Tikrit university
College of Engineering
Mechanical Engineering Department

Lectures on Numerical Analysis

Chapter 1
Solving Non-linear Equations

Assistant prof. Dr. Eng. Ibrahim Thamer Nazzal

Solution of Nonlinear Equations

If we have $f\ (x) = 0$ means to find such points that $f\ (x) = 0$.
We call such point **roots of function** $f\ (x)$. A number ***r*** that satisfies an equation is called a root of the equation.

$f(x) = 0$ is called an **algebraic equation** if the corresponding $f\ (x)$ is **a polynomial.**

An example is $7x^2 + x - 8 = 0$.

$f(x) = 0$ is called **transcendental equation** if the $f\ (x)$ contains **trigonometric, or exponential or logarithmic** functions.

Examples of transcendental equations are $\sin x - x = 0$, $\tan x - x = 0$ and $7x^3 + \log(3x - 6) + 3ex \cos x + \tan x = 0$.

There are Several methods to **solve nonlinear equation of form** $f\ (x) = 0$.

1. Direct Methods: direct methods give the exact value of the roots in a finite number of steps. We assume here that there are no round off errors. Direct methods determine all the roots at the same time. Which can be divided into

A) Analytical Methods

Analytical Solutions are available for special equations only.

Analytical solution of : $a x^2 + b x + c = 0$

$$roots = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

No analytical solution is available for : $\boldsymbol{x - e^{-x} = 0}$

b. Graphical Methods

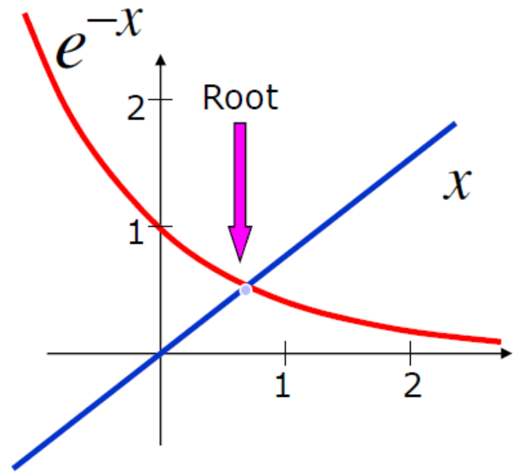
Graphical methods are useful to provide an initial guess to be used by other methods.

$$Solve$$

$$x = e^{-x}$$

$$The \ \ root \in [0,1]$$

$$root \approx 0.6$$



2. Indirect or Iterative Methods: Indirect or iterative methods are based on the concept of successive approximations.

Numerical Solutions

Finding Roots of Equations using numerical solutions.

The indirect or iterative methods are further divided into two categories: bracketing and open methods.

A) The bracketing methods

require the limits between which the root lies, whereas the open methods require the initial estimation of the solution.

Bracketing Methods (Need two initial estimates that will bracket the root. Always converge.)

- Bisection Method
- False-Position Method

B) Open Methods In the open methods, the method starts with **one or more initial guess points**. In each iteration, a new guess of the root is obtained.

- Open methods are usually more efficient than bracketing methods.
  - Fix point iteration
  - Newton-Raphson Method (Needs the derivative of the function.)
  - Secant Method

A) Bracketing Methods

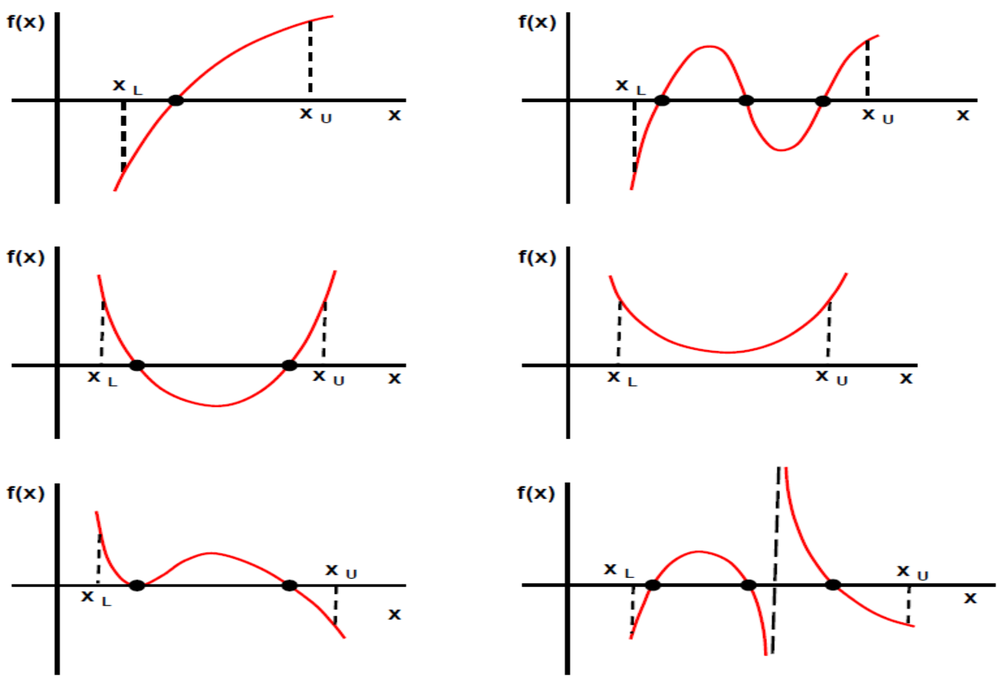
In bracketing methods, the method starts with an **interval** that contains the root and a procedure is used to obtain a **smaller interval containing the root**.

