CPE 310: Numerical Analysis for Engineers

Chapter 1: Solving Nonlinear Equations

Ahmed Tamrawi

An important problem in applied mathematics is to:
"**solve** $f(x) = 0$" where $f(x)$ is a function of x

The values of x that make $f(x) = 0$ are called the **roots of the equation** or **zeros of** $\boldsymbol{f}$

The problem is known as **root finding** or **zero finding**

"1-D Nonlinear Equation"

We are concerned about solving <u>*single nonlinear equation*</u> in **one unknown**, where

$$f\colon \mathbb{R} \to \mathbb{R}$$

Solution is **scalar** x for which $f(x) = 0$

Example: $f(x) = 3x + \sin x - e^x$ for which $x = 0.3604$ is one approximate solution

These lectures describe some of the many methods for solving $f(x) = 0$ by ***numerical procedures*** where $f(x)$ is *single nonlinear equation* in **one unknown**

Interval Halving (Bisection)

Secant Method

False-Position Method

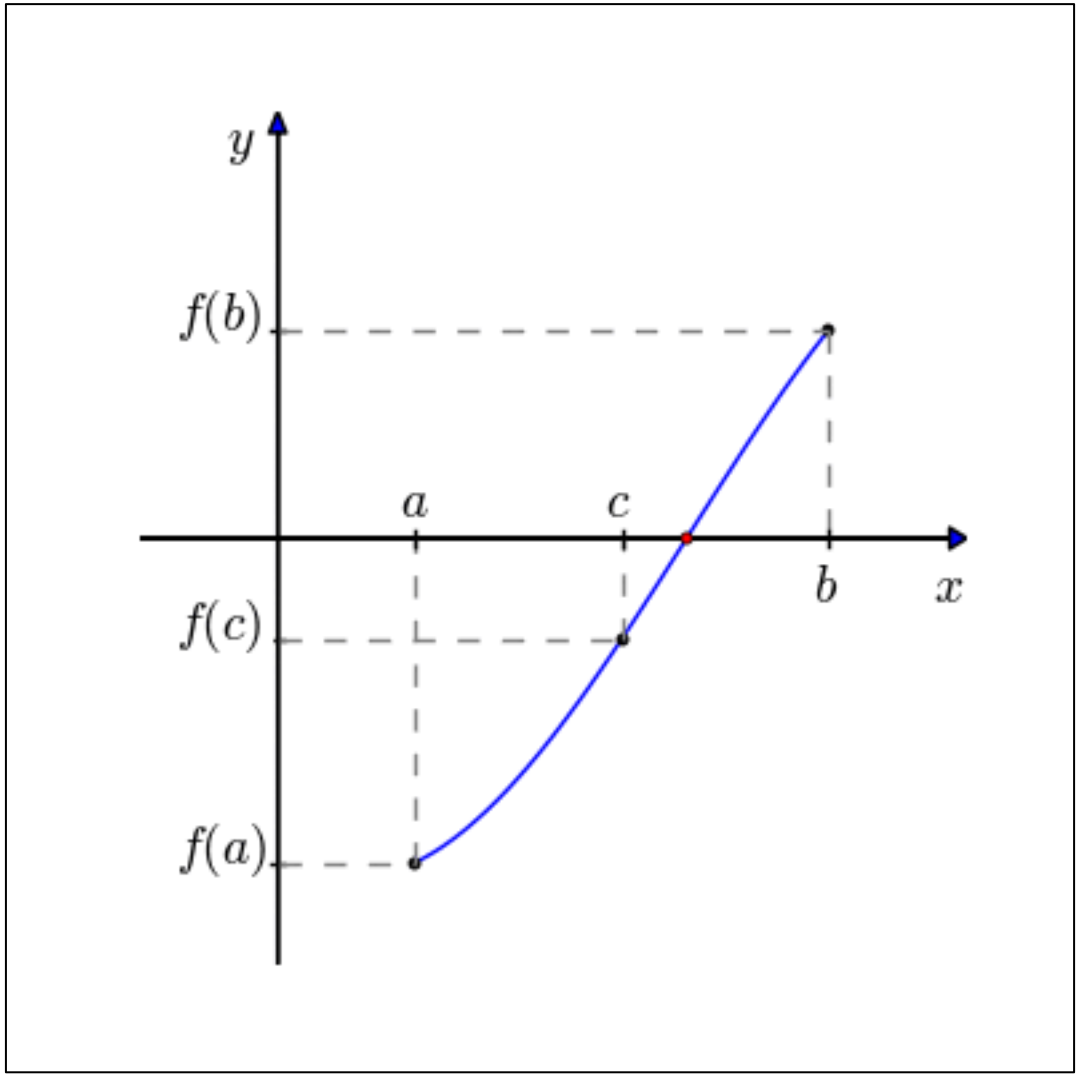
Linear Interpolation Methods

Newton's Method

Fixed-Point Iteration Method

Interval Halving (Bisection) Method

Interval Halving (Bisection) Method



Ancient but effective method for finding a zero of $f(x)$

It **begins with two values** for x that ***bracket*** a root

It determines that they do bracket a root because $f(x)$ ***changes signs*** at these two x-values

if $f(x)$ is **continuous**, there **must be** at least one root between the values

A plot of $f(x)$ is useful to know where to start

Interval Halving (Bisection) Algorithm

INPUT

- x_1 and x_2 such that $f(x_1)f(x_2) < 0$
- `tol`: the specified tolerance value

```
REPEAT
    SET x3 = (x1 + x2)/2
    IF f(x3)f(x1) < 0 THEN
        SET x2 = x3
    ELSE
        SET x1 = x3
UNTIL (|x1 − x2| < tol) OR f(x3) = 0
```

NOTES

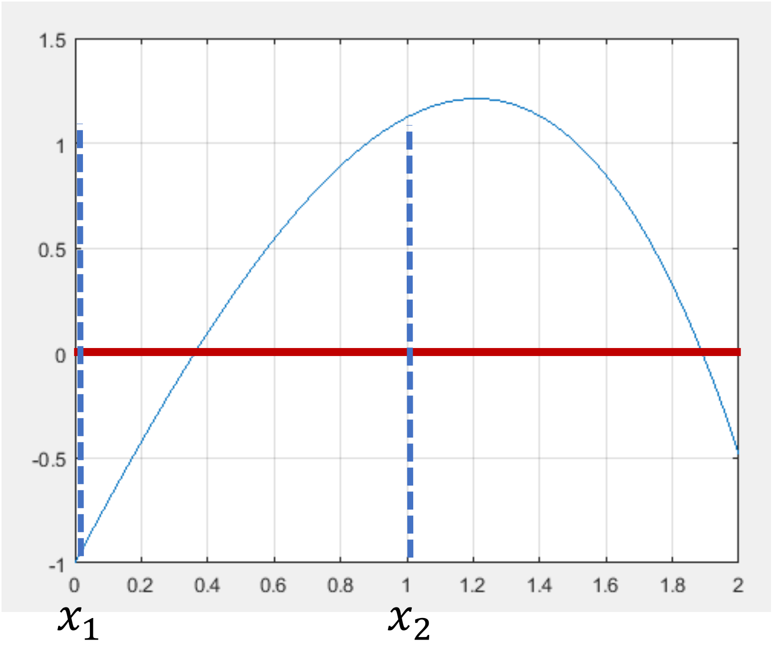
- The final value of x_3 approximates the root, and it is in error by not more than $\frac{|x_1 - x_2|}{2}$ ← *"Maximum Error"*
- The algorithm may produce a **false root** if $f(x)$ is discontinuous on $[x_1, x_2]$

$$f(x) = 3x + \sin x - e^x = 0$$

$x_{\text{Actual}} = 0.36042170296032440136932951583028$

$x_1 = 0, x_2 = 1$ such that $f(0)f(1) < 0$

`tol` $= 1E-3$ **OR** (0.001)



Interval Halving (Bisection) Algorithm

INPUT

- x_1 and x_2 such that $f(x_1)f(x_2) < 0$
- `tol`: the specified tolerance value

REPEAT

SET $x_3 = \frac{(x_1+x_2)}{2}$

IF $f(x_3)f(x_1) < 0$ **THEN**

SET $x_2 = x_3$

ELSE

SET $x_1 = x_3$

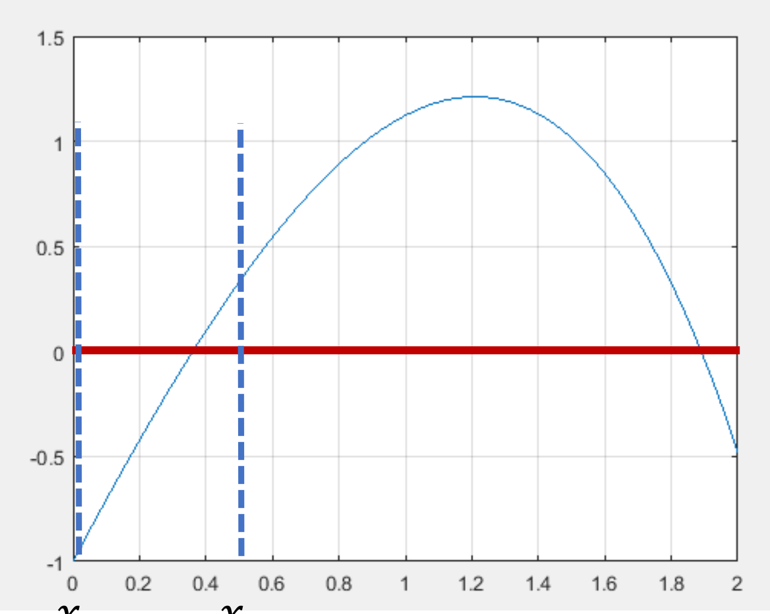
UNTIL ($|x_1 - x_2|$ < `tol`) **OR** $f(x_3) = 0$

Max Error: $\frac{|x_1 - x_2|}{2}$

Actual Error: $x_{\text{Actual}} - x_3$

Absolute Error: $|x_{\text{Actual}} - x_3|$

Iter	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$	Max Error	Actual Error	Absolute Error
1	0.00000	1.00000		-1.00000	1.12320				



x_1 x_2

$$f(x) = 3x + \sin x - e^x = 0$$

Actual $x =$ .360421702960324401369329515830 28

$x_1 = 0, x_2 = 1$ such that $f(0)f(1) < 0$

$\texttt{tol} = 1E - 3$ **OR** (0.001)

Iter	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$	Max Error	Actual Error	Absolute Error
1	0.00000	1.00000	0.50000	-1.00000	1.12320	0.33070	0.50000	-0.13958	0.13958
2	0.00000	0.50000		-1.00000	0.33070				

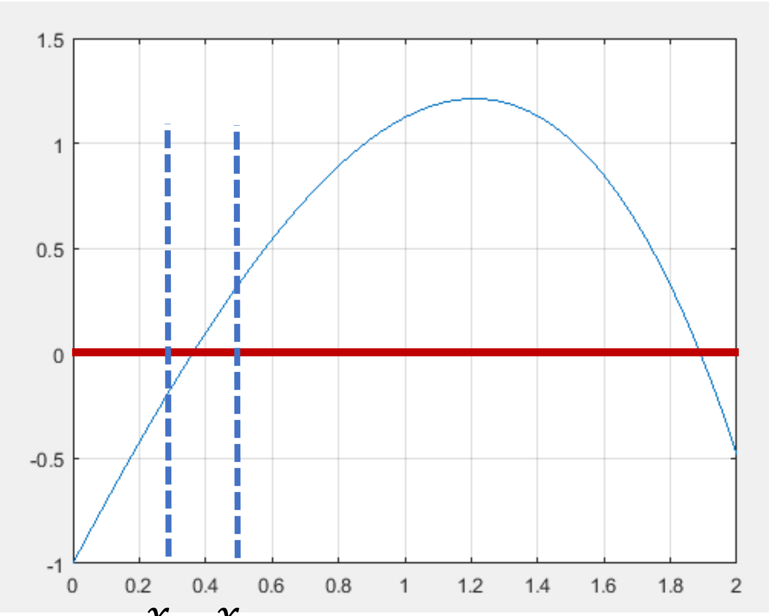
Interval Halving (Bisection) Algorithm

INPUT
- x_1 and x_2 such that $f(x_1)f(x_2) < 0$
- $\texttt{tol}$: the specified tolerance value

REPEAT
 SET $x_3 = \frac{(x_1+x_2)}{2}$
 IF $f(x_3)f(x_1) < 0$ **THEN**
  SET $x_2 = x_3$
 ELSE
  SET $x_1 = x_3$
UNTIL $(|x_1 - x_2| < \texttt{tol})$ **OR** $f(x_3) = 0$

$$f(x) = 3x + \sin x - e^x = 0$$

Actual $x =$ .36042170296032440136932951583028



$x_1 \quad x_2$

$x_1 = 0, x_2 = 1$ such that $f(0)f(1) < 0$

$\texttt{tol} = 1E - 3$ **OR** (0.001)

Interval Halving (Bisection) Algorithm

INPUT

- x_1 and x_2 such that $f(x_1)f(x_2) < 0$
- tol: the specified tolerance value

REPEAT

SET $x_3 = \frac{(x_1+x_2)}{2}$

IF $f(x_3)f(x_1) < 0$ **THEN**

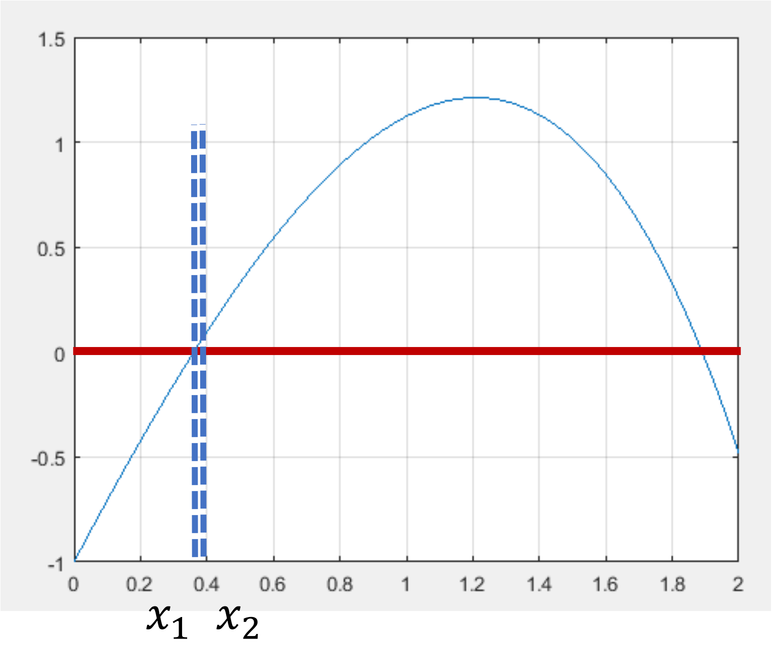
SET $x_2 = x_3$

ELSE

SET $x_1 = x_3$

UNTIL $(|x_1 - x_2| < \texttt{tol})$ **OR** $f(x_3) = 0$

Iter	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$	Max Error	Actual Error	Absolute Error
1	0.00000	1.00000	0.50000	-1.00000	1.12320	0.33070	0.50000	-0.13958	0.13958
2	0.00000	0.50000	0.25000	-1.00000	0.33070	-0.28662	0.25000	0.11042	0.11042
3	0.25000	0.50000		-0.28662	0.33070				



$$f(x) = 3x + \sin x - e^x = 0$$

Actual $x =$ .36042170296032440136932951583028

$x_1 = 0, x_2 = 1$ such that $f(0)f(1) < 0$

`tol` $= 1E - 3$ **OR** (0.001)

Interval Halving (Bisection) Algorithm

INPUT

- x_1 and x_2 such that $f(x_1)f(x_2) < 0$
- `tol`: the specified tolerance value

REPEAT

SET $x_3 = \frac{(x_1+x_2)}{2}$

IF $f(x_3)f(x_1) < 0$ **THEN**

SET $x_2 = x_3$

ELSE

SET $x_1 = x_3$

UNTIL ($|x_1 - x_2|$ < `tol`) **OR** $f(x_3) = 0$

Iter	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$	Max Error	Actual Error	Absolute Error
1	0.00000	1.00000	0.50000	-1.00000	1.12320	0.33070	0.50000	-0.13958	0.13958
2	0.00000	**0.50000**	0.25000	-1.00000	0.33070	-0.28662	0.25000	0.11042	0.11042
3	**0.25000**	0.50000	0.37500	-0.28662	0.33070	0.03628	0.12500	-0.01458	0.01458
4	0.25000	**0.37500**	0.31250	-0.28662	0.03628	-0.12190	0.06250	0.04792	0.04792
5	**0.31250**	0.37500	0.34375	-0.12190	0.03628	-0.04196	0.03125	0.01667	0.01667
6	**0.34375**	0.37500	0.35938	-0.04196	0.03628	-0.00262	0.01563	0.00105	0.00105
7	**0.35938**	0.37500	0.36719	-0.00262	0.03628	0.01689	0.00781	-0.00677	0.00677
8	0.35938	**0.36719**	0.36328	-0.00262	0.01689	0.00715	0.00391	-0.00286	0.00286
9	0.35938	**0.36328**	0.36133	-0.00262	0.00715	0.00227	0.00195	-0.00092	0.00092
10	0.35938	**0.36133**	0.36035	-0.00262	0.00227	-0.00018	0.00098	0.00007	0.00007
11	**0.36035**	0.36133	0.36084	-0.00018	0.00227	0.00105	0.00049	-0.00042	0.00042

$$|0.36035 - 0.36133| < 0.001 \implies 0.00098 < 0.001$$

Tolerance Met

Simple and **guaranteed** to work if:

1. $f(x)$ is **continuous** in $[x_1, x_2]$; and
2. The values x_1 and x_2 **actually bracket a root**

Needed iterations to achieve a **specified accuracy** is known in advance

Error after n iterations $< \left|\frac{x_2 - x_1}{2^n}\right|$

Plotting the function helps defining the bracket

Good for initial guess for other root finding algorithms

Interval Halving (Bisection) Algorithm

INPUT

- x_1 and x_2 such that $f(x_1)f(x_2) < 0$
- `tol`: the specified tolerance value

```
REPEAT
    SET  x3 = (x1 + x2)/2
    IF  f(x3)f(x1) < 0  THEN
        SET  x2 = x3
    ELSE
        SET  x1 = x3
UNTIL  (|x1 − x2| < tol)  OR  f(x3) = 0
```

REPEAT

 SET $x_3 = \frac{(x_1 + x_2)}{2}$

 IF $f(x_3)f(x_1) < 0$ **THEN**

  SET $x_2 = x_3$

 ELSE

  SET $x_1 = x_3$

UNTIL $(|x_1 - x_2| <$ `tol`$)$ **OR** $f(x_3) = 0$

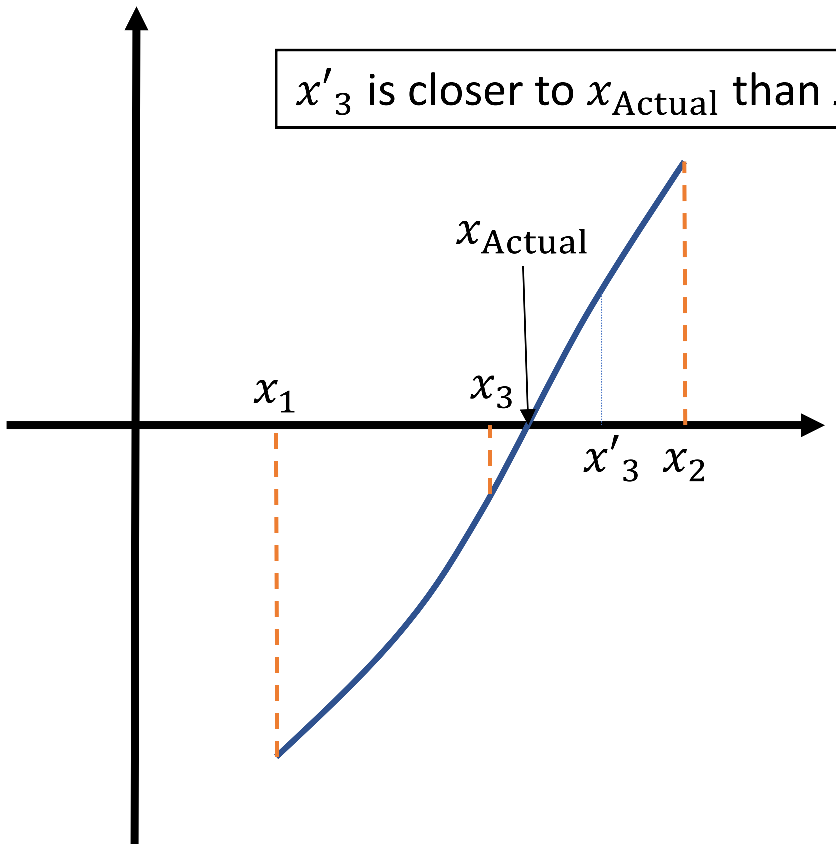
Slow Convergence compared to other techniques we will see next

