt after a maximum of $N = 100$ iterations.
- We will use four decimal digit arithmetic to find a solution and the resulting iteration is shown in Table 1.

n	x_{n-1}	x_n	x_{n+1}	$\lvert f(x_{n+1})\rvert$	$\lvert x_{n+1} - x_n\rvert$
1	0.0	-0.1	-0.5136	0.1522	0.4136
2	-0.1	-0.5136	-0.6100	0.0457	0.0964
3	-0.5136	-0.6100	-0.6514	0.0065	0.0414
4	-0.6100	-0.6514	-0.6582	0.0013	0.0068
5	-0.6514	-0.6582	-0.6598	0.0006	0.0016
6	-0.6582	-0.6598	-0.6595	0.0002	0.0003

Newton-Raphson Method / Newton's Method

At an approximate x_k to the root ,the curve is approximated by the tangent to the curve at x_k and the next approximation x_{k+1} is the point where the tangent meets the x-axis.




y
2
1
(p0, f(p0))
(p1, f(p1))
(p2, f(p2))
x
p
p3
p2
p1
p0

Tangent at (x_k, f_k) :

$$y = f(x_k) + f'(x_k)(x-x_k)$$

This tangent cuts the x-axis at x_{k+1}

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

<u>Warning</u> : If $f'(x_k)$ is very small , method fails.

- Two function Evaluations per iteration

Newton's Method - Pseudo code

1. Choose $\in > 0$ (function tolerance $|f(x)| < \in$)

   $m > 0$ (Maximum number of iterations)

   x_0 - initial approximation

   k - iteration count

   Compute $f(x_0)$

2. Do { $q = f'(x_0)$ (evaluate derivative at x_0)

   $x_1 = x_0 - f_0/q$

   $x_0 = x_1$

   $f_0 = f(x_0)$

   $k = k+1$

   }

3. While $(|f_0| \geq \in)$ and $(k \leq m)$

4. $x = x_1$ the root.

Newton's Method for finding the square root of a number $x = \sqrt{a}$

$f(x) = x^2 - a^2 = 0$

$$x_{k+1} = x_k - \frac{x_k^2 - a^2}{2x_k}$$

Example : a = 5 , initial approximation $x_0 = 2$.

$x_1 = 2.25$

$x_2 = 2.236111111$

$x_3 = 2.236067978$

$x_4 = 2.236067978$

As an example of Newton's method, suppose we wish to find a root of the function $f(x) = \cos(x) + 2\sin(x) + x^2$.

A closed form solution for x does not exist so we must use a numerical technique. We will use $x0 = 0$ as our initial approximation. We will let the two values $\varepsilon step = 0.001$ and $\varepsilon abs = 0.001$ and we will halt after a maximum of $N = 100$ iterations.

From calculus, we know that the derivative of the given function is

$f'(x) = -\sin(x) + 2\cos(x) + 2x$.

We will use four decimal digit arithmetic to find a solution and the resulting iteration is shown in

Table 2.

Table 2. Newton's method applied to $f(x) = \cos(x) + 2\sin(x) + x^2$.

n	x_n	x_{n+1}	$\lvert f(x_{n+1})\rvert$	$\lvert x_{n+1} - x_n\rvert$
0	0.0	-0.5000	0.1688	0.5000
1	-0.5000	-0.6368	0.0205	0.1368
2	-0.6368	-0.6589	0.0008000	0.02210
3	-0.6589	-0.6598	0.0006	0.0009

Thus, with the last step, both halting conditions are met, and therefore, after four iterations, our approximation to the root is -0.6598 .

General remarks on Convergence

\# The false position method in general converges faster than the bisection method. (But not always) .

\# The bisection method and the false position method are guaranteed for convergence.

\# The secant method and the Newton-Raphson method are not guaranteed for convergence.

Comparison of Methods

Method	Initial guesses	Convergence rate	Stability	
Bisection	2	Slow	Always	
False position	2	Medium	Always	
Fixed-pointed iteration	1	Slow	Possibly divergent	
Newton-Raphson	1	Fast	Possibly divergent	Evaluate f'(x)
Modified Newton-Raphson	1	Fast: multiple roots Medium:single root	Possibly divergent	F"(x) and f'(x)
Secant	2	Medium to fast	Possibly divergent	Initial guesses don't have to bracket root
Modified secant	2	Fast	Possibly divergent	