Examples of bracketing methods

General Idea of Bracketing Methods

Bisection method

False position method



Rule 1: If $f(x_L)*f(x_U) < 0$ than there are
odd number of roots

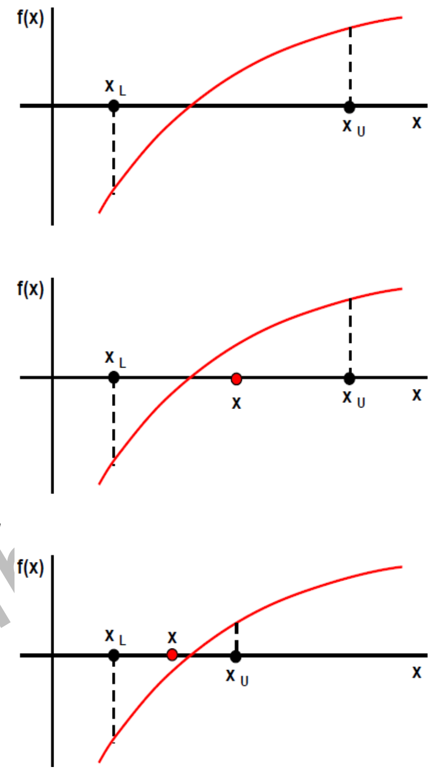
Rule 2: If $f(x_L)*f(x_U) > 0$ than ther are
i) even number of roots
ii) no roots

Violations:
i) multiple roots
ii) discontinuities

Bisection method

The **bisection method** is one of the bracketing methods for finding roots of an equation. It is also called **interval halving** method.

- We start off with two points x_L and x_U, chosen to lie on opposite sides of the solution. Hence $f(x_L)$ and $f(x_U)$ have opposite signs. If the interval (x_L, x_U) is small enough, it is likely to contain a single root.
- i.e., an interval $[x_L, x_U]$ must contain a zero of a continuous function f. Geometrically, this means that if $f(x_L)\, f(x_U) < 0$, then the curve f has to cross the x-axis at some point in between x_L and x_U.
- We then bisect this interval, so take $x_3 = 0.5(x_L + x_U)$, and evaluate $f(x_3)$.
- For the next iteration we retain x_3 and whichever of x_L or x_U gave the opposite sign of f to $f(x_3)$. The solution thus lies between the two points we retain.
- We continue bisecting the interval as above until it becomes sufficiently small, ie. $|x_n - x_{n-1}| < \varepsilon$ for some small convergence criteria.



- Start with two initial guesses, x_{LOWER} and x_{UPPER} .
- They should bracket the root, i.e. $f(x_L) * f(x_U) < 0$
- Estimate the root as the midpoint of this interval. $x = (x_L + x_U)/2$
- Determine the interval which contains the root

  if $f(x_L) * f(x) < 0$ root is between x_L and x

  else root is between x and x_U
- Estimate a new root in this new interval
- Iterate like this and compute the error after each iteration.
- Stop when the specified tolerance is reached.

Example Solve $x^3 - 9x + 1 = 0$ for the root between $x = 2$ and $x = 4$, by bisection method.

Solution

Given $f(x) = x^3 - 9x + 1$. Now $f(2) = -9$, $f(4) = 29$ so that $f(2)\,f(4) < 0$ and hence a root lies between 2 and 4.

Set $a_0 = 2$ and $b_0 = 4$. Then

$$x_0 = \frac{a_0 + b_0}{2} = \frac{2+4}{2} = 3 \quad \text{and} \quad f(x_0) = f(3) = 1$$

Since $f(2)\,f(3) < 0$, a root lies between 2 and 3, hence we set $a_1 = a_0 = 2$ and $b_1 = x_0 = 3$. Then

$$x_1 = \frac{a_1 + b_1}{2} = \frac{2+3}{2} = 2.5 \quad \text{and} \quad f(x_1) = f(2.5) = -5.875$$

Since $f(2)\,f(2.5) > 0$, a root lies between 2.5 and 3, hence we set $a_2 = x_1 = 2.5$ and $b_2 = b_1 = 3$.

Then

$$x_2 = \frac{a_2 + b_2}{2} = \frac{2.5 + 3}{2} = 2.75 \quad \text{and} \quad f(x_2) = f(2.75) = -2.9531$$

The steps are illustrated in the following table.

n	x_n	$f(x_n)$
0	3	1.0000
1	2.5	−5.875
2	2.75	−2.9531
3	2.875	−1.1113
4	2.9375	−0.0901

Example: Find a real root of the equation $f(x) = x^3 - x - 1 = 0$ by bisection method

Solution

Take a = 1 and b= 2

Since $f(1)$ is negative and $f(2)$ positive, a root lies between 1 and 2 and therefore we take x_0 = 3/2= 1.5. Then

$f(x_0) = \frac{27}{8} - \frac{3}{2}$ -1 = $\frac{15}{8}$ is positive and hence $f(1)\, f(1.5) < 0$ and Hence the root lies between 1 and 1.5 and we obtain

$$x_1 = \frac{1+1.5}{2} = 1.25$$

$f(1.25) = -19/64$, which is negative and hence $f(1)\, f(1.25) > 0$ and hence a root lies between 1.25 and 1.5. Also,