Other methods require less number of iterations

Some earlier iteration may result on a closer approximation than later ones

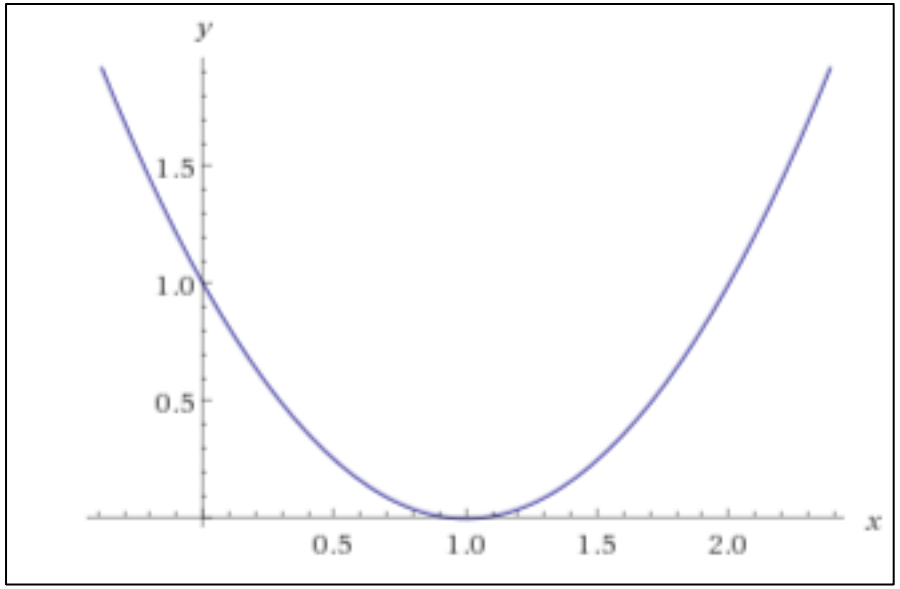
Iter	x_1	x_2	x_3	$f(x_1)$	$f(x_2)$	$f(x_3)$	Max Error	Actual Error	Absolute Error
1	0.00000	1.00000	0.50000	-1.00000	1.12320	0.33070	0.50000	-0.13958	0.13958
2	0.00000	**0.50000**	0.25000	-1.00000	0.33070	-0.28662	0.25000	0.11042	0.11042
3	**0.25000**	0.50000	0.37500	-0.28662	0.33070	0.03628	0.12500	-0.01458	0.01458
4	0.25000	**0.37500**	0.31250	-0.28662	0.03628	-0.12190	0.06250	0.04792	0.04792
5	**0.31250**	0.37500	0.34375	-0.12190	0.03628	-0.04196	0.03125	0.01667	0.01667
6	**0.34375**	0.37500	0.35938	-0.04196	0.03628	-0.00262	0.01563	0.00105	0.00105
7	**0.35938**	0.37500	0.36719	-0.00262	0.03628	0.01689	0.00781	-0.00677	0.00677
8	0.35938	**0.36719**	0.36328	-0.00262	0.01689	0.00715	0.00391	-0.00286	0.00286
9	0.35938	**0.36328**	0.36133	-0.00262	0.00715	0.00227	0.00195	-0.00092	0.00092
10	0.35938	**0.36133**	0.36035	-0.00262	0.00227	-0.00018	0.00098	0.00007	0.00007
11	**0.36035**	0.36133	0.36084	-0.00018	0.00227	0.00105	0.00049	-0.00042	0.00042

← ***Closer*** (row 5)
← ***Further*** (row 6)
← ***Closer*** (row 10)
← ***Further*** (row 11)



When there are multiple roots, **interval halving may not be applicable**, because the function *<u>may not change sign</u>* at points on either side of the roots.

We may be able to find the roots by working with $f'(x)$, which will be zero at a multiple root.



Example:

$f(x) = x^2 - 2x + 1 = 0$, in bracket $[0, 2]$

Bisection Method - Animation

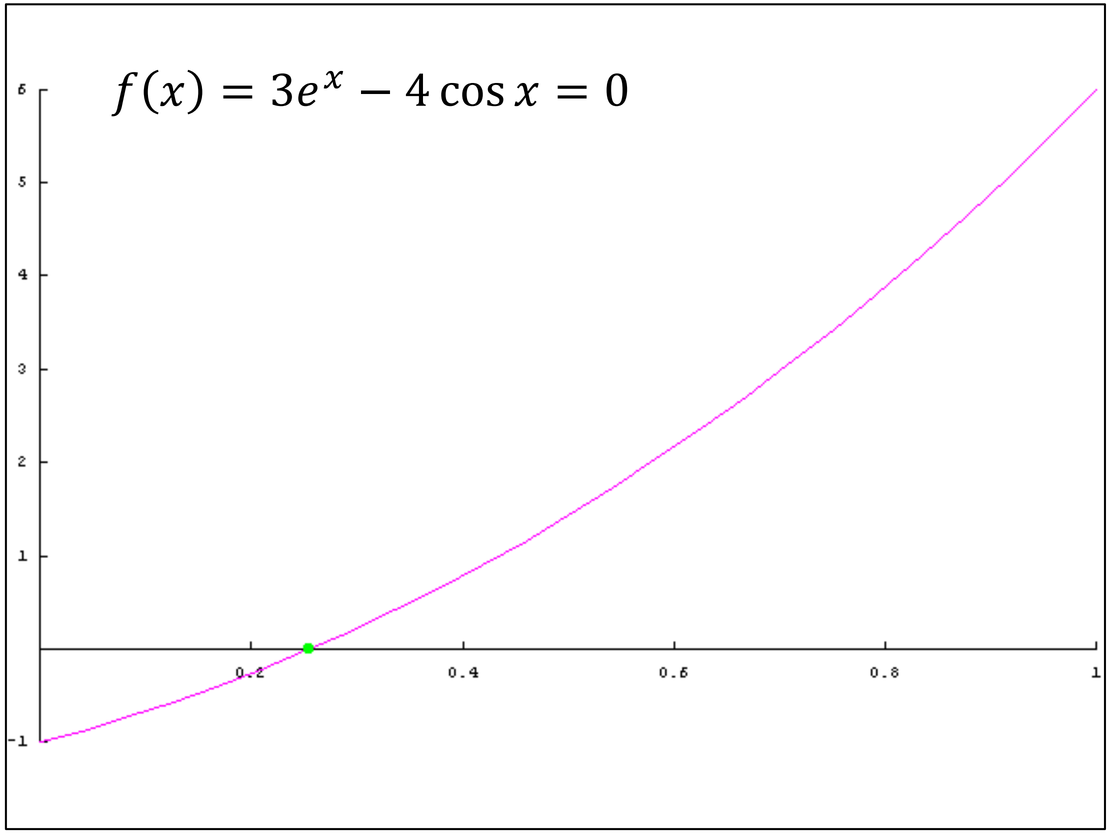
http://mathfaculty.fullerton.edu/mathews/a2001/Animations/RootFinding/BisectionMethod/BisectionMethod.html

Linear Interpolation Methods

Most functions can be approximated by a straight line over a small interval

Secant Method

Secant Method



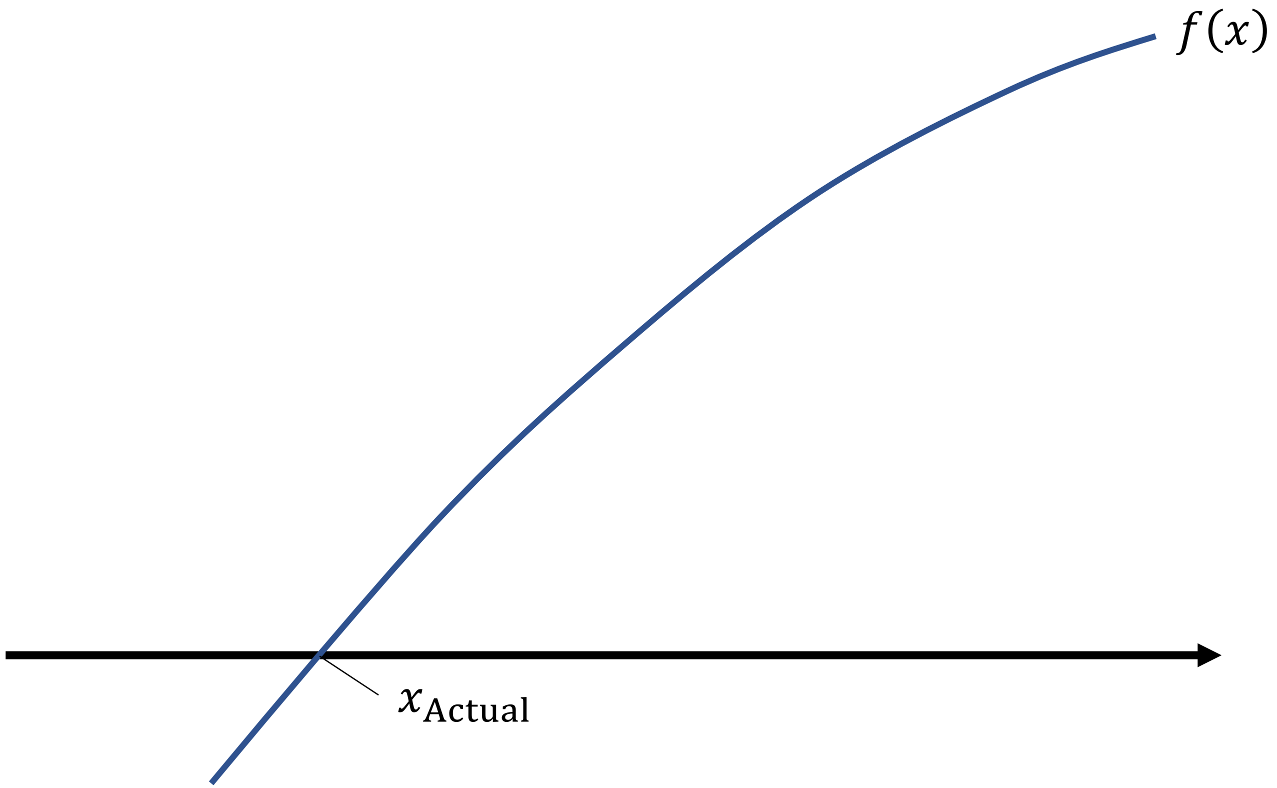
In geometry, a **secant of a curve** is a line that *intersects two points* on the curve

It **begins with two values** for x that are ***near*** *the* root

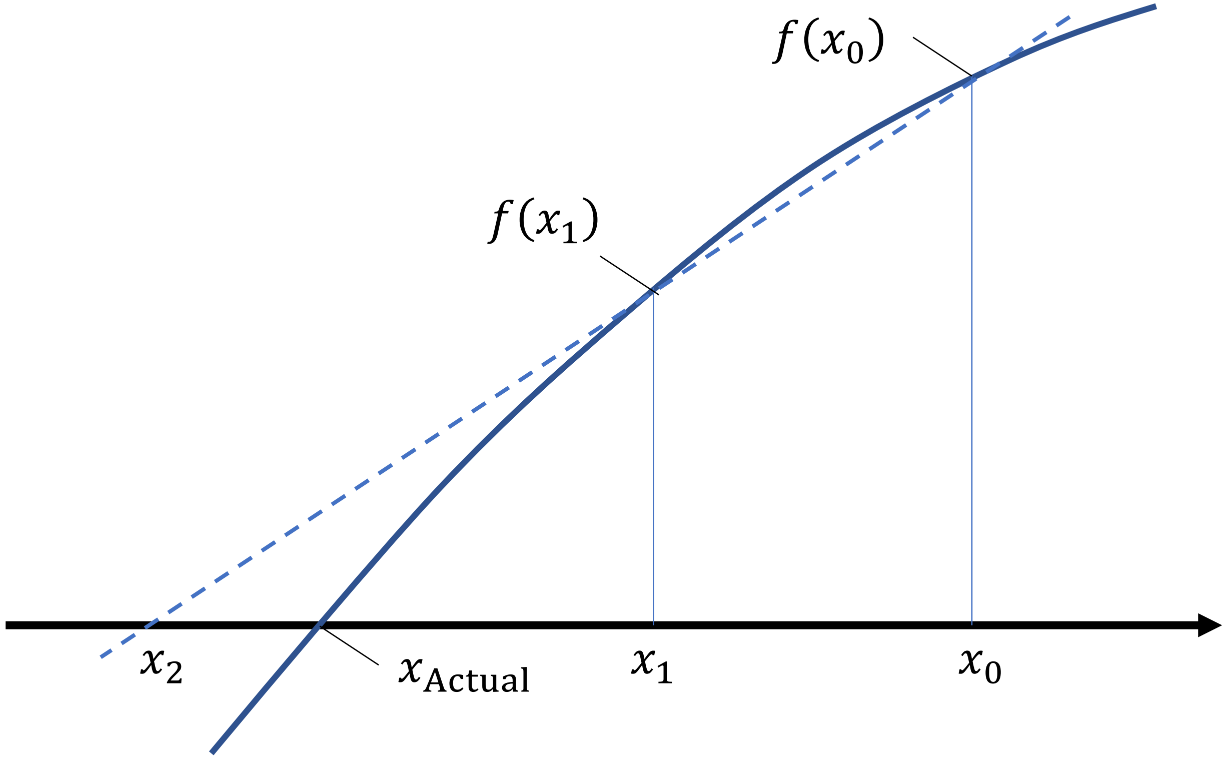
The two values maybe on the **same side** from the root or on **opposite side**

if $f(x)$ is **linear**, the secant intersects the x-axis at the root exactly

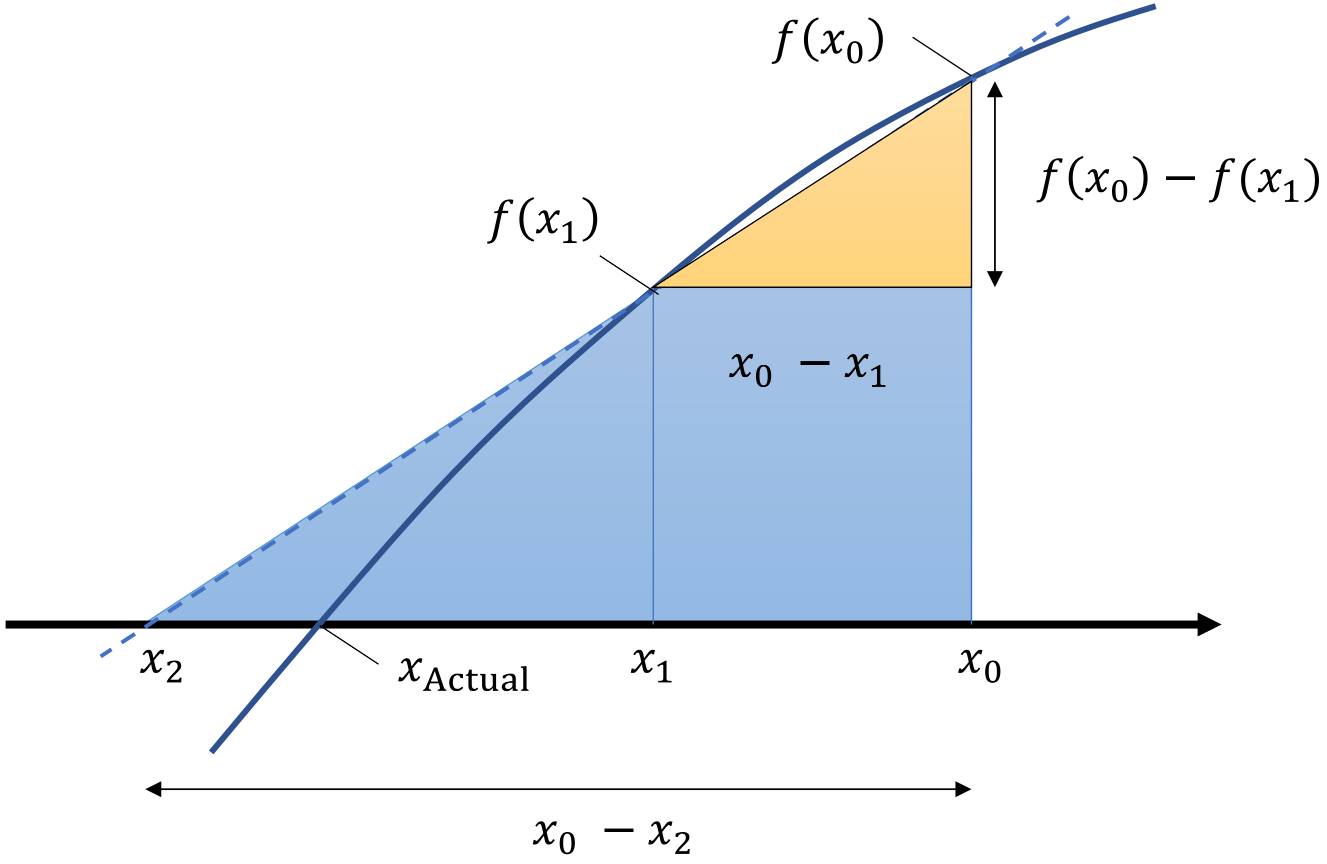
A plot of $f(x)$ is useful to know where to start