$$x_2 = \frac{1.25+1.5}{2} = 1.375$$

The procedure is repeated and the successive approximations are

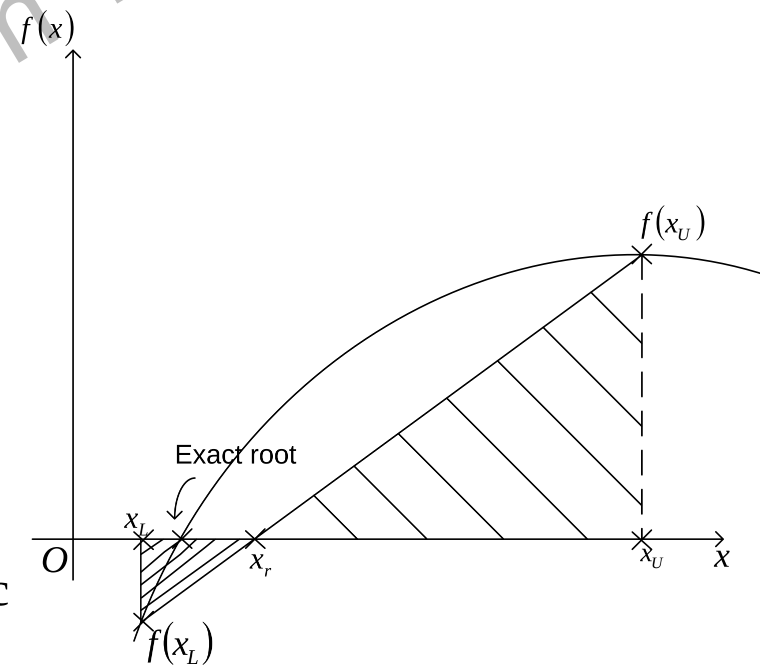
$x_3 = 1.3125$, $x_4 = 1.34375$, $x_5 = 1.328125$, etc.

False-Position Method of Solving a Nonlinear Equation

The **false-position method** is a modification on the bisection **method**. In this method, we choose two points a and b such that $f\,(a)$ and $f\,(b)$ are of opposite signs. We can write this condition as $f(a) \times f(b) < 0$.

The idea for the False position method is to connect the points $(a,f(a))$ and $(b,f(b))$ with a straight line and identify a "false" position x, which of course may not be a true solution. Thus, to find a real root of $f\,(x)$, we replace the part of the curve between the points by a chord in that interval and we take the point of intersection of this chord with the x-axis as a first approximation to the root. We can keep repeating this procedure to get approximations of the solution, x2, x3, ….

Based on two similar triangles, shown in Figure 1, one gets



$$\frac{0 - f(x_L)}{x_r - x_L} = \frac{0 - f(x_U)}{x_r - x_U}$$

From above Equation, one obtains

$$(x_r - x_L) f(x_U) = (x_r - x_U) f(x_L)$$

$$x_U f(x_L) - x_L f(x_U) = x_r \{f(x_L) - f(x_U)\}$$

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

The above equation, through simple algebraic manipulations, can also be expressed as

$$x_r = x_U - \frac{f(x_U)}{\left\{\frac{f(x_L) - f(x_U)}{x_L - x_U}\right\}}$$

or

$$x_r = x_L - \frac{f(x_L)}{\left\{\frac{f(x_U) - f(x_L)}{x_U - x_L}\right\}}$$

Advantage: Convergence is faster than bisection method.

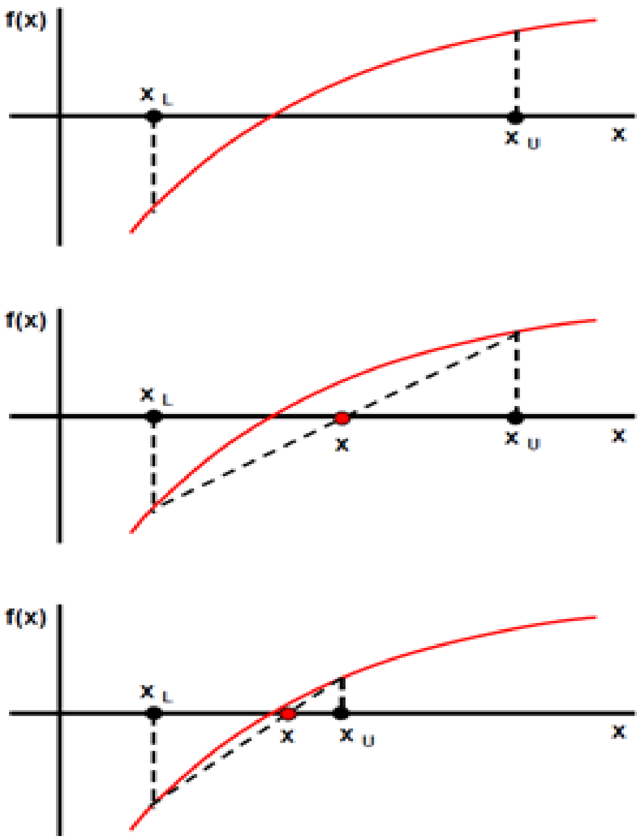
Disadvantages:

1. It requires a and b.

2. The convergence is generally slow.

3. It is only applicable to $f(x)$ of certain fixed curvature in $[a, b]$.

4. It cannot handle multiple zeros.



- Start with two initial guesses, x_{LOWER} and x_{UPPER} .
- They should bracket the root, i.e. $f(x_L) * f(x_U) < 0$

$$\text{let } U = a \text{ and } L = b$$

- Estimate the root using similar triangles.

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

- Determine the interval which contains the root

  if $f(x_L) * f(x) < 0$ root is between x_L and x

  else root is between x and x_U

- Estimate a new root in this new interval
- Iterate like this and compute the error after each iteration.
- Stop when the specified tolerance is reached.

Example : Find a root of an equation $f(x) = x^3 - x - 1$ using False Position method

Solution:

Take $a = 1$, $b = 2$.

Hence the root lies in between 1 and 2.

$f(1) = -1$ and $f(2) = 5$

Hence the root lies in between 1 and 2.