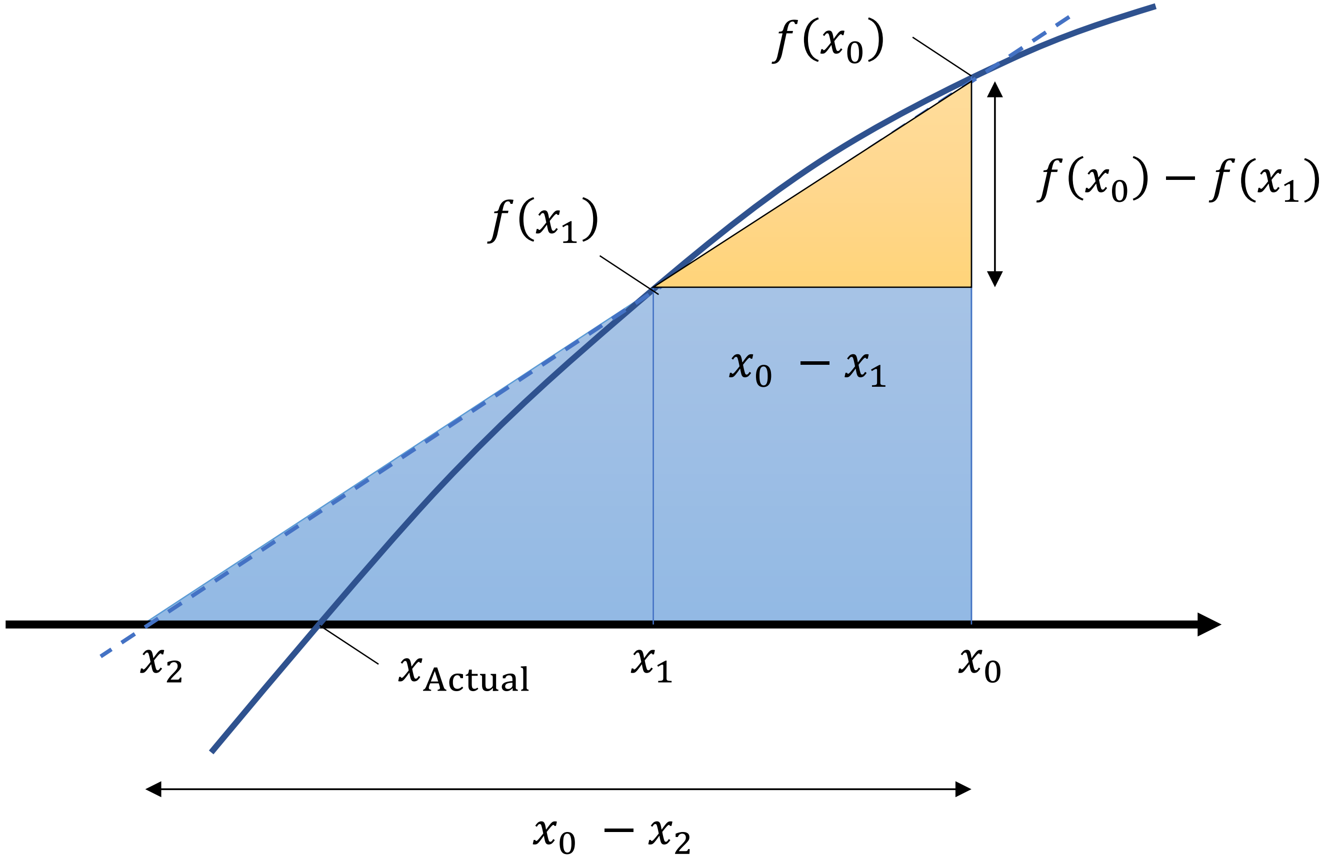
$f(x)$
x_{Actual}



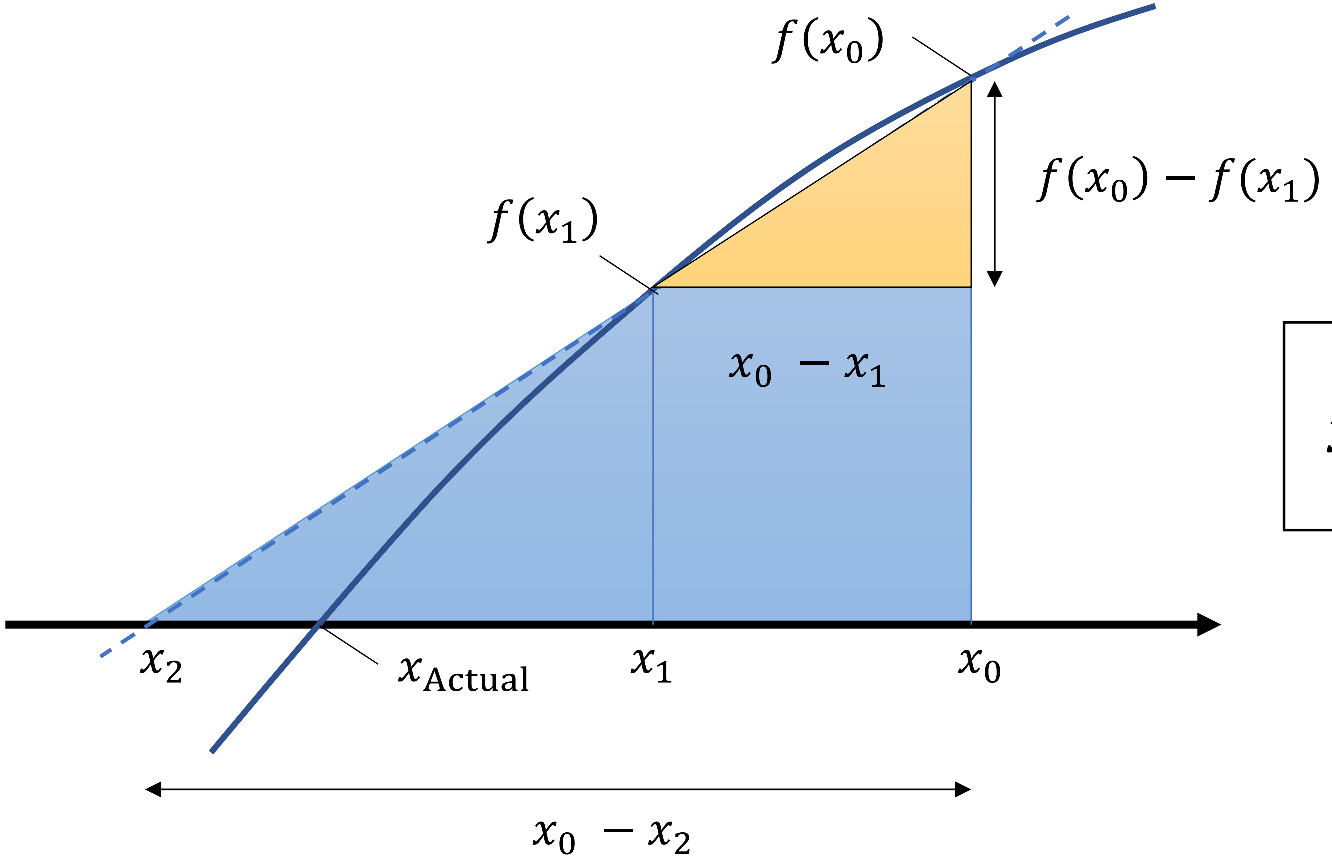
$f(x_0)$
$f(x_1)$
x_2
x_{Actual}
x_1
x_0



$f(x_0)$
$f(x_0) - f(x_1)$
$f(x_1)$
$x_0 - x_1$
x_2
x_{Actual}
x_1
x_0
$x_0 - x_2$



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



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

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

Secant Method

INPUT

- x_0 and x_1 that are **near** the root
- `tol`: the specified tolerance value

IF $|f(x_0)| < |f(x_1)|$ **THEN**

  SWAP x_0 with x_1

REPEAT

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

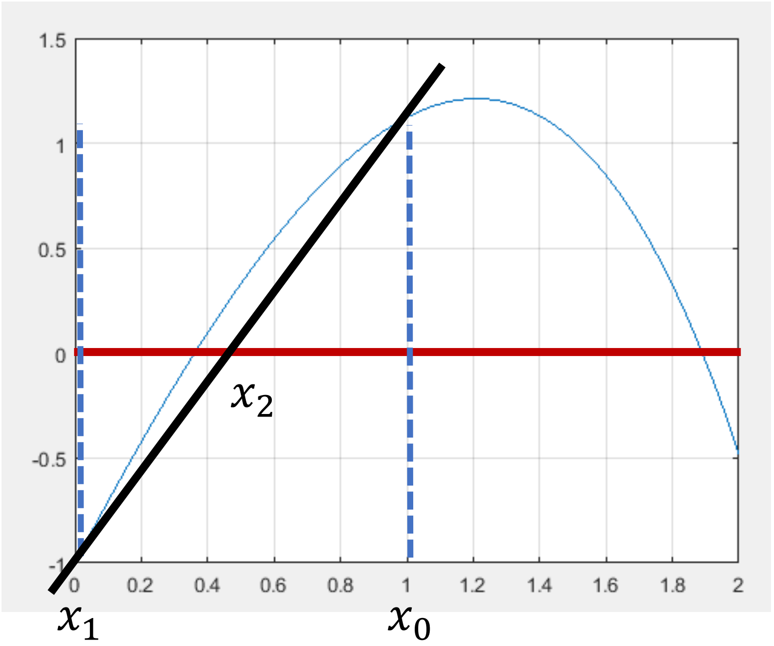
  SET $x_0 = x_1$

  SET $x_1 = x_2$

UNTIL $(|f(x_2)| <$ `tol`$)$

NOTES

- Another stopping "termination" criteria is when the pair of points being used are sufficiently close together: "**UNTIL** $(|x_0 - x_1| <$ `tol`$)$"
- The algorithm may **fail** if $f(x)$ is not continuous.



$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$

$x_0 = 1 \; x_1 = 0$

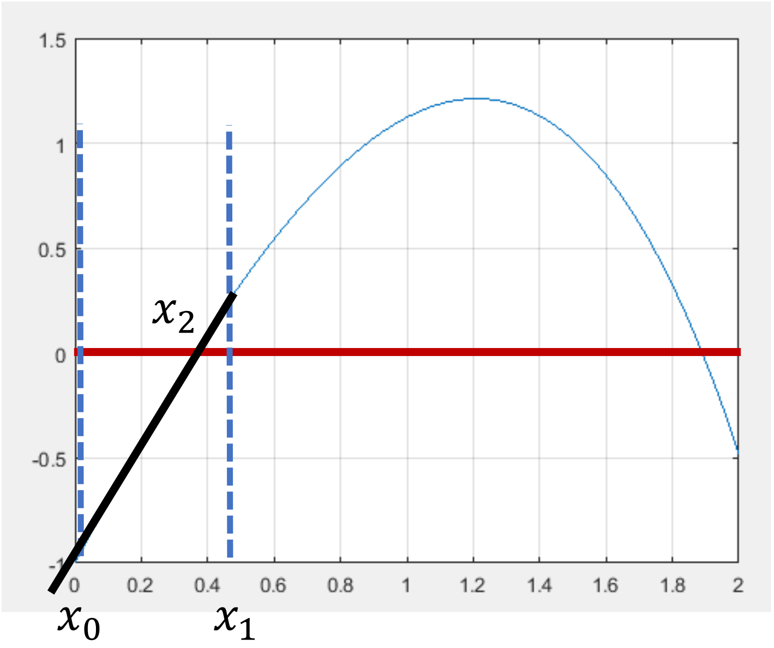
$\texttt{tol} = 1E - 7$ **OR** (0.0000001)

Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	1.000000	0.000000		1.12320	-1.00000		

Secant Method

INPUT
- x_0 and x_1 that are **near** the root
- tol: the specified tolerance value

IF $|f(x_0)| < |f(x_1)|$ **THEN**
  SWAP x_0 with x_1
REPEAT
  SET $x_2 = x_0 - f(x_0)\,\frac{x_0 - x_1}{f(x_0) - f(x_1)}$
  SET $x_0 = x_1$
  SET $x_1 = x_2$
UNTIL $(|f(x_2)| < \texttt{tol})$



$$f(x) = 3x + \sin x - e^x = 0$$

$x_{Actual} = 0.36042170296032440136932951583028$

$x_0 = 1$ $x_1 = 0$

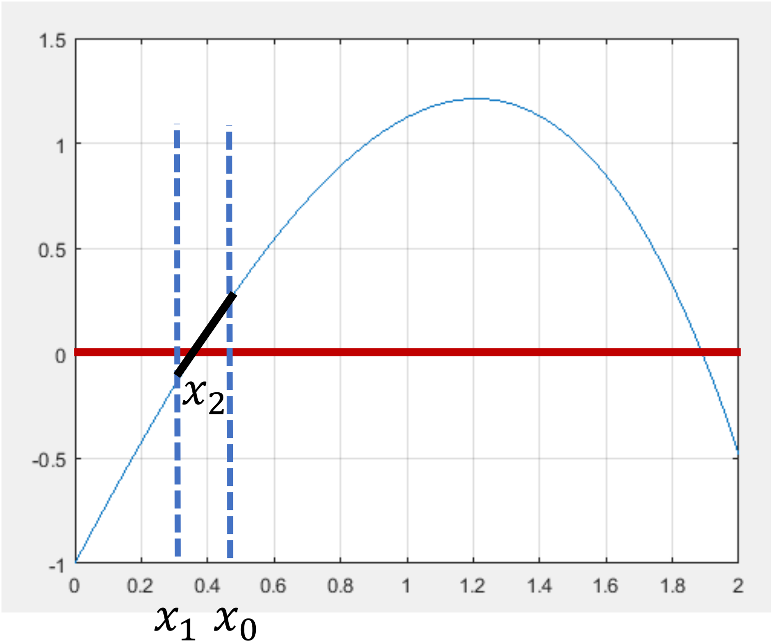
tol $= 1E-7$ **OR** (0.0000001)

Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	1.00000	0.00000	0.4709896	1.12320	-1.000000	0.2651588	-0.110567
2	0.00000	**0.4709896**		-1.000000	0.2651588		

Secant Method

INPUT
- x_0 and x_1 that are **near** the root
- tol: the specified tolerance value

IF $|f(x_0)| < |f(x_1)|$ **THEN**
    SWAP x_0 with x_1
REPEAT
    SET $x_2 = x_0 - f(x_0)\frac{x_0 - x_1}{f(x_0) - f(x_1)}$
    SET $x_0 = x_1$
    SET $x_1 = x_2$
UNTIL ($|f(x_2)| <$ tol)



$$f(x) = 3x + \sin x - e^x = 0$$

$x_{\text{Actual}} = 0.36042170296032440136932951583028$

$x_0 = 1$ $x_1 = 0$

tol $= 1E - 7$ **OR** (0.0000001)

Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	1.00000	0.00000	0.4709896	1.12320	-1.000000	0.2651588	-0.110567
2	0.00000	**0.4709896**	0.3722771	-1.000000	0.2651588	0.0295336	-0.011849
3	0.4709896	**0.3722771**		0.2651588	0.0295336		

Secant Method

INPUT

- x_0 and x_1 that are **near** the root
- tol: the specified tolerance value

IF $|f(x_0)| < |f(x_1)|$ **THEN**

  SWAP x_0 with x_1

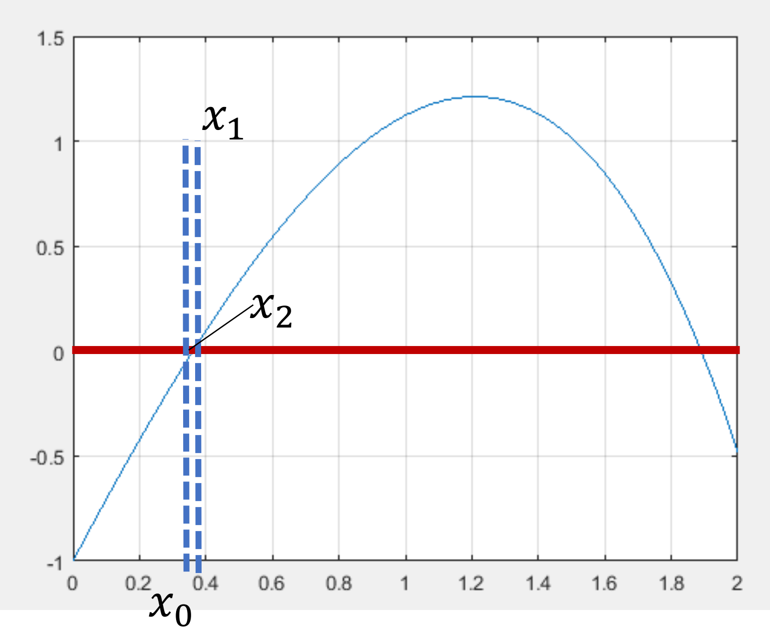
REPEAT

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

  SET $x_0 = x_1$

  SET $x_1 = x_2$

UNTIL $(|f(x_2)| <$ tol$)$



$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$

$x_0 = 1$ $x_1 = 0$

tol $= 1E - 7$ **OR** (0.0000001)

Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	1.00000	0.00000	0.4709896	1.12320	-1.000000	0.2651588	-0.110567
2	0.00000	**0.4709896**	0.3722771	-1.000000	0.2651588	0.0295336	-0.011849
3	0.4709896	**0.3722771**	0.3599043	0.2651588	0.0295336	-0.001294	0.000517
4	0.3722771	**0.3599043**	0.3604239	0.0295336	-0.001294	0.0000552	-0.000002
5	0.3599043	**0.3604239**	0.3604217	-0.001294	0.0000552	0.00000003	0.000000002

Secant Method

INPUT
- x_0 and x_1 that are **near** the root
- tol: the specified tolerance value

```
IF |f(x0)| < |f(x1)| THEN
    SWAP x0 with x1
REPEAT
    SET x2 = x0 − f(x0) (x0−x1)/(f(x0)−f(x1))
    SET x0 = x1
    SET x1 = x2
UNTIL (|f(x2)| <tol)
```

IF $|f(x_0)| < |f(x_1)|$ **THEN**

SWAP x_0 with x_1

REPEAT

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

SET $x_0 = x_1$

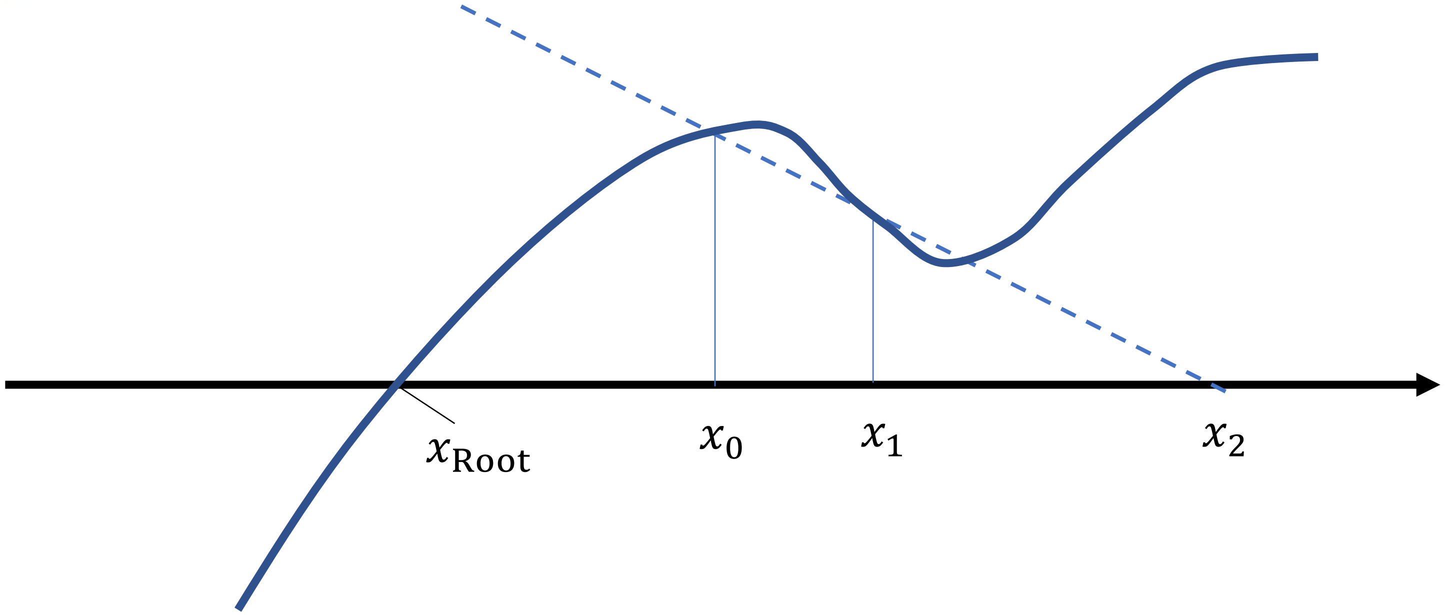
SET $x_1 = x_2$

UNTIL $(|f(x_2)| < \text{tol})$

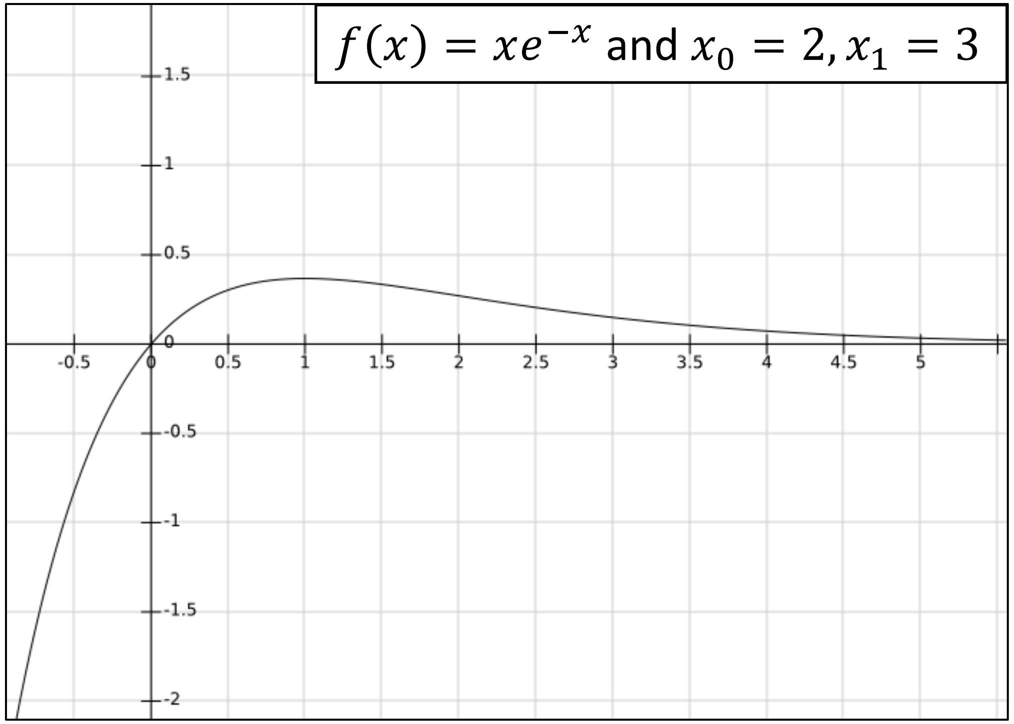
Tolerance Met

$$|f(x_2)| < 1\text{E} - 7$$

If the function is far from linear near the root, the successive iterates can **fly off to points far from the root**



The **bad choice** of x_0 and x_1 or **duplicating a previous result**, may result on an <u>**endless loop**</u>, thus, never reaching an answer



Secant Method - Animation

http://mathfaculty.fullerton.edu/mathews/a2001/Animations/RootFinding/SecantMethod/SecantMethod.html

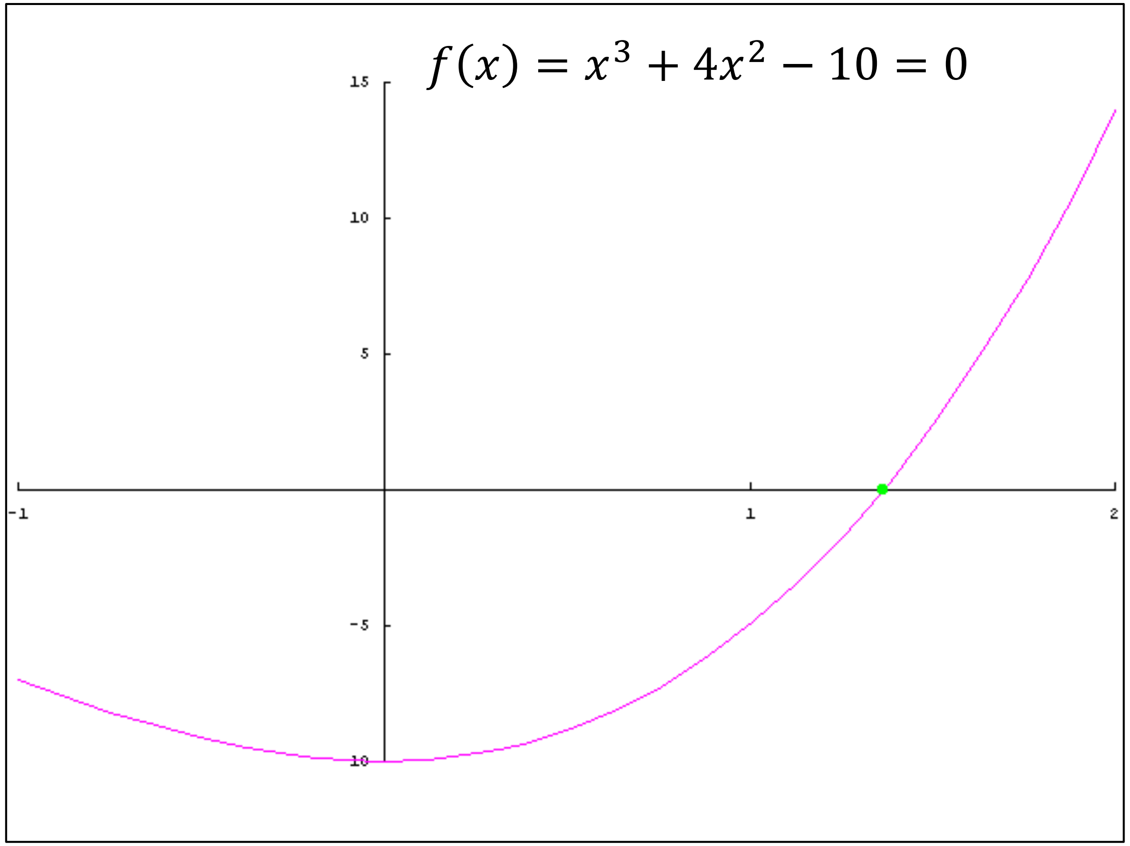
Linear Interpolation Methods

Most functions can be approximated by a straight line over a small interval

False-Position Method

"Regula Falsi" Method

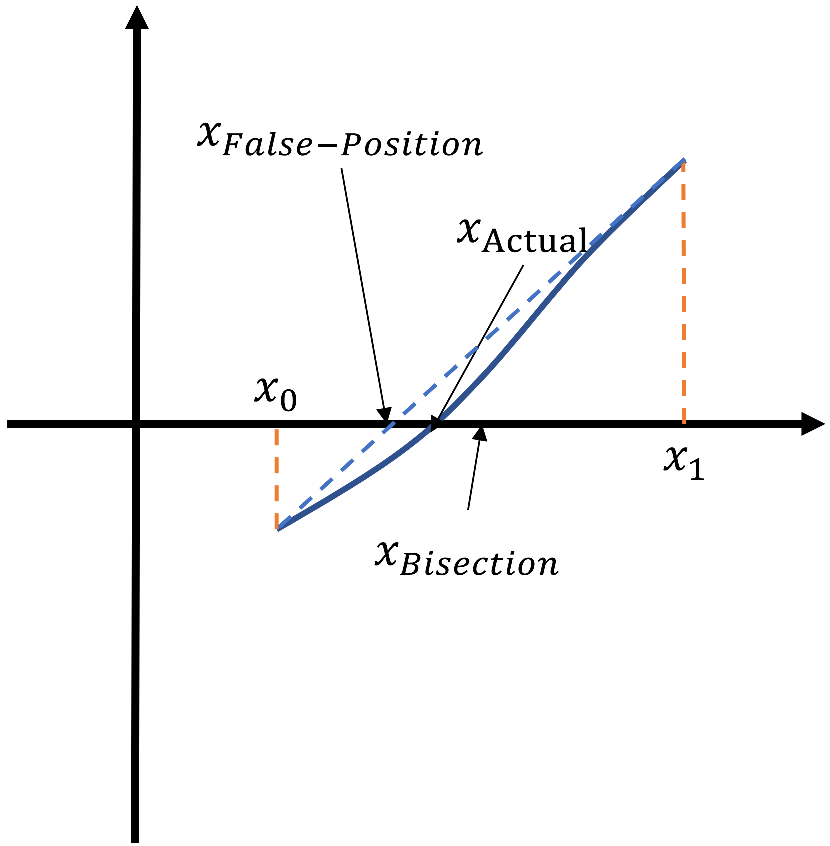
False-Position Method



Problem of Secant Method
If the function is far from linear near the root, the successive iterates can **fly off to points far from the root**

False-Position **begins with two values** for x that ***bracket*** a root

Ensure that the root is bracketed between the two starting values and remains between the successive pairs.



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

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

Similar to bisection except the next iterate is taken at the intersection of a line between the pair of *x*-values and the *x*-axis rather than at the midpoint

Faster than Bisection but more complicated

False-Position (*regula falsi*) Method

INPUT	
	• x_0 and x_1 that **bracket a root** such that $f(x_0)$ and $f(x_1)$ are of opposite sign • `tol`: the specified tolerance value

```
REPEAT
      SET x2 = x0 - f(x0) (x0 - x1) / (f(x0) - f(x1))
      IF f(x2) is of opposite sign to f(x0) THEN
            SET x1 = x2
      ELSE
            SET x0 = x1
UNTIL (|f(x2)| < tol)
```

REPEAT

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

IF $f(x_2)$ is of opposite sign to $f(x_0)$ **THEN**

SET $x_1 = x_2$

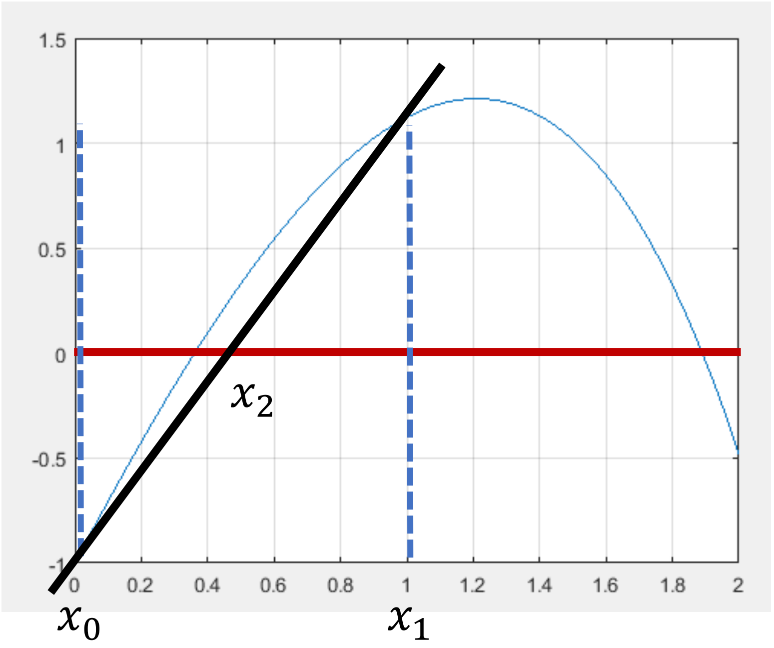
ELSE

SET $x_0 = x_1$

UNTIL $(|f(x_2)| < \texttt{tol})$

NOTES

- Another stopping "termination" criteria is when the pair of points being used are sufficiently close together: "**UNTIL** $(|x_0 - x_1| < \texttt{tol})$"
- The algorithm may **fail** if $f(x)$ is not continuous.



$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$

$x_0 = 1$ $x_1 = 0$

tol $= 1E - 4$ **OR** (0.0001)

Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	0.000000	1.000000		-1.000000	1.12320		

False-Position (*regula falsi*) Method

INPUT

- x_0 and x_1 that **bracket a root** such that $f(x_0)$ and $f(x_1)$ are of opposite sign
- tol: the specified tolerance value

REPEAT

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

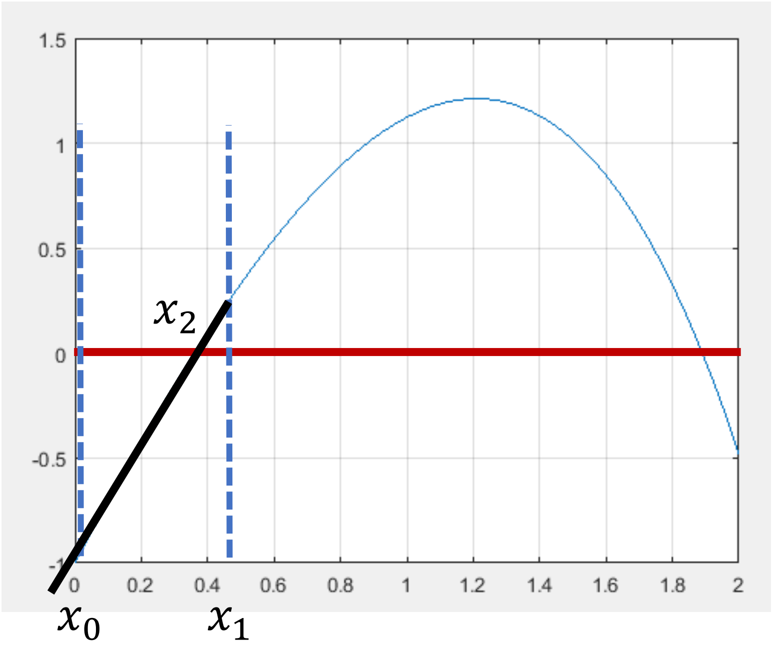
IF $f(x_2)$ is of opposite sign to $f(x_0)$ **THEN**

SET $x_1 = x_2$

ELSE

SET $x_0 = x_1$

UNTIL $(|f(x_2)| <$ tol$)$



$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$

$x_0 = 1$ $x_1 = 0$

tol $= 1E - 4$ **OR** (0.0001)

Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	0.000000	1.000000	0.4709896	-1.000000	1.12320	0.2651588	-0.110567
2	0.000000	0.4709896		-1.000000	0.2651588		

False-Position (*regula falsi*) Method

INPUT

- x_0 and x_1 that **bracket a root** such that $f(x_0)$ and $f(x_1)$ are of opposite sign
- tol: the specified tolerance value

REPEAT

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

IF $f(x_2)$ is of opposite sign to $f(x_0)$ **THEN**

SET $x_1 = x_2$

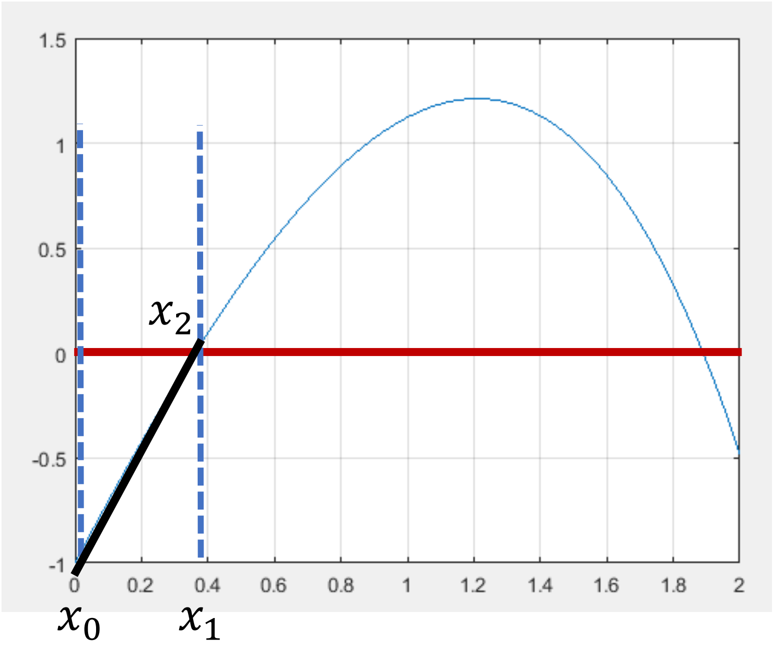
ELSE

SET $x_0 = x_1$

UNTIL $(|f(x_2)| < \text{tol})$

$$f(x) = 3x + \sin x - e^x = 0$$

$x_{\text{Actual}} = 0.36042170296032440136932951583028$



$x_0 = 1$ $x_1 = 0$

`tol` $= 1E - 4$ **OR** (0.0001)

Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	0.000000	1.000000	0.4709896	-1.000000	1.12320	0.2651588	-0.110567
2	0.000000	0.4709896	0.3722771	-1.000000	0.2651588	0.0295336	-0.011849
3	0.000000	0.3722771		-1.000000	0.0295336		

False-Position (*regula falsi*) Method

INPUT

- x_0 and x_1 that **bracket a root** such that $f(x_0)$ and $f(x_1)$ are of opposite sign
- `tol`: the specified tolerance value

REPEAT

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

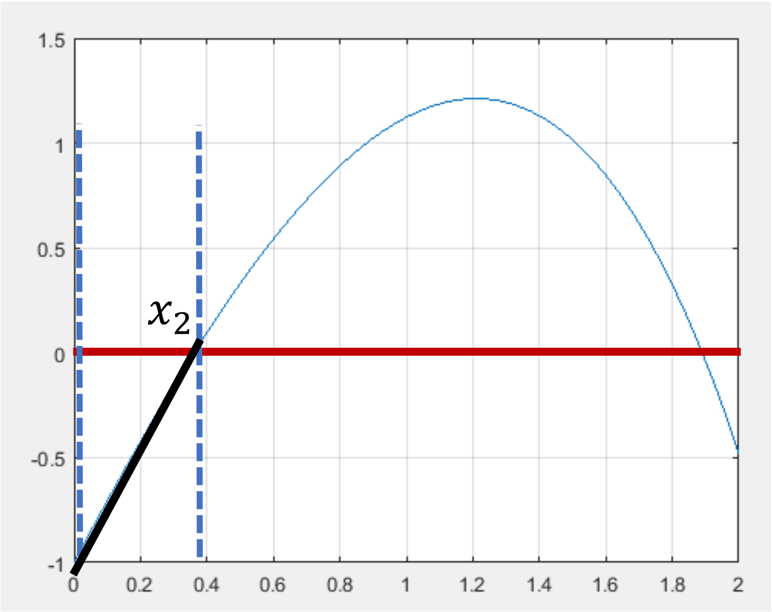
IF $f(x_2)$ is of opposite sign to $f(x_0)$ **THEN**

SET $x_1 = x_2$

ELSE

SET $x_0 = x_1$

UNTIL $(|f(x_2)| <$ `tol`$)$



$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$

$x_0 = 1$ $x_1 = 0$

tol $= 1E - 4$ **OR** (0.0001)

Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	0.000000	1.000000	0.4709896	-1.000000	1.12320	0.2651588	-0.110567
2	0.000000	0.4709896	0.3722771	-1.000000	0.2651588	0.0295336	-0.011849
3	0.000000	0.3722771	0.361598	-1.000000	0.0295336	0.002941	-0.0011760

False-Position (*regula falsi*) Method

INPUT

- x_0 and x_1 that **bracket a root** such that $f(x_0)$ and $f(x_1)$ are of opposite sign
- tol: the specified tolerance value

REPEAT

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

IF $f(x_2)$ is of opposite sign to $f(x_0)$ **THEN**

SET $x_1 = x_2$

ELSE

SET $x_0 = x_1$

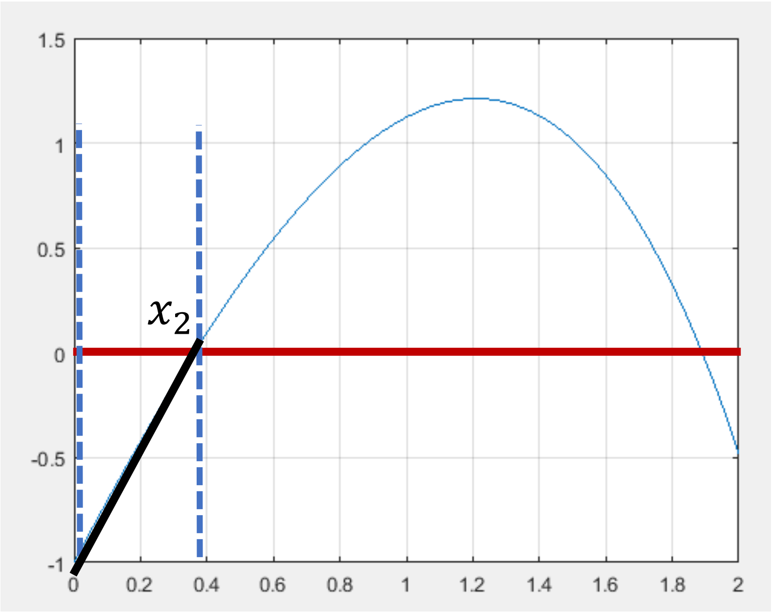
UNTIL $(|f(x_2)| <$ tol$)$

$$f(x) = 3x + \sin x - e^x = 0$$

$x_{\text{Actual}} = 0.36042170296032440136932951583028$

$x_0 = 1 \; x_1 = 0$

$\texttt{tol} = 1E-4$ **OR** (0.0001)



Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	0.000000	1.000000	0.4709896	-1.000000	1.12320	0.2651588	-0.110567
2	0.000000	0.4709896	0.3722771	-1.000000	0.2651588	0.0295336	-0.011849
3	0.000000	0.3722771	0.36159774	-1.000000	0.0295336	0.002941	-0.0011760
4	0.000000	0.36159774	0.36053740	-1.000000	0.002941	0.00028944	-0.0001157
5	0.000000	0.36053740	0.36043307	-1.000000	0.00028944	0.00002845	0.00001373

False-Position (*regula falsi*) Method

INPUT

- x_0 and x_1 that **bracket a root** such that $f(x_0)$ and $f(x_1)$ are of opposite sign
- `tol`: the specified tolerance value