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

$$x_1 = \frac{2\,(-1) - 1(5)}{-1 - 5} = 1.16667$$

$f(x_r) = f(1.16667) = -0.5787 < 0.$

Therefore the root lies between 1.16667 and 2. Again, using the formula, we get the second approximation as,

$$x_2 = \frac{2(-0.5787) - 1.16667(5)}{(-0.5787) - 5} = 1.25311$$

Proceeding like this, we get the next approximation as,

$x_3 = 1.29344$, $x_4 = 1.31128$, $x_5 = 1.31899$,

$x_6 = 1.32228$, $x_7 = 1.32368$

Example :You are working for "DOWN THE TOILET COMPANY" that makes floats for ABC commodes. The floating ball has a specific gravity of 0.6 and has a radius of 5.5cm. You are asked to find the depth to which the ball is submerged when floating in water. The equation that gives the depth *x* to which the ball is submerged under water is given by

$$x^3 - 0.165x^2 + 3.993\times10^{-4} = 0$$

Use the false-position method of finding roots of equations to find the depth *x* to which the ball is submerged under water. Conduct three iterations to estimate the root of the above equation. Find the absolute relative approximate error at the end of each iteration, and the number of significant digits at least correct at the end of third iteration.

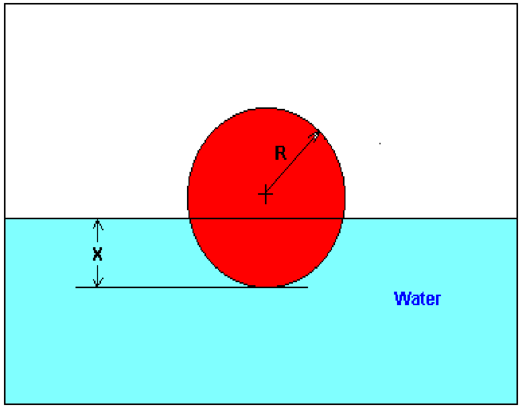
Solution

From the physics of the problem, the ball would be submerged between $x = 0$ *and* $x = 2R$ Where R = radius of the ball that is

$$0 \le x \le 2R$$

$$0 \le x \le 2(0.055)$$

$$0 \le x \le 0.11$$



Floating ball problem.

Let us assume $x_L = 0,\ x_U = 0.11$

Check if the function changes sign between $x_L = 0$, $x_U = 0.11$

$$f(x_L) = f(0) = (0)^3 - 0.165(0)^2 + 3.993\times10^{-4} = 3.993\times10^{-4}$$

$$f(x_U) = f(0.11) = (0.11)^3 - 0.165(0.11)^2 + 3.993\times10^{-4} = -2.662\times10^{-4}$$

Hence

$$f(x_L)f(x_U) = f(0)f(0.11) = (3.993\times10^{-4})(-2.662\times10^{-4}) < 0$$

Therefore, there is at least one root between x_L and x_U that is between *0* and *0.11*.

Iteration 1

The estimate of the root is

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

$$= \frac{0.11\times3.993\times10^{-4} - 0\times(-2.662\times10^{-4})}{3.993\times10^{-4} - (-2.662\times10^{-4})}$$

$$= 0.0660$$

$$f(x_r) = f(0.0660)$$

$$= (0.0660)^3 - 0.165(0.0660)^2 + (3.993\times10^{-4})$$

$$= -3.1944\times10^{-5}$$

$$f(x_L)f(x_r) = f(0)f(0.0660) = (+)(-) < 0$$

Hence, the root is bracketed between x_L and x_r that is, between 0 and 0.0660. So, the lower and upper limits of the new bracket are $x_L = 0$ and $x_U = 0.066$ respectively.

Iteration 2

The estimate of the root is

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

$$= \frac{0.0660\times3.993\times10^{-4} - 0\times(-3.1944\times10^{-5})}{3.993\times10^{-4} - (-3.1944\times10^{-5})}$$

$$= 0.0611$$

The absolute relative approximate error for this iteration is

$$\in_a = \left| \frac{0.0611 - 0.0660}{0.0611} \right| \times 100 \cong 8\%$$

$$f(x_r) = f(0.0611)$$
$$= (0.0611)^3 - 0.165(0.0611)^2 + (3.993\times10^{-4})$$
$$= 1.1320\times10^{-5}$$

$$f(x_L)f(x_r) = f(0)f(0.0611) = (+)(+) > 0$$

Hence, the lower and upper limits of the new bracket are $x_L = 0.0611$ and $x_U = 0.066$ respectively.

Table 1 Root of $f(x) = x^3 - 0.165x^2 + 3.993\times10^{-4} = 0$ for false-position method.

Iteration	x_L	x_U	x_r	$\lvert\in_a\rvert\%$	$f(x_m)$
1	0.0000	0.1100	0.0660	----	-3.1944×10^{-5}
2	0.0000	0.0660	0.0611	8.00	-1.1320×10^{-5}
3	0.0611	0.0660	0.0624	2.05	-1.1313×10^{-7}

Simple Fixed-point Iteration

We have seen the non-linear equation is *f(x) =0*. The idea of the fixed point iteration methods is to first reformulate a equation to an equivalent fixed point problem:

$$f(x) = 0 \iff x = g(x)$$

and then to use the iteration: with an initial guess x_0 chosen, compute a sequence

The utility of above the equation is that it provides a formula to predict a new value of x as a function of an old value of x. Thus, given an initial guess at the root x_i, above equation can be used to compute a new estimate x_{i+1} as expressed by the iterative formula

$$x_{i+1} = g(x_i)$$

There are infinite many ways to introduce an equivalent fixed point problem for a given equation.

Now in this method, the process is

- **To solve $f(x) = 0$.**
- **Rearrange $f(x)$ in such a way that $x = g(x)$**

→ ***Start with an initial guess for x* say x_i**

→**Evaluate $g(x_i)$**

→**If not equal then, $x_{i+1} = g(x_i)$**

→**Evaluate $g(x_{i+1})$**

→**Continue till some tolerance ε i.e., $|x_{i+1} - x_i| \le \varepsilon$**

It is called 'fixed point iteration' because the root α of the equation $x - g(x) = 0$ is a fixed point of the function $g(x)$, meaning that α is a number for which g(α) = α

As with many other iterative formulas in this book, the approximate error for this equation can be determined using the error estimator:

$$e_a = \left| \frac{x_1 - x_0)}{x_1} \right| \times 100$$

***Example:-* Find the root of equation $2x = \cos x + 3$ and correct to three decimal places.**

solution: we rewrite the equation in the form

$$x = \frac{1}{2}(cosx + 3)$$

So that $g(x) = \frac{1}{2}(cosx + 3)$

Hence the iteration method can be applied to the above eq. and we start with $x_0 = \Pi / 2$. The successive iterates are

$$x_1 = 1.5, \quad x_2 = 1.535, \quad x_3 = 1.518,$$
$$x_4 = 1.526, \quad x_5 = 1.522, \quad x_6 = 1.524,$$
$$x_7 = 1.523, \quad x_8 = 1.524.$$

We accept the solution as 1.524 correct to three decimal places.