REPEAT

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

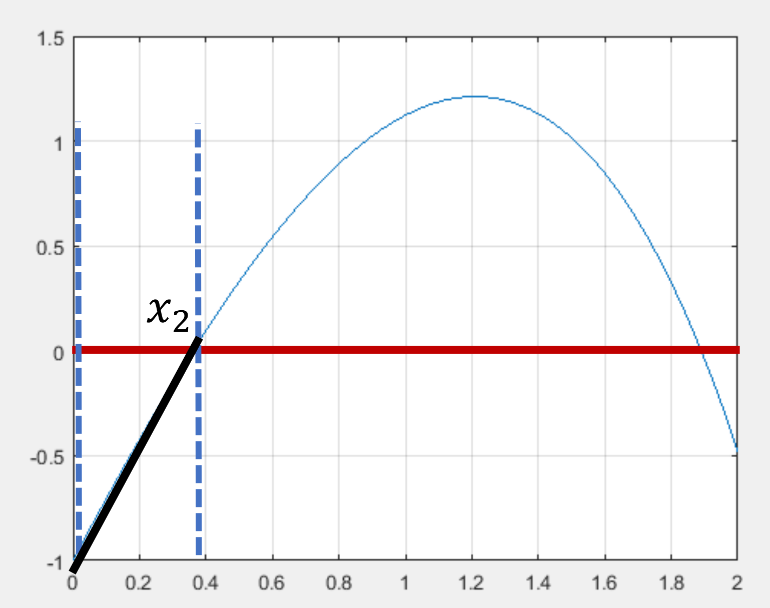
IF $f(x_2)$ is of opposite sign to $f(x_0)$ **THEN**

SET $x_1 = x_2$

ELSE

SET $x_0 = x_1$

UNTIL $(|f(x_2)| < \texttt{tol})$



$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$

$x_0 = 1$ $x_1 = 0$

tol $= 1E-4$ **OR** (0.0001)

Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	0.000000	1.000000	0.4709896	-1.000000	1.12320	0.2651588	-0.110567
2	0.000000	0.4709896	0.3722771	-1.000000	0.2651588	0.0295336	-0.011849
3	0.000000	0.3722771	0.36159774	-1.000000	0.0295336	0.002941	-0.0011760
4	0.000000	0.36159774	0.36053740	-1.000000	0.002941	0.00028944	-0.0001157
5	0.000000	0.36053740	0.36043307	-1.000000	0.00028944	0.00002845	0.00001373

False-Position (*regula falsi*) Method

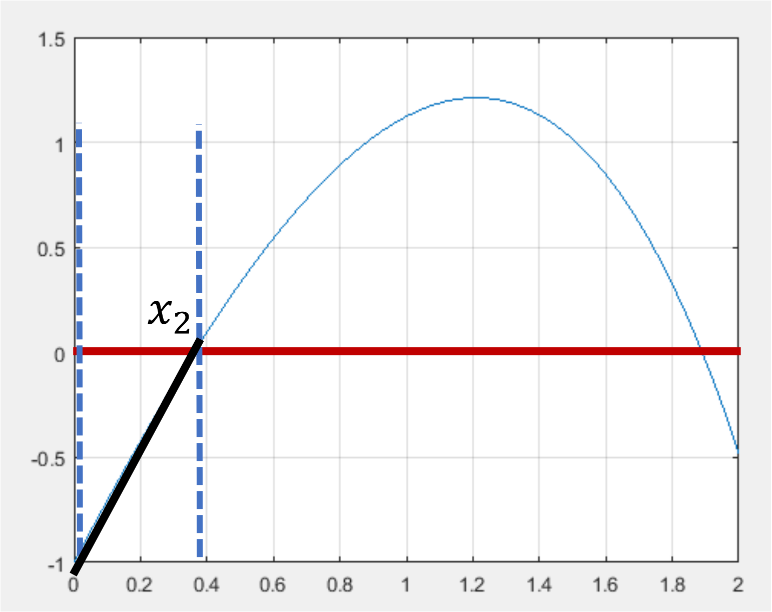
INPUT

- x_0 and x_1 that **bracket a root** such that $f(x_0)$ and $f(x_1)$ are of opposite sign
- tol: the specified tolerance value

```
REPEAT
    SET  x2 = x0 - f(x0) (x0 - x1) / (f(x0) - f(x1))
    IF  f(x2) is of opposite sign to f(x0)  THEN
        SET  x1 = x2
    ELSE
        SET  x0 = x1
UNTIL  (|f(x2)| < tol)
```

Tolerance Met

$$|f(x_2)| < 1\text{E}-4$$



$$f(x) = 3x + \sin x - e^x = 0$$

$x_{\text{Actual}} = 0.36042170296032440136932951583028$

$x_0 = 1 \; x_1 = 0$

tol $= 1E-5$ **OR** (0.00001)

Iter	x_0	x_1	x_2	$f(x_0)$	$f(x_1)$	$f(x_2)$	Actual Error
1	0.000000	1.000000	0.4709896	-1.000000	1.12320	0.2651588	-0.110567
2	0.000000	0.4709896	0.3722771	-1.000000	0.2651588	0.0295336	-0.011849
3	0.000000	0.3722771	0.36159774	-1.000000	0.0295336	0.002941	-0.0011760
4	0.000000	0.36159774	0.36053740	-1.000000	0.002941	0.00028944	-0.0001157
5	0.000000	0.36053740	0.36043307	-1.000000	0.00028944	0.00002845	0.00001373

False Position converges to the root from only one side, slowing it down, especially if that end of the interval is farther from the root.

There is a way to avoid this result, called modified linear interpolation

False-Position (*regula falsi*) Method

INPUT

- x_0 and x_1 that **bracket a root** such that $f(x_0)$ and $f(x_1)$ are of opposite sign
- tol: the specified tolerance value

```
REPEAT
    SET x2 = x0 - f(x0) (x0 - x1) / (f(x0) - f(x1))
    IF f(x2) is of opposite sign to f(x0) THEN
        SET x1 = x2
    ELSE
        SET x0 = x1
UNTIL (|f(x2)| < tol)
```

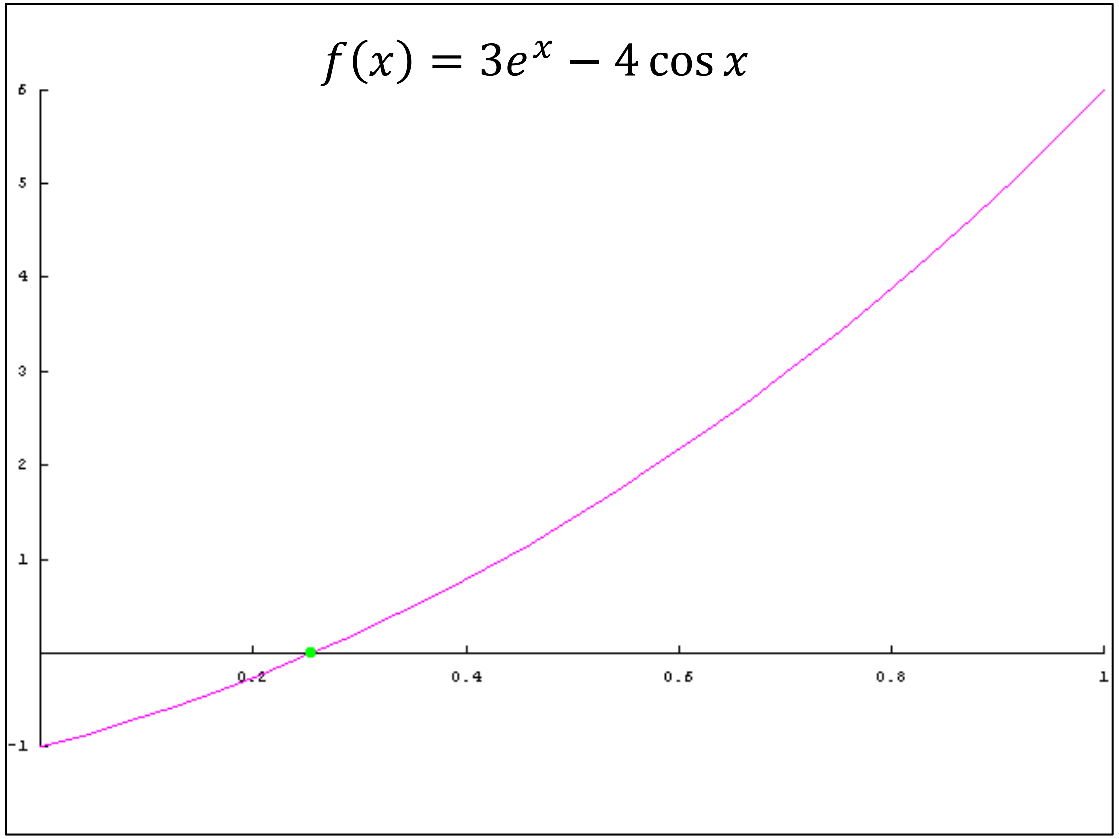
False-Position Method- Animation

http://mathfaculty.fullerton.edu/mathews/a2001/Animations/RootFinding/RegulaFalsi/RegulaFalsi.html

Newton’s Method

Linear approximation of the function using tangent line over small interval

Newton's Method



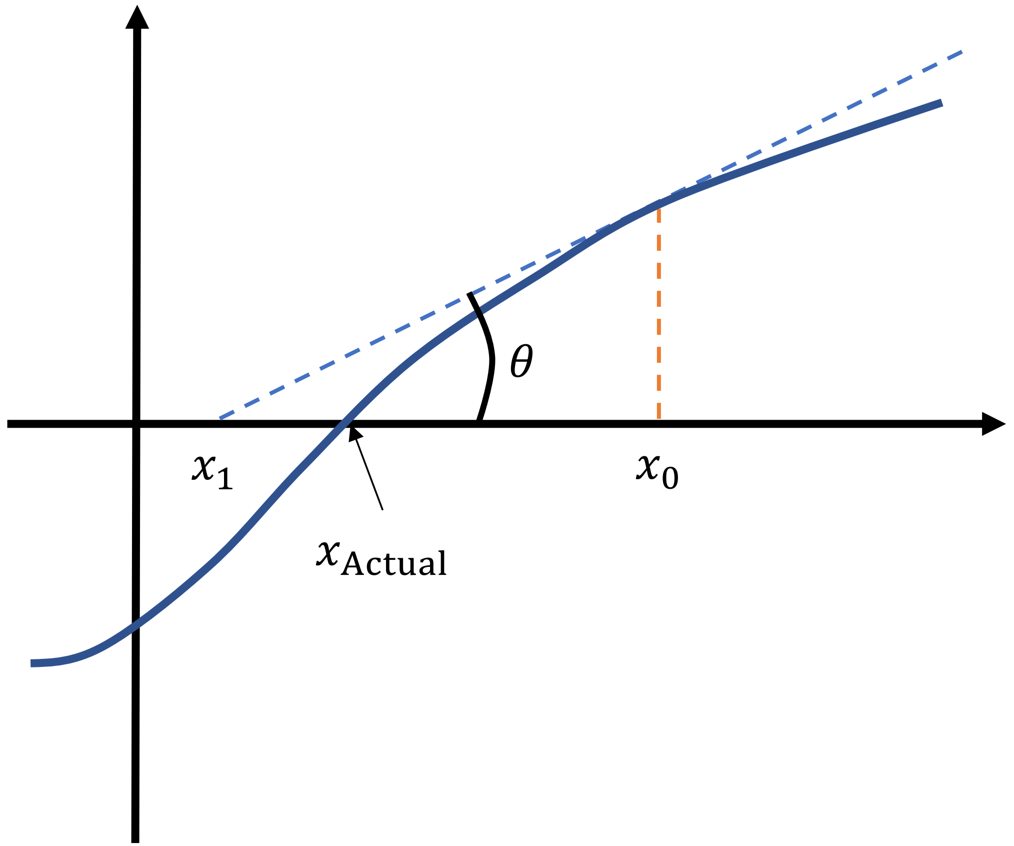
One of the most widely used methods of solving equations

Based on a linear approximation of the function using a **tangent** to the curve

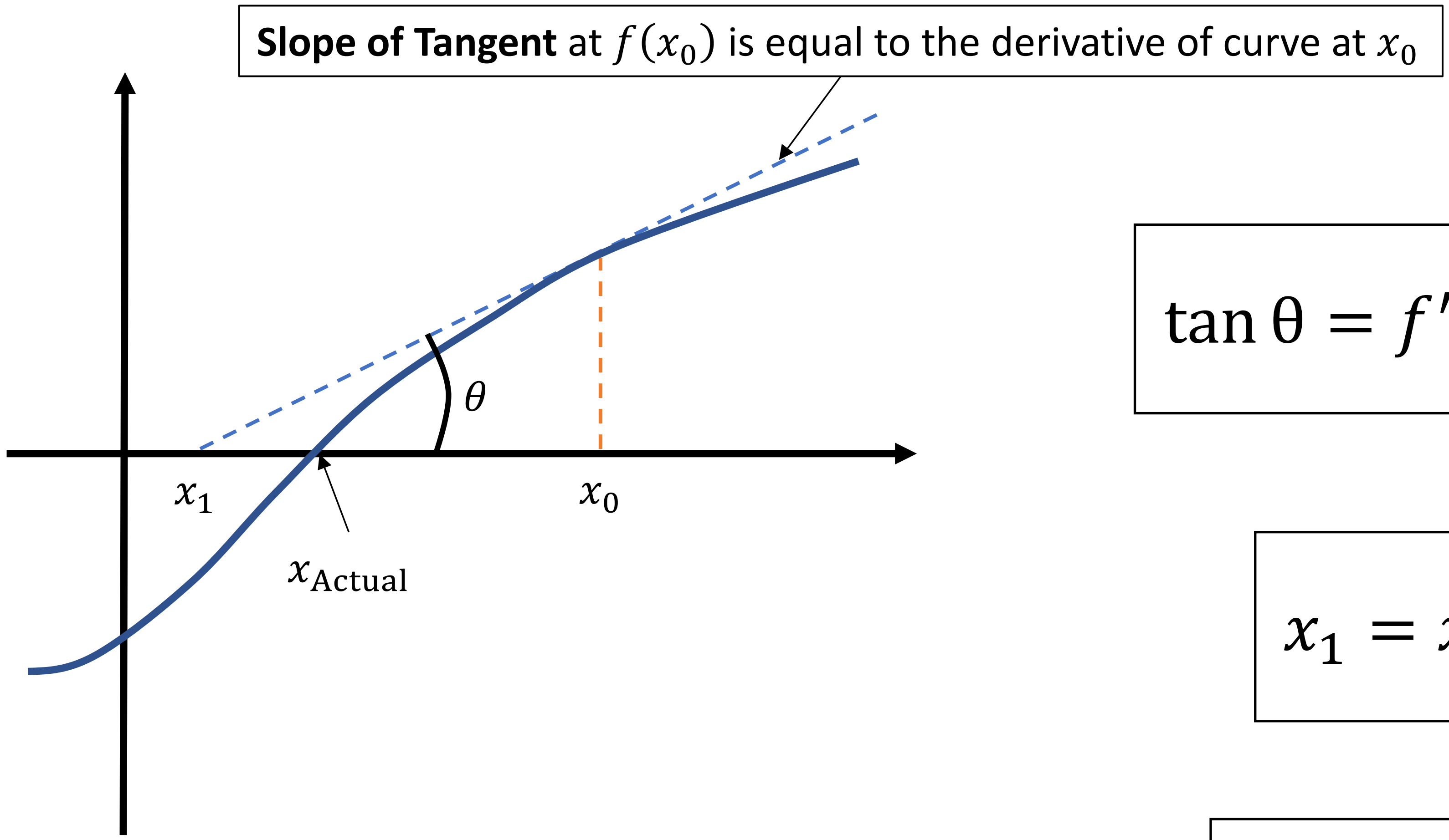
Starting from **a single initial estimate**, x_0 that is **not too far** from a root

Move along **the tangent** to its **intersection with the x-axis**, and take that as the next approximation

Continue until either the successive ***x*-values are sufficiently close** or the **value of the function is sufficiently near zero**



θ
x_1
x_0
x_{Actual}



$$\tan\theta = f'(x_0) = \frac{f(x_0)}{x_0 - x_1}$$

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

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

Newton's Method

INPUT

- x_0 **reasonably close** to the root
- `tol1`: the specified tolerance value for difference between successive *x*-values.
- `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

COMPUTE $f(x_0), f'(x_0)$

SET $x_1 = x_0$

IF $(f(x_0) \neq 0)$ **AND** $(f'(x_0) \neq 0)$ **THEN**

    REPEAT

        SET $x_0 = x_1$

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

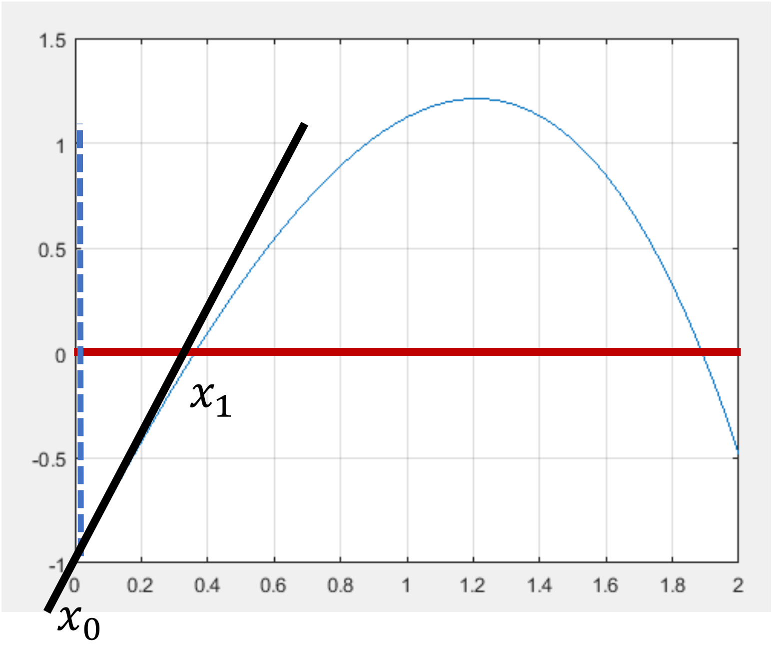
    UNTIL $(|x_1 - x_0| <$ `tol1`$)$ **OR** $(|f(x_1)| <$ `tol2`$)$

NOTES

- The method **may converge to a root different from the expected one** or **diverge** if the starting value **is not close enough** to the root.