Example Solve $f(x) = x^2 - 3x + 1 = 0$, by fixed-point iteration method.

Solution

Write the given equation as

$x^2 = 3x - 1$ or $x = 3 - 1/x$.

Choose $g(x) = 3 - 1/x$

The iterative formula is given by

$$x_{i+1} = 3 - \frac{1}{x_i} \quad (i = 1, 2. 3, \ldots\ldots)$$

Starting with, $x_0 = 1$, *we obtain the sequence*

$x_0 = 1.000$, $x_1 = 2.000$, $x_2 = 2.500$, $x_3 = 2.600$, $x_4 = 2.615$, . . .

Example Use simple fixed-point iteration to locate the root of $f(x) = e^{-x} - x$.

Solution. The function can be separated directly and expressed in the form of eqaution as
$x_{i+1} = e^{-x}$
Starting with an initial guess of $x_0 = 0$, this iterative equation can be applied to compute:

i	x_i	$\lvert \varepsilon_a \rvert$, %	$\lvert \varepsilon_t \rvert$, %	$\lvert \varepsilon_t \rvert_i / \lvert \varepsilon_t \rvert_{i-1}$
0	0.0000		100.000	
1	1.0000	100.000	76.322	0.763
2	0.3679	171.828	35.135	0.460
3	0.6922	46.854	22.050	0.628
4	0.5005	38.309	11.755	0.533
5	0.6062	17.447	6.894	0.586
6	0.5454	11.157	3.835	0.556
7	0.5796	5.903	2.199	0.573
8	0.5601	3.481	1.239	0.564
9	0.5711	1.931	0.705	0.569
10	0.5649	1.109	0.399	0.566

Thus, each iteration brings the estimate closer to the true value of the root: **0.56714329**.

Newton-Raphson Method of Solving a Nonlinear Equation

The Newton-Raphson method, or Newton Method, is a powerful technique for solving equations numerically. In the Newton-Raphson method, the root is not bracketed. In fact, only one initial guess of the root is needed to get the iterative process started to find the root of an equation. The method hence falls in the category of open methods. Convergence in open methods is not guaranteed but if the method does converge, it does so much faster than the bracketing methods. Consider $f(x) = 0$, where f has continuous derivative f'. From figure we can see that, the tangent to the curve f at $(x_i, f(x_i))$ (with slope $f'(x_i)$) touches the x-axis at x_{i+1}.

Graphical derivation

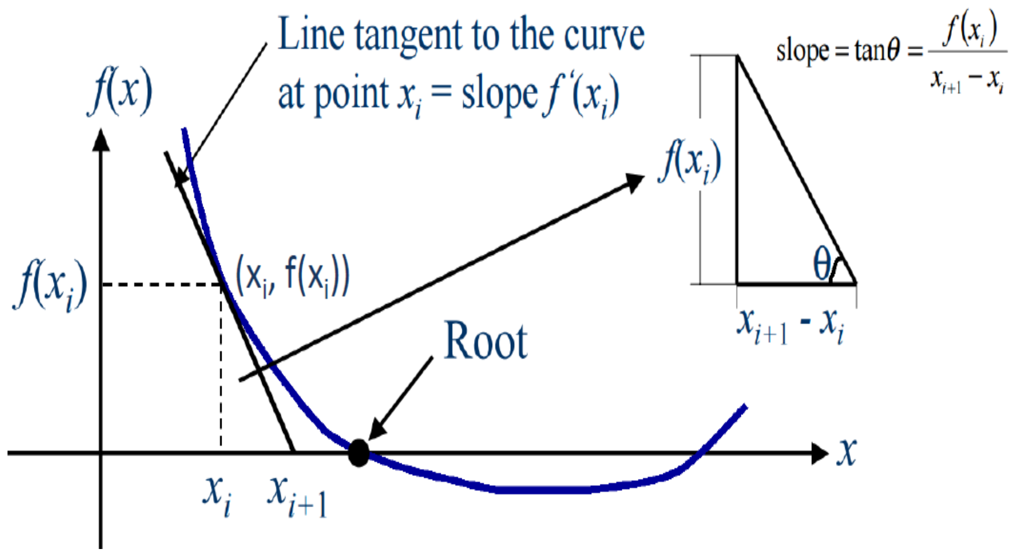
The Newton-Raphson method is based on the principle that if the initial guess of the root of $f(x)=0$ is at x_i then if one draws the tangent to the curve at $f(x_i)$ the point x_{i+1} where the tangent crosses the x axis is an improved estimate of the root (Figure 1). Using the definition of the slope of a function, a $x = x_i$

$$f'(x_i) = \tan\theta$$

$$= \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

which gives

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



More generally, we can be written

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

The main steps of the Newton Raphson

1. Evaluate $f\,(\acute{x})$
2. Choose an initial guess, x_0
3. Use an initial guess of the root, x_i, to estimate the new value of the root, x_{i+1} , as

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

4. Find the absolute relative approximate error

$$\epsilon_a \left| x_{i+1} = \frac{x_1 - x_0)}{x_1} \right| \times 100$$

Compare the absolute relative approximate error with the pre-specified relative error tolerance,