$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$



We begin with $x_0 = 0.0$

`tol1` $= 1E - 5$ **OR** (0.00001)

`tol2` $= 1E - 5$ **OR** (0.00001)

Newton's Method

INPUT

- x_0 **reasonably close** to the root
- `tol1`: the specified tolerance value for difference between successive *x*-values.
- `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

COMPUTE $f(x_0), f'(x_0)$
SET $x_1 = x_0$
IF $(f(x_0) \neq 0)$ **AND** $(f'(x_0) \neq 0)$ **THEN**
    REPEAT
        SET $x_0 = x_1$
        SET $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$
    UNTIL $(|x_1 - x_0| <$`tol1`$)$ **OR** $(|f(x_1)| <$`tol2`$)$

$$f(x) = 3x + \sin x - e^x = 0$$

$x_{\text{Actual}} = 0.36042170296032440136932951583028$

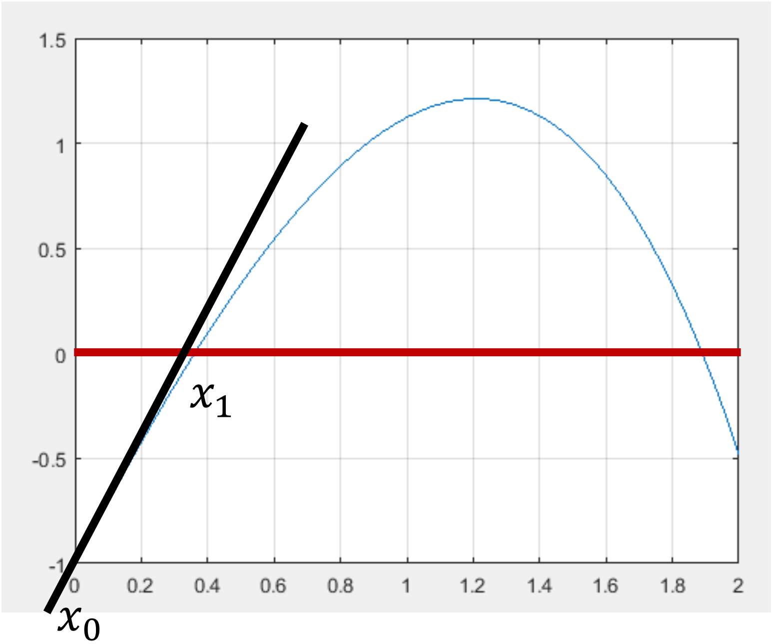
We begin with $x_0 = 0.0$

`tol1` $= 1E - 5$ **OR** (0.00001)

`tol2` $= 1E - 5$ **OR** (0.00001)

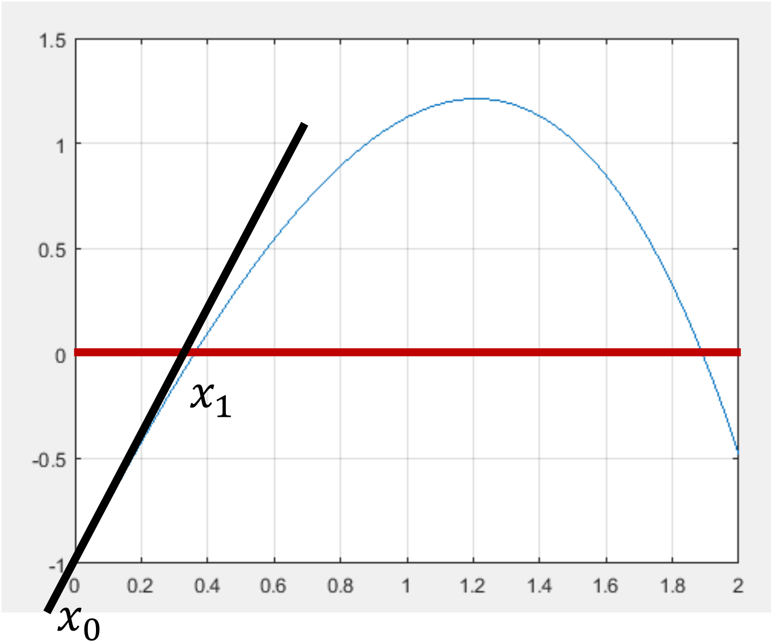
$f(x) = 3x + \sin x - e^x$

$f'(x) = 3 + \cos x - e^x$



Newton's Method	
INPUT	• x_0 **reasonably close** to the root • `tol1`: the specified tolerance value for difference between successive *x*-values. • `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

```
COMPUTE  f(x0), f'(x0)
SET  x1 = x0
IF  (f(x0) ≠ 0)  AND  (f'(x0) ≠ 0)  THEN
    REPEAT
        SET  x0 = x1
        SET  x1 = x0 − f(x0)/f'(x0)
    UNTIL  (|x1 − x0| <tol1)OR(|f(x1)| <tol2)
```



$$f(x) = 3x + \sin x - e^x = 0$$

$x_{\text{Actual}} = 0.36042170296032440136932951583028$

We begin with $x_0 = 0.0$

`tol1` $= 1E-5$ **OR** (0.00001)

`tol2` $= 1E-5$ **OR** (0.00001)

Newton's Method	
INPUT	• x_0 **reasonably close** to the root • `tol1`: the specified tolerance value for difference between successive *x*-values. • `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

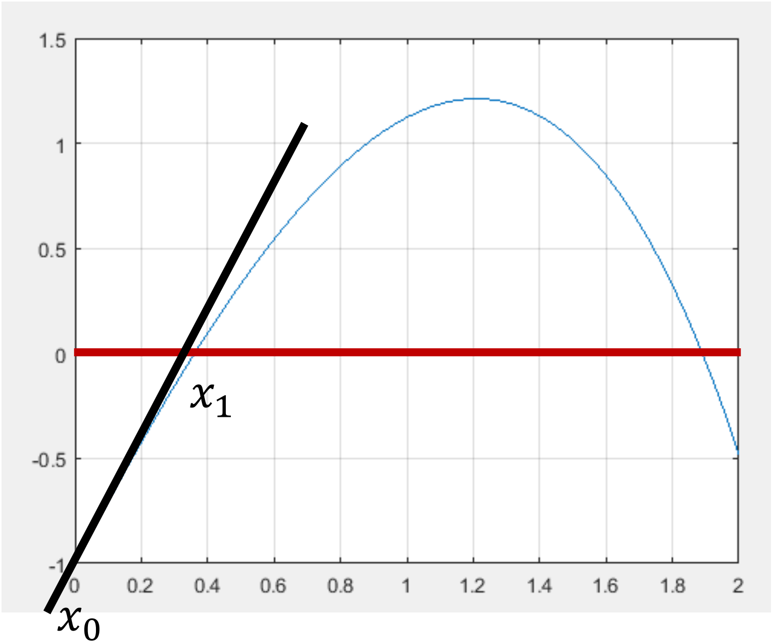
COMPUTE $f(x_0), f'(x_0)$
SET $x_1 = x_0$
IF $(f(x_0) \neq 0)$ **AND** $(f'(x_0) \neq 0)$ **THEN**
  REPEAT
    SET $x_0 = x_1$
    SET $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$
  UNTIL $(|x_1 - x_0| <$`tol1`$)$ **OR** $(|f(x_1)| <$`tol2`$)$

$$f(x) = 3x + \sin x - e^x$$

$$f'(x) = 3 + \cos x - e^x$$

$$x_0 = 0.0, x_1 = 0.0$$

$$f(x_0) = -1.0, f'(x_0) = 3.0$$



$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$

We begin with $x_0 = 0.0$

`tol1` $= 1E-5$ **OR** (0.00001)

`tol2` $= 1E-5$ **OR** (0.00001)

$$f(x) = 3x + \sin x - e^x$$

$$f'(x) = 3 + \cos x - e^x$$

$$x_0 = 0.0, x_1 = 0.0$$

$$f(x_0) = -1.0, f'(x_0) = 3.0$$

Iteration 1: $x_0 = 0.0, x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0.0 - \frac{-1.0}{3.0} = 0.33333$

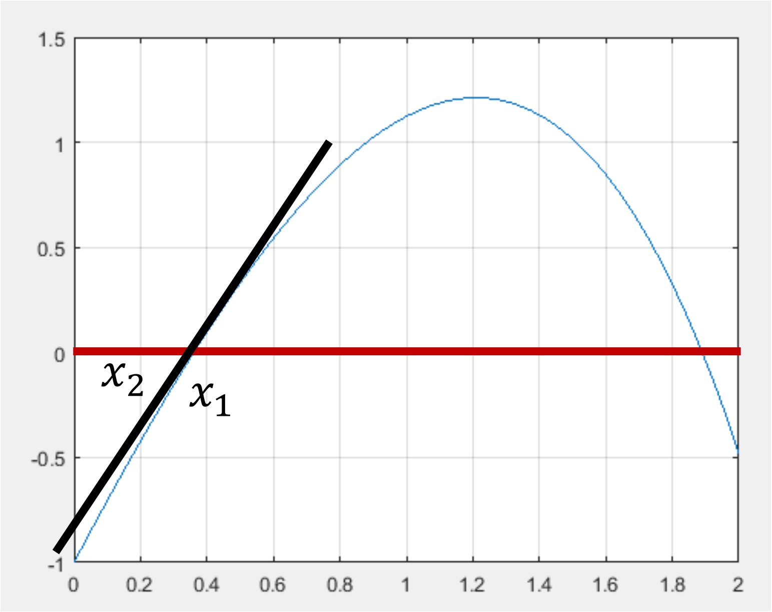
$|0.33333 - 0.0| < 0.00001$ **OR** $|-0.068418| < 0.00001$

Newton's Method

INPUT

- x_0 **reasonably close** to the root
- `tol1`: the specified tolerance value for difference between successive *x*-values.
- `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

```
COMPUTE  f(x0), f'(x0)
SET  x1 = x0
IF  (f(x0) ≠ 0)  AND  (f'(x0) ≠ 0)  THEN
    REPEAT
        SET  x0 = x1
        SET  x1 = x0 − f(x0)/f'(x0)
    UNTIL  (|x1 − x0| <tol1)OR(|f(x1)| <tol2)
```



$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$

We begin with $x_0 = 0.0$

`tol1` $= 1E - 5$ **OR** (0.00001)

`tol2` $= 1E - 5$ **OR** (0.00001)

$$f(x) = 3x + \sin x - e^x$$

$$f'(x) = 3 + \cos x - e^x$$

$$x_0 = 0.0, x_1 = 0.0$$

$$f(x_0) = -1.0, f'(x_0) = 3.0$$

Iteration 1: $x_0 = 0.0, x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0.0 - \frac{-1.0}{3.0} = 0.33333$

Iteration 2: $x_1 = 0.33333, x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 0.33333 - \frac{-0.068418}{2.54934} = 0.36017$

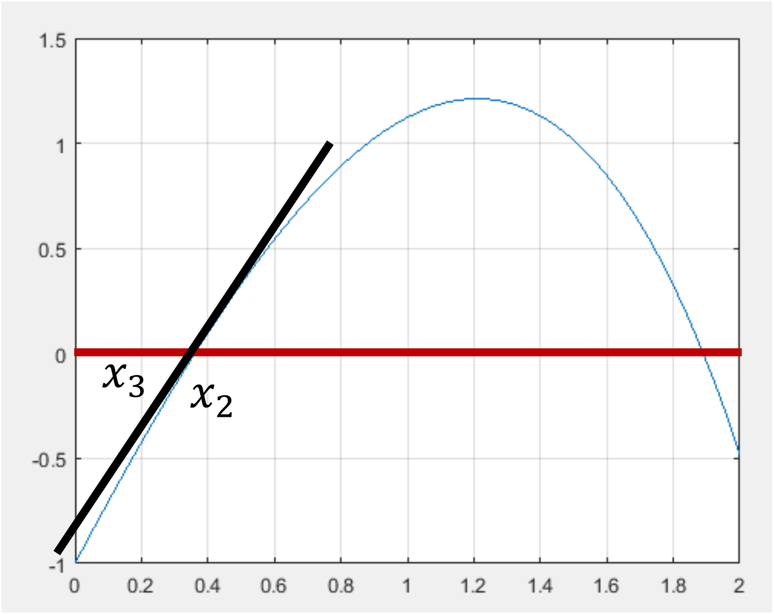
$|0.36017 - 0.33333| < 0.00001$ **OR** $|-0.0006279| < 0.00001$

Newton's Method

INPUT

- x_0 **reasonably close** to the root
- `tol1`: the specified tolerance value for difference between successive x-values.
- `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

```
COMPUTE  f(x0), f'(x0)
SET  x1 = x0
IF  (f(x0) ≠ 0)  AND  (f'(x0) ≠ 0)  THEN
    REPEAT
        SET  x0 = x1
        SET  x1 = x0 - f(x0)/f'(x0)
    UNTIL  (|x1 - x0| <tol1)OR(|f(x1)| <tol2)
```



$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$

We begin with $x_0 = 0.0$

`tol1` $= 1E-5$ **OR** (0.00001)

`tol2` $= 1E-5$ **OR** (0.00001)

$$f(x) = 3x + \sin x - e^x$$

$$f'(x) = 3 + \cos x - e^x$$

$$x_0 = 0.0, x_1 = 0.0$$

$$f(x_0) = -1.0, f'(x_0) = 3.0$$

Newton's Method

INPUT

- x_0 **reasonably close** to the root
- `tol1`: the specified tolerance value for difference between successive *x*-values.
- `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

```
COMPUTE  f(x0), f'(x0)
SET  x1 = x0
IF  (f(x0) ≠ 0)  AND  (f'(x0) ≠ 0)  THEN
    REPEAT
        SET  x0 = x1
        SET  x1 = x0 − f(x0)/f'(x0)
    UNTIL  (|x1 − x0| <tol1)OR(|f(x1)| <tol2)
```

Iteration 1: $x_0 = 0.0, x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0.0 - \frac{-1.0}{3.0} = 0.33333$

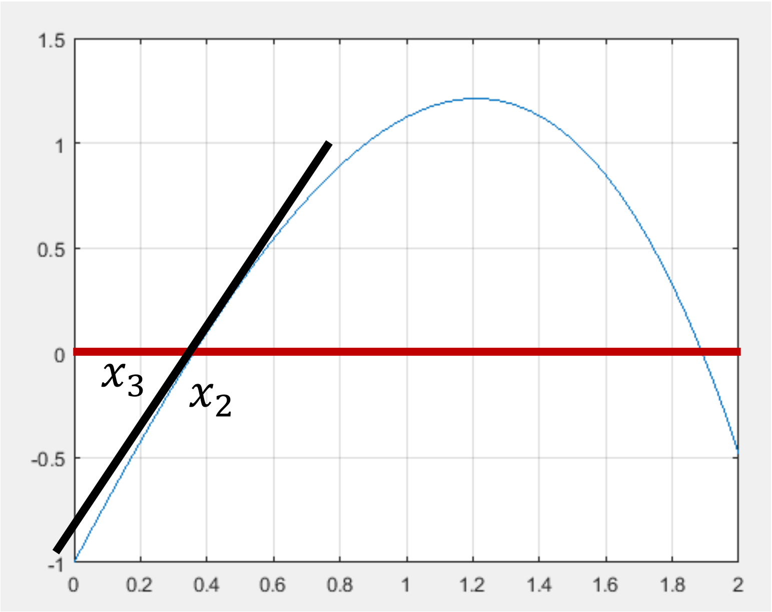
Iteration 2: $x_1 = 0.33333, x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 0.33333 - \frac{-0.068418}{2.54934} = 0.36017$

Iteration 3: $x_2 = 0.36017, x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 0.36017 - \frac{-0.0006279}{2.50226} = 0.3604217$

$|0.3604217 - 0.36017| < 0.00001$ **OR** $|-0.00000005| < 0.00001$

$$f(x) = 3x + \sin x - e^x = 0$$

$$x_{\text{Actual}} = 0.36042170296032440136932951583028$$



We begin with $x_0 = 0.0$

`tol1` $= 1E - 5$ **OR** (0.00001)

`tol2` $= 1E - 5$ **OR** (0.00001)

$$f(x) = 3x + \sin x - e^x$$

$$f'(x) = 3 + \cos x - e^x$$

$$x_0 = 0.0, x_1 = 0.0$$

$$f(x_0) = -1.0, f'(x_0) = 3.0$$

Iteration 1: $x_0 = 0.0, x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0.0 - \frac{-1.0}{3.0} = 0.33333$

Iteration 2: $x_1 = 0.33333, x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 0.33333 - \frac{-0.068418}{2.54934} = 0.36017$

Iteration 3: $x_2 = 0.36017, x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 0.36017 - \frac{-0.0006279}{2.50226} = 0.3604217$

$|0.3604217 - 0.36017| < 0.00001$ **OR** $|-0.00000005| < 0.00001$

Actual Error = $x_{Actual} - x_3 = 0.000000002960324$

Absolute Error = $|x_{Actual} - x_3| = 0.000000002960324$

Newton's Method

INPUT	
	• x_0 **reasonably close** to the root • `tol1`: the specified tolerance value for difference between successive x-values. • `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

```
COMPUTE  f(x0), f'(x0)
SET  x1 = x0
IF  (f(x0) ≠ 0)  AND  (f'(x0) ≠ 0)  THEN
    REPEAT
        SET  x0 = x1
        SET  x1 = x0 − f(x0)/f'(x0)
    UNTIL  (|x1 − x0| <tol1)OR(|f(x1)| <tol2)
```

Newton's algorithm is widely used because, at least in the near neighborhood of a root, it is **more rapidly convergent** than any of the methods discussed so far

The number of decimal places of accuracy **nearly doubles** at each iteration

There is the need for two function evaluations at each step $f(x)$ and $f'(x)$ and we must obtain the derivative function at the start

Finding $f'(x)$ may be difficult. Computer algebra systems can be a real help

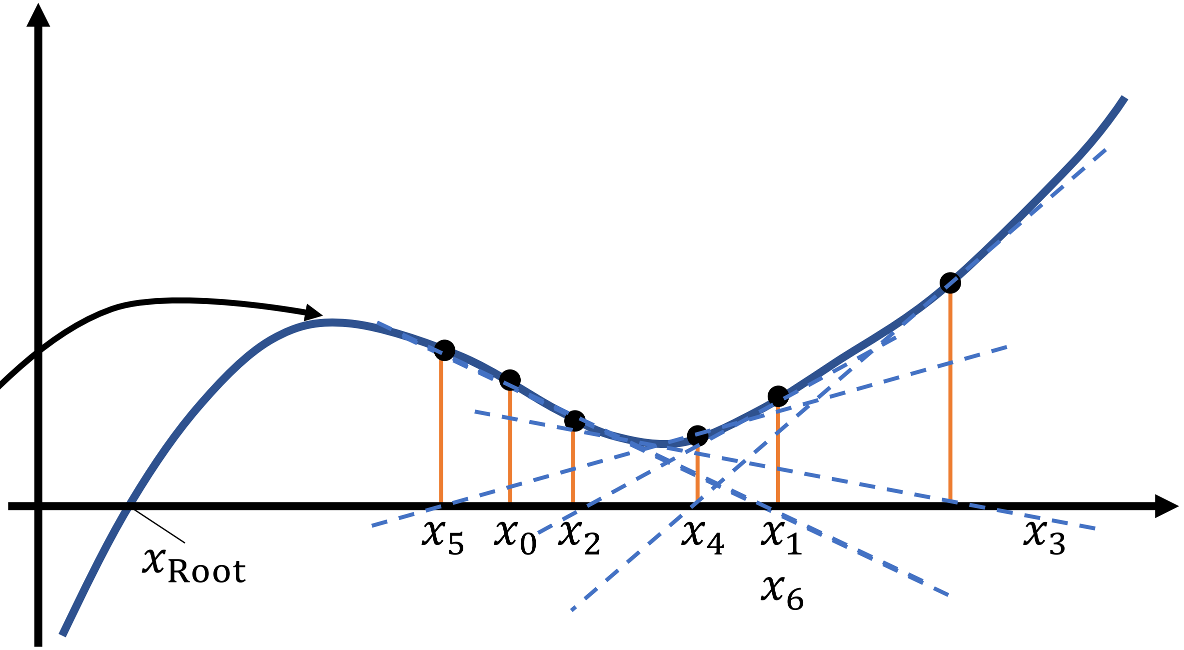
In some cases Newton's method **<u>will not converge</u>**

The **bad choice** of x_0 may never lead us to reach an answer; we are stuck in an **infinite loop**

Reaching **local minimum or local maximum** of the function, **<u>the answer will fly off to infinity</u>**

$$f'(x_0) = 0$$

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



Relating Newton's to Other Methods

Linear Interpolation Methods

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

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

The denominator of the fraction is an **approximation** of the derivative at x_0

Newton's Method

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

$$f'(x_0) = \lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

$$x_1 = x_0 + h$$

$$f'(x_0) = \lim_{(x_1 - x_0) \to 0} \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$x_2 = x_0 - \frac{f(x_0)}{\lim_{(x_1 - x_0) \to 0} \frac{f(x_1) - f(x_0)}{x_1 - x_0}}$$

Secant method looks like Newton’s and **its good to use when the derivative is not easy to achieve**