If $\epsilon_a < \epsilon_s$ ϵ_s some pre-specified value

$x_0 = x_1$, Go to 3

Example

Use the Newton-Raphson iteration method to estimate the root of the following function employing an initial guess of $x_0 = 0$: $f(x) = e^{-x} - x$

Let's find the derivative of the function first, $f'(x) = \frac{df(x)}{dx} = -e^{-x} - 1$

The initial guess is $x_0 = 0$, hence,

$\underline{i = 0:}$

$f(0) = e^{-(0)} - 0 = 1$ $\qquad f(x) = e^{-x} - x$

$f'(0) = -e^{-(0)} - 1 = -1-1=-2$ $\qquad f'(x) = \frac{df(x)}{dx} = -e^{-x} - 1$

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

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0 - \frac{1}{-2} = 0.5$$

Now $x_1 = 0.5$, hence,

$\underline{i = 1}$

$f(0.5) = e^{-(0.5)} - (0.5) = 0.1065$ $\qquad f(x) = e^{-x} - x$

$= -e^{-(0.5)} - 1 = -1.6065$ $\qquad f'(x) = \frac{df(x)}{dx} = -e^{-x} - 1$

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

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 0.5 - \frac{0.1065}{-1.6065} = 0.5663$$

Now x_2= 0.5663, hence,

i=2

$$f'(x)=\frac{df(x)}{dx}=-e^{-x}-1$$

$$f(0.5663) = e^{-(0.5663)} - (0.5663) = 0.001322$$

$$= -e^{-(0.5663)} - 1 = -1.567622$$

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

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 0.5663 - \frac{0.001322}{-1.567622} = 0.5671$$

Now x_3= 0.5671, hence,

$$f'(x)=\frac{df(x)}{dx}=-e^{-x}-1$$

$$f(0.5671) = e^{-(0.5671)} - (0.5671) = 0.00006784$$

$$= -e^{-(0.5671)} - 1 = -1.56716784$$

Thus, the approach rapidly converges on the true root of 0.5671 to four significant digits.

i	x_i	$f(x_i)$	$f'(x_i)$	Percent $\lvert\varepsilon_r\rvert$
0	0	1	-2	---
1	0.5	0.106531	-1.6065307	100
2	0.566311003	0.001305	-1.5676155	11.709291
3	0.567143165	1.96E-07	-1.5671434	0.14672871
4	0.56714329	4.44E-15	-1.5671433	2.2106E-05
5	0.56714329	0	-1.5671433	5.0897E-13

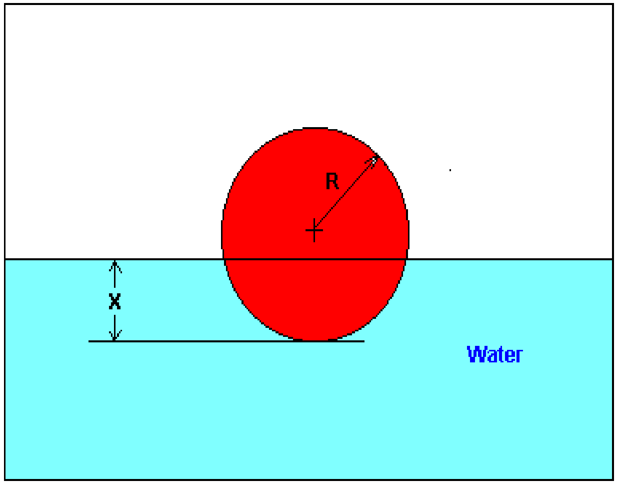
Hence, the root is 0.5671.

Example

You are working for 'DOWN THE TOILET COMPANY' that makes floats for ABC commodes. The floating ball has a specific gravity of 0.6 and has a radius of 5.5 cm. You are asked to find the depth to which the ball is submerged when floating in water.

The equation that gives the depth x in meters to which the ball is submerged under water is given by

$$x^3 - 0.165x^2 + 3.993\times10^{-4} = 0$$



Use the Newton-Raphson method of finding roots of equations to find

a) the depth x to which the ball is submerged under water. Conduct three iterations to estimate the root of the above equation.
b) the absolute relative approximate error at the end of each iteration, and
c) the number of significant digits at least correct at the end of each iteration

Solution

$$f(x) = x^3 - 0.165x^2 + 3.993\times10^{-4} \qquad f'(x) = 3x^2 - 0.33x$$

Let us assume the initial guess of the root of $f(x) = 0$ is $x_0 = 0.05$ m. This is a reasonable guess (discuss why $x = 0$ and $x = 0.11m$) are not good choices) as the extreme values of the depth x would be *0* and the diameter (*0.11* m) of the ball.

Iteration 1

The estimate of the root is

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \qquad = 0.05 - \frac{(0.05)^3 - 0.165(0.05)^2 + 3.993\times10^{-4}}{3(0.05)^2 - 0.33(0.05)}$$

$$= 0.05 - \frac{1.118\times10^{-4}}{-9\times10^{-3}}$$

$$= 0.05 - (-0.01242) \quad = 0.06242$$

The absolute relative approximate error $|\epsilon_a|$ at the end of Iteration 1 is

$$|\in_a| = \left|\frac{x_1 - x_0}{x_1}\right|\times100 \quad = \left|\frac{0.06242 - 0.05}{0.06242}\right|\times100$$

$$= 19.90\%$$

The number of significant digits at least correct is 0, as you need an absolute relative approximate error of 5% or less for at least one significant digit to be correct in your result.