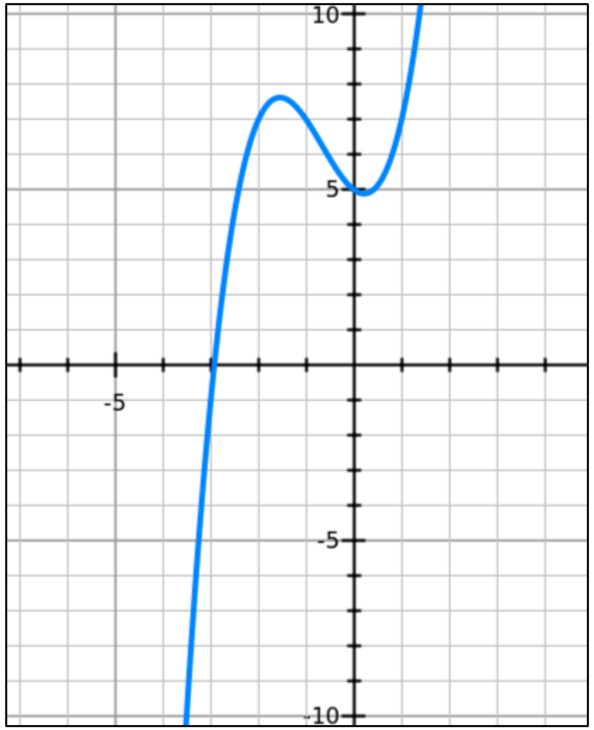
Secant Method

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

Newton’s Method

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

Newton's method **works with complex roots** if we give it a complex value for the starting value x_0



$$f(x) = x^3 + 2x^2 - x + 5 = 0$$

$$x_{\text{Actual}} = -2.9259, 0.46293 \pm 1.22254i$$

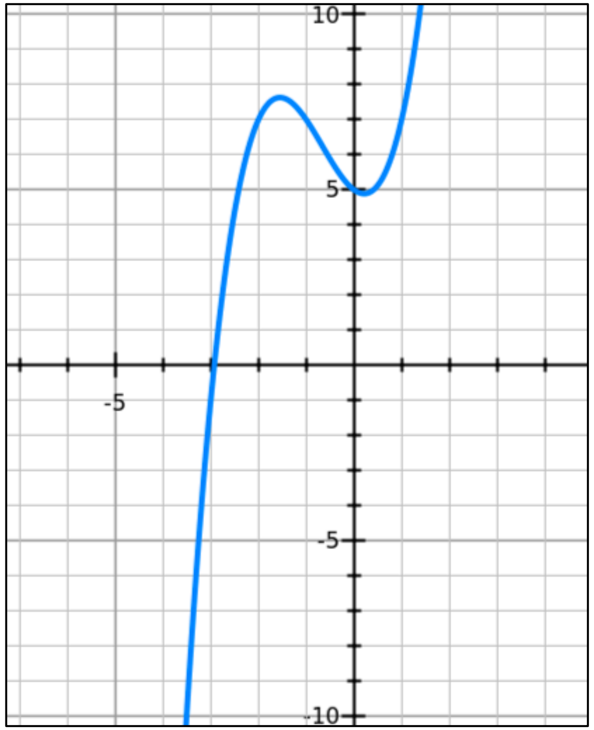
We begin with $x_0 = 1 + i$

`tol1` $= 1E - 4$ **OR** (0.0001)

`tol2` $= 1E - 4$ **OR** (0.0001)

Newton's Method	
INPUT	• x_0 **reasonably close** to the root • `tol1`: the specified tolerance value for difference between successive *x*-values. • `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

COMPUTE $f(x_0), f'(x_0)$
SET $x_1 = x_0$
IF $(f(x_0) \neq 0)$ **AND** $(f'(x_0) \neq 0)$ **THEN**
    REPEAT
        SET $x_0 = x_1$
        SET $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$
    UNTIL $(|x_1 - x_0| <$`tol1`$)$**OR**$(|f(x_1)| <$`tol2`$)$



$$f(x) = x^3 + 2x^2 - x + 5 = 0$$

$$x_{\text{Actual}} = -2.9259, 0.46293 \pm 1.22254i$$

We begin with $x_0 = 1 + i$

`tol1` $= 1E - 4$ **OR** (0.0001)

`tol2` $= 1E - 4$ **OR** (0.0001)

$$f(x) = x^3 + 2x^2 - x + 5$$

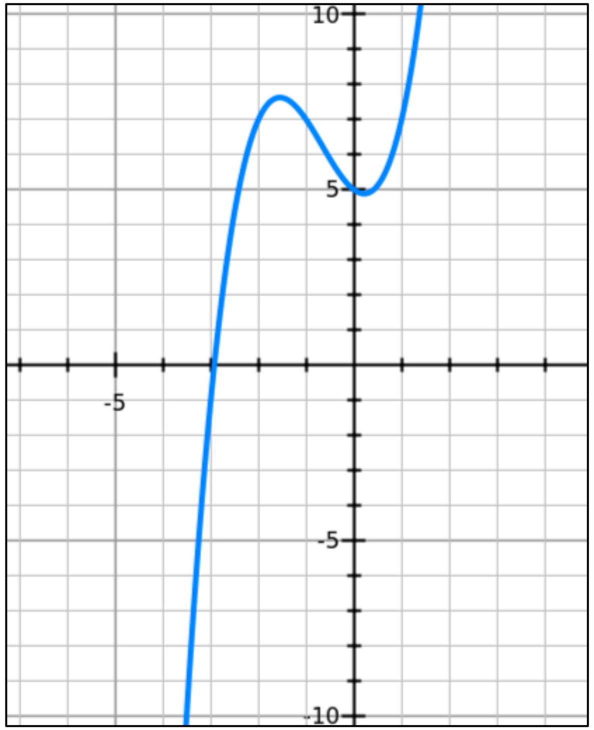
$$f'(x) = 3x^2 + 4x - 1$$

$$x_0 = 1 + i, x_1 = 1 + i$$

$$f(x_0) = 2 + 5i, f'(x_0) = 3 + 10i$$

Newton's Method	
INPUT	• x_0 **reasonably close** to the root • `tol1`: the specified tolerance value for difference between successive x-values. • `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

COMPUTE $f(x_0), f'(x_0)$
SET $x_1 = x_0$
IF $(f(x_0) \neq 0)$ **AND** $(f'(x_0) \neq 0)$ **THEN**
    REPEAT
        SET $x_0 = x_1$
        SET $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$
    UNTIL $(|x_1 - x_0| <$ `tol1`$)$ **OR** $(|f(x_1)| <$ `tol2`$)$



$$f(x) = x^3 + 2x^2 - x + 5 = 0$$

$$x_{\text{Actual}} = -2.9259, 0.46293 \pm 1.22254i$$

We begin with $x_0 = 1 + i$

$\texttt{tol1} = 1E - 4$ **OR** (0.0001)

$\texttt{tol2} = 1E - 4$ **OR** (0.0001)

$$f(x) = x^3 + 2x^2 - x + 5$$

$$f'(x) = 3x^2 + 4x - 1$$

$$x_0 = 1 + i, x_1 = 1 + i$$

$$f(x_0) = 2 + 5i, f'(x_0) = 3 + 10i$$

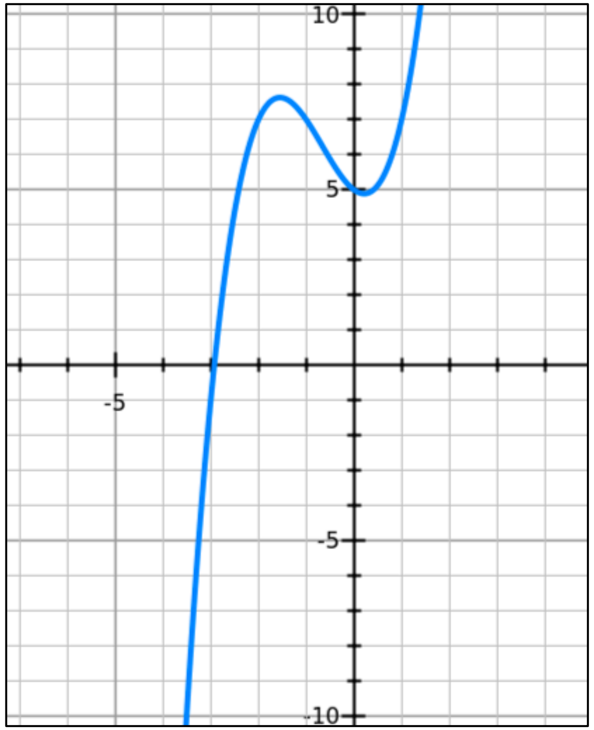
Iteration 1: $x_0 = 1 + i, x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = (1 + i) - \frac{2+5i}{3+10i} = 0.486238 + 1.04587i$

Newton's Method

INPUT

- x_0 **reasonably close** to the root
- `tol1`: the specified tolerance value for difference between successive x-values.
- `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

```
COMPUTE  f(x_0), f'(x_0)
SET  x_1 = x_0
IF  (f(x_0) ≠ 0)  AND  (f'(x_0) ≠ 0)  THEN
    REPEAT
        SET  x_0 = x_1
        SET  x_1 = x_0 − f(x_0)/f'(x_0)
    UNTIL  (|x_1 − x_0| <tol1)OR(|f(x_1)| <tol2)
```



$$f(x) = x^3 + 2x^2 - x + 5 = 0$$

$$x_{\text{Actual}} = -2.9259, 0.46293 \pm 1.22254i$$

We begin with $x_0 = 1 + i$

`tol1` $= 1E - 4$ **OR** (0.0001)

`tol2` $= 1E - 4$ **OR** (0.0001)

$$f(x) = x^3 + 2x^2 - x + 5$$

$$f'(x) = 3x^2 + 4x - 1$$

$$x_0 = 1 + i, x_1 = 1 + i$$

$$f(x_0) = 2 + 5i, f'(x_0) = 3 + 10i$$

Iteration 1: $x_0 = 1 + i, x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = (1 + i) - \frac{2+5i}{3+10i} = 0.486238 + 1.04587i$

Iteration 2: $x_1 = 0.486238 + 1.04587i, x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} =$

$$= 0.486238 + 1.04587i - \frac{0.519354 - 0.402202i}{-1.62730 + 7.23473i}$$

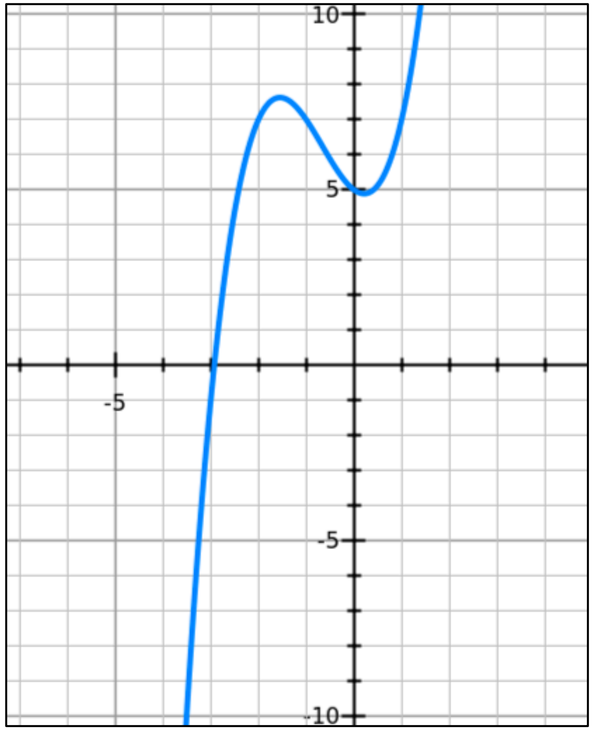
$$= 0.448139 + 1.23665i$$

Newton's Method

INPUT

- x_0 **reasonably close** to the root
- `tol1`: the specified tolerance value for difference between successive x-values.
- `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

```
COMPUTE f(x0), f'(x0)
SET x1 = x0
IF (f(x0) ≠ 0) AND (f'(x0) ≠ 0) THEN
    REPEAT
        SET x0 = x1
        SET x1 = x0 − f(x0)/f'(x0)
    UNTIL (|x1 − x0| <tol1)OR(|f(x1)| <tol2)
```



$$f(x) = x^3 + 2x^2 - x + 5 = 0$$

$$x_{\text{Actual}} = -2.9259, 0.46293 \pm 1.22254i$$

We begin with $x_0 = 1 + i$

`tol1` $= 1E - 4$ **OR** (0.0001)

`tol2` $= 1E - 4$ **OR** (0.0001)

$$f(x) = x^3 + 2x^2 - x + 5$$

$$f'(x) = 3x^2 + 4x - 1$$

$$x_0 = 1 + i, x_1 = 1 + i$$

$$f(x_0) = 2 + 5i, f'(x_0) = 3 + 10i$$

Iteration 1: $x_0 = 1 + i, x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = (1 + i) - \frac{2+5i}{3+10i} = 0.486238 + 1.04587i$

Iteration 2: $x_1 = 0.486238 + 1.04587i, x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} =$

$$= 0.486238 + 1.04587i - \frac{0.519354 - 0.402202i}{-1.62730 + 7.23473i}$$

$$= 0.448139 + 1.23665i$$

Iteration 3: $x_2 = 0.448139 + 1.23665i, x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} =$

$$= 0.448139 + 1.23665i - \frac{-0.0711105 - 0.166035i}{-3.19287 + 8.27175i}$$

$$= 0.462720 + 1.22242i$$

Newton's Method

INPUT

- x_0 **reasonably close** to the root
- `tol1`: the specified tolerance value for difference between successive x-values.
- `tol2`: the specified tolerance value for how close $f(x_1)$ to zero.

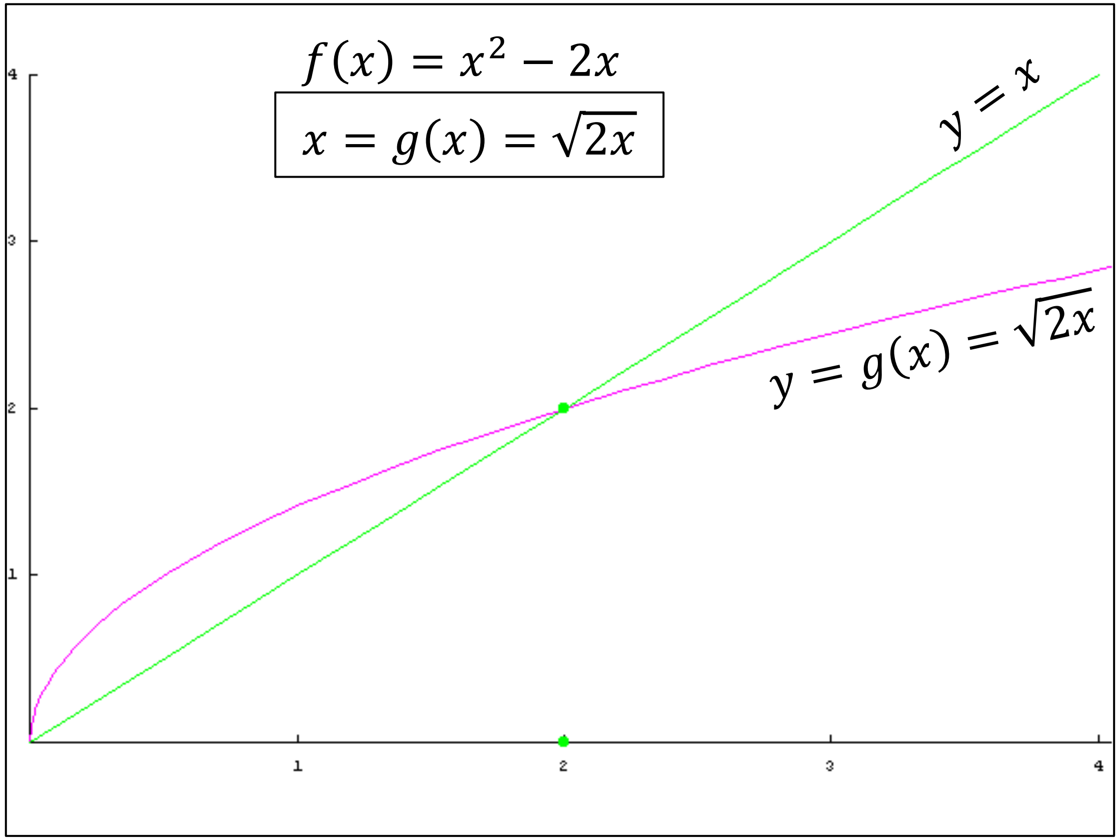
```
COMPUTE  f(x0), f'(x0)
SET  x1 = x0
IF  (f(x0) ≠ 0)  AND  (f'(x0) ≠ 0)  THEN
    REPEAT
        SET  x0 = x1
        SET  x1 = x0 − f(x0)/f'(x0)
    UNTIL  (|x1 − x0| <tol1)OR(|f(x1)| <tol2)
```

Newton’s Method- Animation

http://mathfaculty.fullerton.edu/mathews/a2001/Animations/RootFinding/NewtonMethod/NewtonMethod.html

Fixed-Point Iteration Method

Fixed-Point Iteration Method



A very useful way to get a root of a given $f(x)$

We rearrange $f(x)$ into an **equivalent** form $x = g(x)$

If r is a root of $f(x)$,
then $f(r) = r - g(r) = 0 \implies r = g(r)$

r is a fixed-point for function g

$x_{n+1} = g(x_n) \qquad n = 0,1,2,3, \ldots$

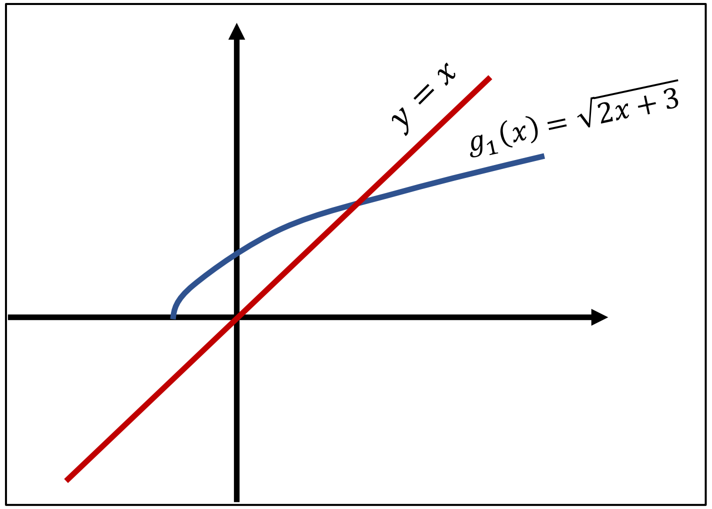
For given equation $f(x) = 0$, there may be **many equivalent** fixed-point problems $x = g(x)$ with different choice of $g(x)$

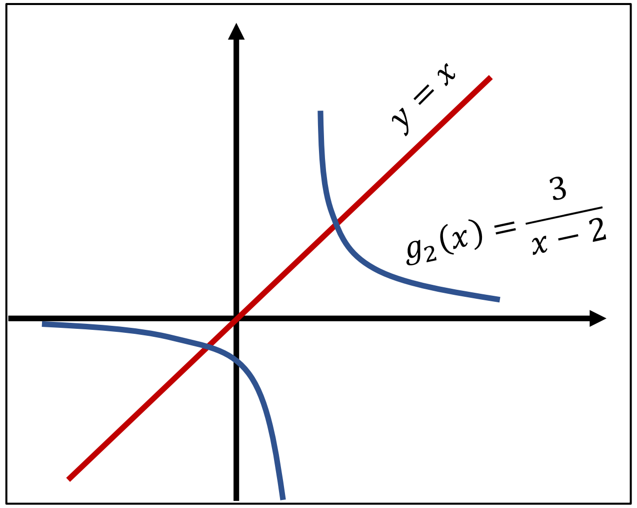
$$f(x) = x^2 - 2x - 3 = 0$$

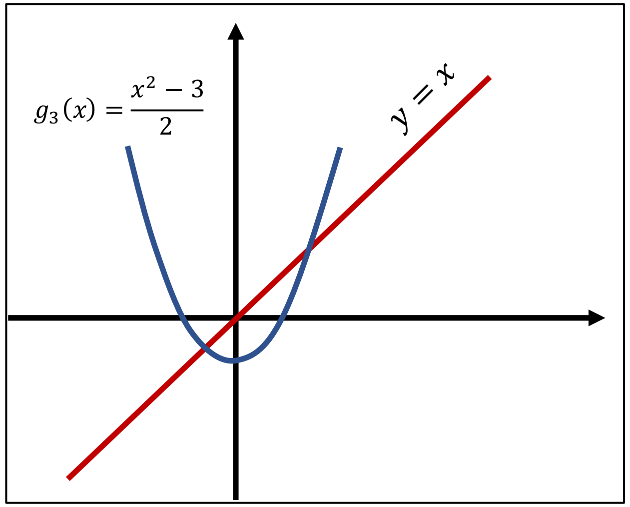
$$g_1(x) = \sqrt{2x+3}$$

$$g_2(x) = \frac{3}{x-2}$$

$$g_3(x) = \frac{x^2-3}{2}$$







Fixed-Point Iteration Method ($x = g(x)$ Method)

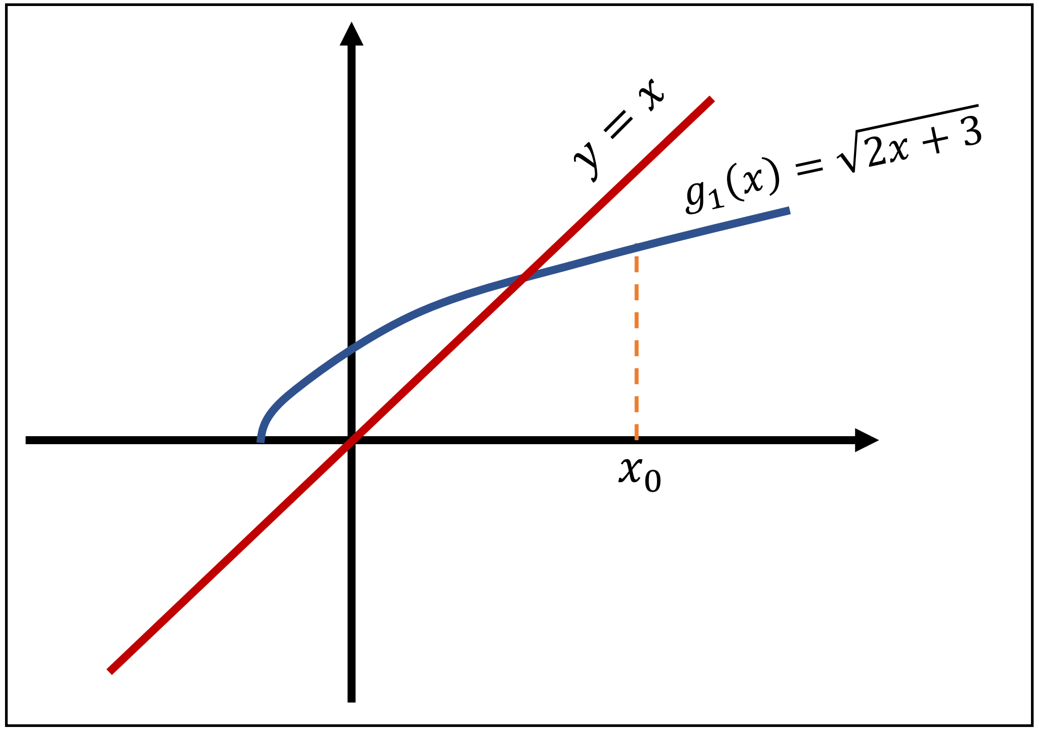
INPUT

- x_1 **reasonably close** to the root
- `tol`: the specified tolerance value

```
REARRANGE f(x) INTO x = g(x)
SET x2 = x1
REPEAT
    SET x1 = x2
    SET x2 = g(x1)
    UNTIL (|x1 − x0| <tol)
```

NOTES

- The method **may converge to a root different from the expected one** or **diverge**.
- Different rearrangements will converge at different rates.



$$f(x) = x^2 - 2x - 3 = 0$$

$x_{\text{Actual}} = -1, 3$

$$g_1(x) = \sqrt{2x+3}$$

We begin with $x_0 = 4$

tol $= 1E - 4$ **OR** (0.0001)

$$x_{n+1} = g(x_n)$$

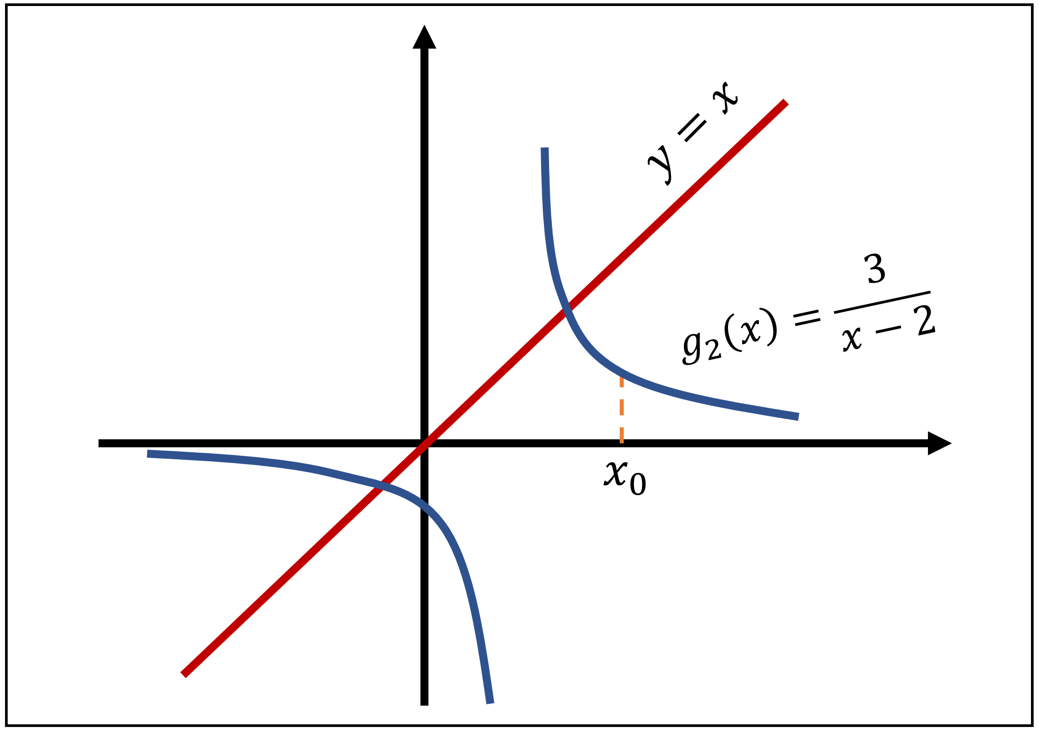
$$x_0 = 4,$$
$$x_1 = \sqrt{2(4)+3} = \sqrt{11} = 3.31663$$
$$x_2 = \sqrt{2(3.31663\ )+3} = \sqrt{9.63325} = 3.10375$$
$$x_3 = \sqrt{2(3.10375)+3} = \sqrt{9.20750} = 3.03439$$
$$x_4 = \sqrt{2(3.31663\ )+3} = \sqrt{9.06877} = 3.01144$$
$$x_5 = \sqrt{2(3.01144)+3} = \sqrt{9.02288} = 3.00381$$
$$x_6 = \sqrt{2(3.00381)+3} = \sqrt{9.00762} = 3.00127$$
$$x_7 = \sqrt{2(3.00127)+3} = \sqrt{9.00254} = 3.00042$$
$$x_8 = \sqrt{2(3.00042)+3} = \sqrt{9.00084} = 3.00013$$
$$x_9 = \sqrt{2(3.00013)+3} = \sqrt{9.00026} = 3.00004$$

Fixed-Point Iteration Method ($x = g(x)$ Method)

INPUT

- x_1 **reasonably close** to the root
- tol: the specified tolerance value

REARRANGE $f(x)$ **INTO** $x = g(x)$
SET $x_2 = x_1$
REPEAT
    SET $x_1 = x_2$
    SET $x_2 = g(x_1)$
    UNTIL $(|x_1 - x_0| <$ tol)



$$f(x) = x^2 - 2x - 3 = 0$$

$x_{\text{Actual}} = -1, 3$

$$g_2(x) = \frac{3}{x-2}$$

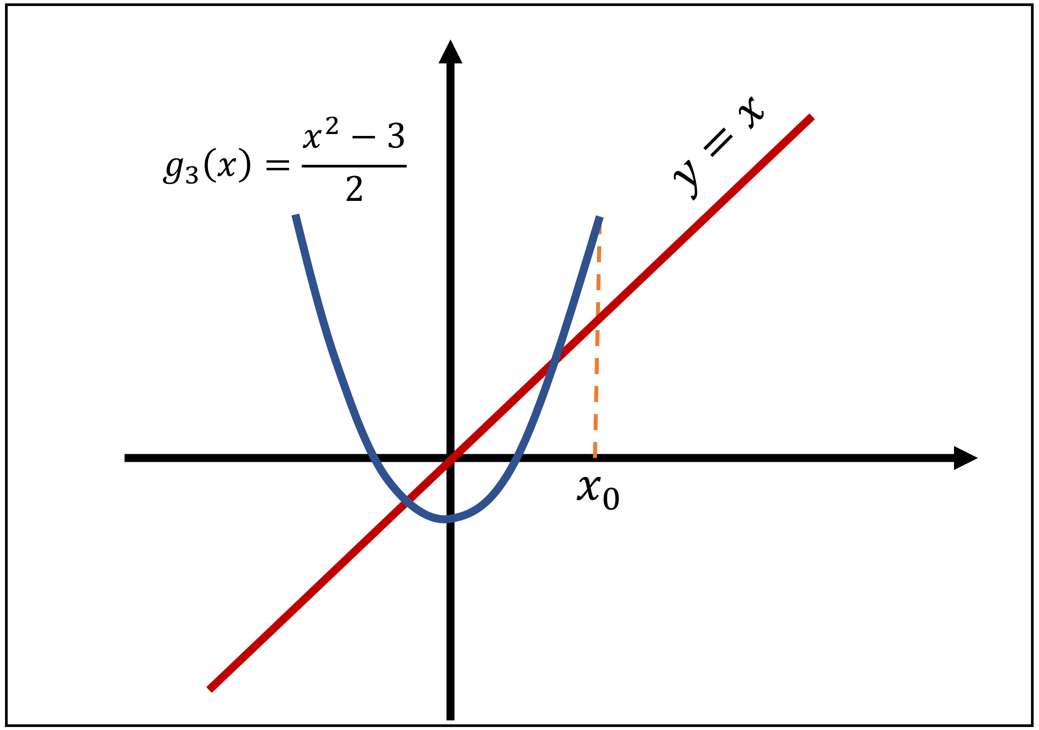
We begin with $x_0 = 4$

`tol` $= 1E - 4$ **OR** (0.0001)

$x_0 = 4,$
$x_1 = 1.5$
$x_2 = -6$
$x_3 = -0.375$
$x_4 = -1.263158$
$x_5 = -0.919355$
$x_6 = -1.02762$
$x_7 = -0.990876$
$x_8 = -1.00305$

Fixed-Point Iteration Method ($x = g(x)$ Method)	
INPUT	• x_1 **reasonably close** to the root • `tol`: the specified tolerance value

```
REARRANGE f(x) INTO x = g(x)
SET x2 = x1
REPEAT
    SET x1 = x2
    SET x2 = g(x1)
    UNTIL (|x1 − x0| <tol)
```



$$f(x) = x^2 - 2x - 3 = 0$$

$x_{\text{Actual}} = -1, 3$

$$g_3(x) = \frac{x^2 - 3}{2}$$

We begin with $x_0 = 4$

$\mathtt{tol} = 1E - 4$ **OR** (0.0001)

$x_0 = 4,$
$x_1 = 6.5$
$x_2 = 19.625$
$x_3 = 191.070$

Diverges

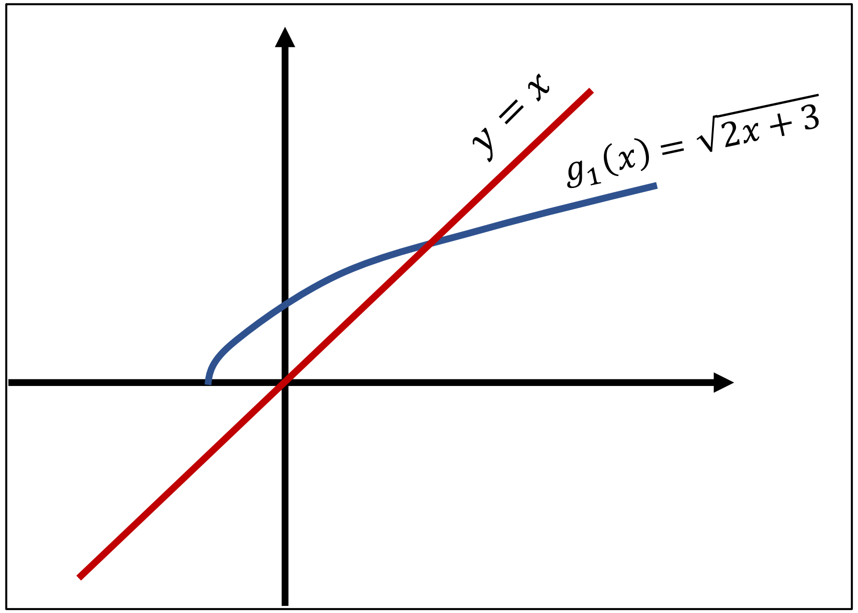
Fixed-Point Iteration Method ($x = g(x)$ Method)	
INPUT	• x_1 **reasonably close** to the root • `tol`: the specified tolerance value

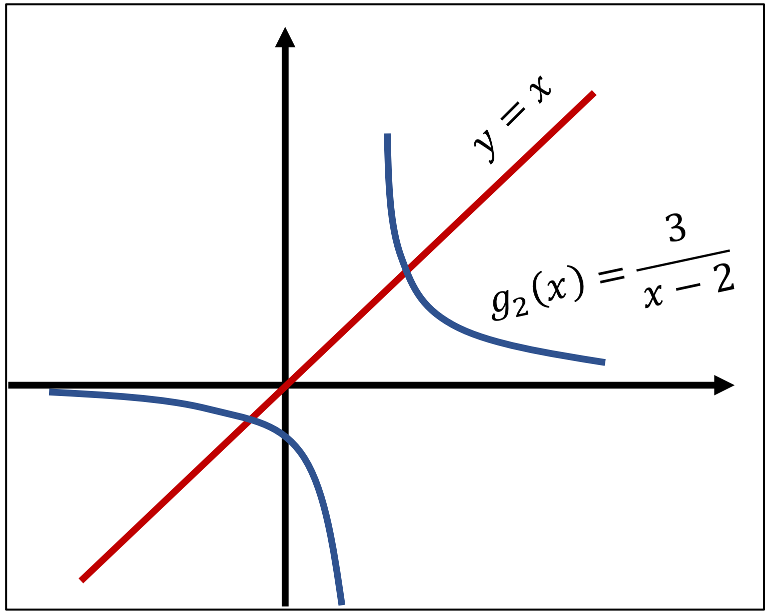
```
REARRANGE f(x) INTO x = g(x)
SET x_2 = x_1
REPEAT
    SET x_1 = x_2
    SET x_2 = g(x_1)
    UNTIL (|x_1 − x_0| <tol)
```

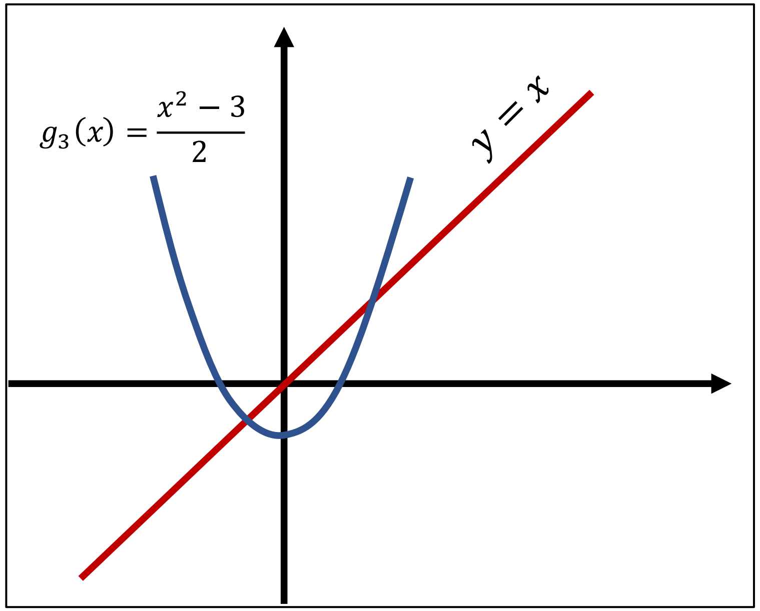
How it works?

Start on the x-axis at the initial x_0, go vertically to the curve $g(x)$, then horizontally to the line $y = x$, then vertically to the curve, and again horizontally to the line.

*Repeat this process until the points on the curve **converge** to a fixed point or else **diverge***



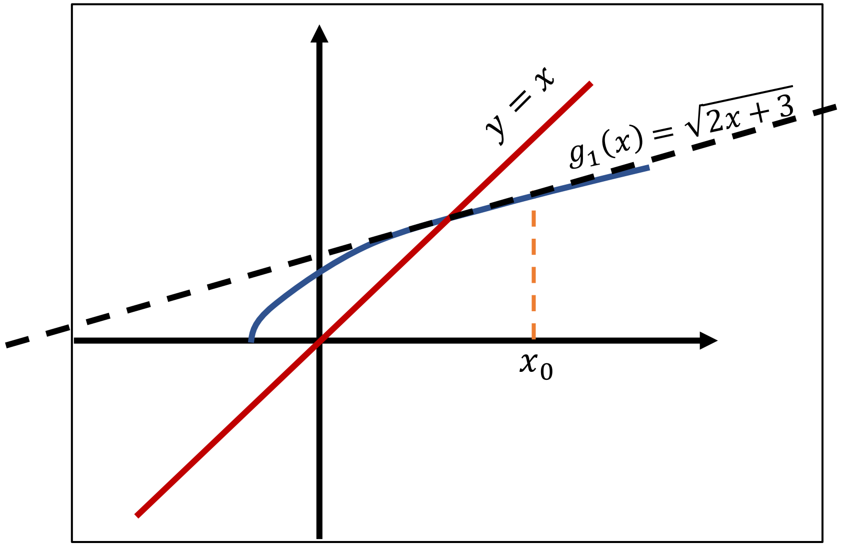




How it works?

The different behaviors depend on whether the **slope of the curve** is greater, less, or of opposite sign to the slope of the line

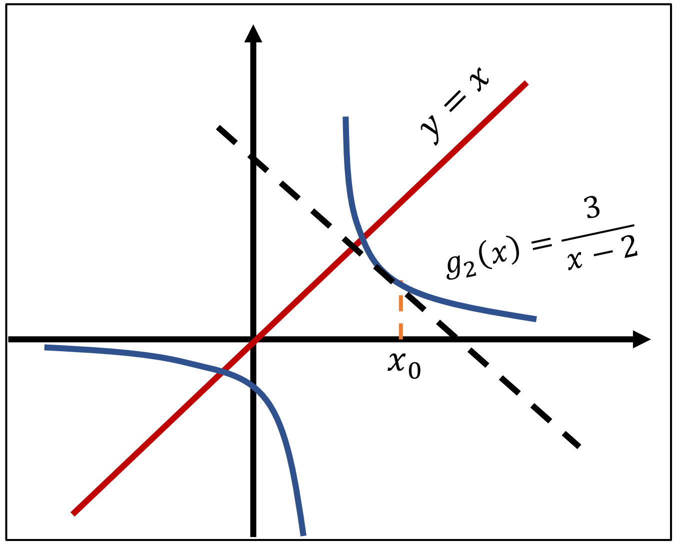
0 < Slope of Curve < 1



Converges

monotonic convergence

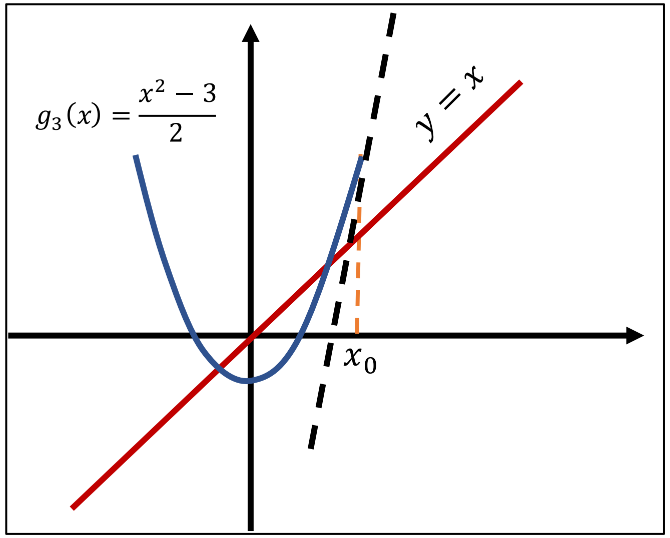
-1 < Slope of Curve < 0



Converges

oscillatory convergence

|Slope of Curve| ≥ 1



Diverges

Fixed-Point Iteration - Animation

http://mathfaculty.fullerton.edu/mathews/a2001/Animations/RootFinding/FixedPoint/FixedPoint.html