Iteration 2

The estimate of the root is

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

$$= 0.06242 - \frac{(0.06242)^3 - 0.165(0.06242)^2 + 3.993\times10^{-4}}{3(0.06242)^2 - 0.33(0.06242)}$$

$$= 0.06242 - \frac{-3.97781\times10^{-7}}{-8.90973\times10^{-3}}$$

$$= 0.06242 - \left(4.4646\times10^{-5}\right) \quad = 0.06238$$

The absolute relative approximate error $|\epsilon_a|$ at the end of Iteration 2 is

$$|\epsilon_a| = \left|\frac{x_2 - x_1}{x_2}\right| \times 100$$

$$= \left|\frac{0.06238 - 0.06242}{0.06238}\right| \times 100 = 0.0716\%$$

Hence, the number of significant digits at least correct in the answer is 2.

Iteration 3

The estimate of the root is

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$

$$= 0.06238 - \frac{(0.06238)^3 - 0.165(0.06238)^2 + 3.993 \times 10^{-4}}{3(0.06238)^2 - 0.33(0.06238)} = 0.06238$$

The absolute relative approximate error $|\epsilon_a|$ at the end of Iteration 3 is

$$|\epsilon_a| = \left|\frac{0.06238 - 0.06238}{0.06238}\right| \times 100 = 0$$

The number of significant digits at least correct is 4, as only 4 significant digits are carried through in all the calculations.

Example You are designing a spherical tank to hold water for a small village in a developing country. The volume of liquid it can hold can be computed as:

$$V = \pi h^2 \frac{(3R - h)}{3}$$

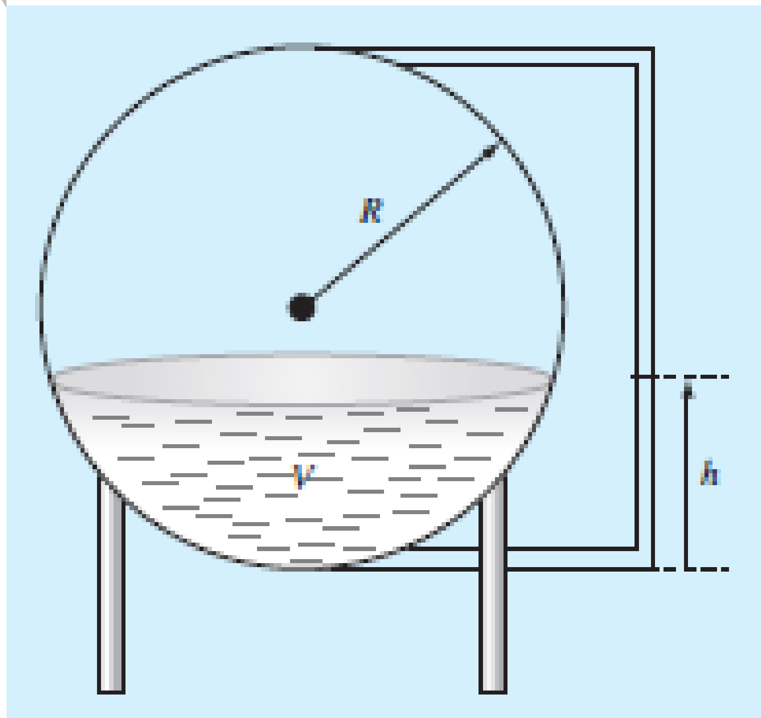
where V= volume [m^3], h = depth of water in tank [m], and R = the tank radius [m]. If R =3 m, what depth must the tank be filled to so that it holds 30 m^3? Use **Newton-Raphson** iterations to determine your answer. Determine the absolute relative approximate error a ε as a percentage after each iteration. For **Newton-Raphson method** to be used, an initial guess of R will always converge.

solution

1. The equation to be solved is

$$f(h) = \pi R h^2 - \left(\frac{\pi}{3}\right) h^3 - V$$

$$h_{i+1} = h_i - \frac{\pi R h_i^2 - \left(\frac{\pi}{3}\right) h_i^3 - V}{2\pi R h_i - \pi h_i^2}$$



Substituting the parameter values,

$$f(h) = \pi R h^2 - \left(\frac{\pi}{3}\right) h^3 - V$$

$$h_{i+1} = h_i - \frac{\pi 3 h_i^2 - \left(\frac{\pi}{3}\right) h_i^3 - 30}{2\pi 3 h_i - \pi h_i^2}$$

$i = 0$; $h_0 = 3$ m:

$$h_1 = h_0 - \frac{\pi 3 h_0^2 - \left(\frac{\pi}{3}\right) h_0^3 - 30}{2\pi 3 h_0 - \pi h_0^2} = 3 - \frac{\pi 3(3)^2 - \left(\frac{\pi}{3}\right)(3)^3 - 30}{2\pi 3(3) - \pi(3)^2} = 2.061033$$

$$|\varepsilon_a| = \left|\frac{h_1 - h_0}{h_1}\right| \times 100 = \left|\frac{2.061033 - 3}{2.061033}\right| \times 100 = 45.558\%$$

$i = 1$; $h_1 = 2.061033$ m:

$$h_2 = h_1 - \frac{\pi 3 h_1^2 - \left(\frac{\pi}{3}\right) h_1^3 - 30}{2\pi 3 h_1 - \pi h_1^2} = 2.061033 - \frac{\pi 3(2.061033)^2 - \left(\frac{\pi}{3}\right)(2.061033)^3 - 30}{2\pi 3(2.061033) - \pi(2.061033)^2} = 2.027042$$

$$|\varepsilon_a| = \left|\frac{h_2 - h_1}{h_2}\right| \times 100 = \left|\frac{2.027042 - 2.061033}{2.027042}\right| \times 100 = 1.667\%$$

$i = 2$: $h_2 = 2.027042$ m:

$$h_3 = h_2 - \frac{\pi 3 h_2^2 - \left(\frac{\pi}{3}\right) h_2^3 - 30}{2\pi 3 h_2 - \pi h_2^2} = 2.027042 - \frac{\pi 3(2.027042)^2 - \left(\frac{\pi}{3}\right)(2.027042)^3 - 30}{2\pi 3(2.027042) - \pi(2.027042)^2} = 2.026906$$

$$|\varepsilon_a| = \left|\frac{h_2 - h_1}{h_2}\right| \times 100 = \left|\frac{2.026906 - 2.027042}{2.026906}\right| \times 100 = 0.007\%$$

Thus, after only three iterations, the root is determined to be 2.026906 with an approximate relative error of 0.007%.

Secant Method of Solving Nonlinear Equations

One of the drawbacks of the Newton-Raphson method is that you have to evaluate the derivative of the function and this is not always possible, particularly in the case of functions arising in practical problems. To overcome these drawbacks, Secant Method is a modified version of Newton's method in which $f'(x_n)$ is approximated by

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

We know that from Newton's method

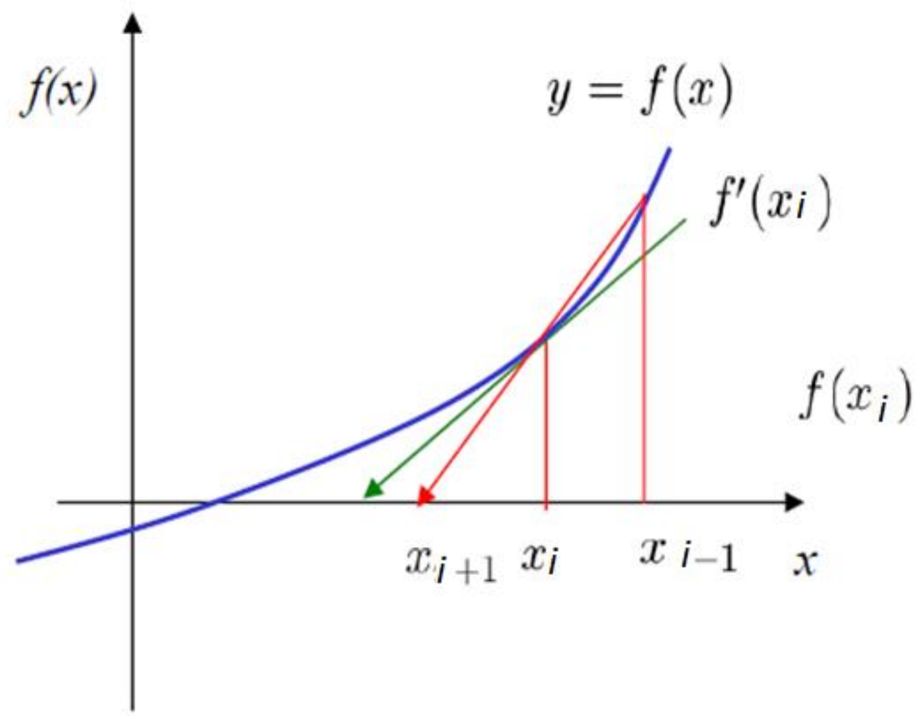
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad (2)$$

Substituting Equation (1) in Equation (2) gives

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

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

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



Only function call per iteration

The above equation is called the secant method

This method now requires two initial guesses

A disadvantage of the secant method is that it may not be convergent to the root (for example, when the initial guess is not near the root). However, since the derivative is approximated as given by Equation (2), it typically converges slower than the Newton-Raphson method.

The secant method is an open method and may or may not converge. However, when secant method converges, it will typically converge faster than the bisection method.

Example find a real root of the equation $x^3 - 2x - 5 = 0$ using secant method.
Let the two initial approximations be given by $x_{-1} = 2$ and $x_0 = 3$.

solution

$f(x_{-1}) = f_{-1} = 8 - 9 = -1$, and $f(x_0) = f_0 = 27 - 11 = 16$.

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

$$x_1 = \frac{2(16) - 3(-1)}{17} = \frac{35}{17} = 2.058823529$$

Also

$$f(x_1) = f_1 = -0.390799923$$

$$x_2 = \frac{f_1\,x_0 - f_0\,x_1}{f_1 - f_0} = \frac{3\,(-0.390799923) - 2.058823529\ 16}{-16.390799923} = 2.08126366$$

again

$$f(x_2) = f_2 = -0.147204057$$

$$x_3 = 2.094824145$$

Example

You are working for 'DOWN THE TOILET COMPANY' that makes floats (Figure 2) for ABC commodes. The floating ball has a specific gravity of 0.6 and a radius of 5.5 cm. You are asked to find the depth to which the ball is submerged when floating in water. The equation that gives the depth to which the ball is submerged under water is given by

$$x^3 - 0.165x^2 + 3.993\times10^{-4} = 0$$

Use the secant method of finding roots of equations to find the depth to which the ball is submerged under water. Conduct three iterations to estimate the root of the above equation. Find the absolute relative approximate error and the number of significant digits at least correct at the end of each iteration.

Solution

$$f(x) = x^3 - 0.165x^2 + 3.993\times10^{-4}$$

Let us assume the initial guesses of the root of $f(x) = 0$ *as* $x_{-1} = 0.02$ and $x_0 = 0.05$

Iteration 1

The estimate of the root is

$$x_1 = x_0 - \frac{f(x_0)(x_0 - x_{-1})}{f(x_0) - f(x_{-1})}$$

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

$$= x_0 - \frac{\left(x_0^3 - 0.165x_0^2 + 3.993\times10^{-4}\right)\times\left(x_0 - x_{-1}\right)}{\left(x_0^3 - 0.165x_0^2 + 3.993\times10^{-4}\right) - \left(x_{-1}^3 - 0.165x_{-1}^2 + 3.993\times10^{-4}\right)}$$

$$= 0.05 - \frac{\left[0.05^3 - 0.165(0.05)^2 + 3.993\times10^{-4}\right]\times\left[0.05 - 0.02\right]}{\left[0.05^3 - 0.165(0.05)^2 + 3.993\times10^{-4}\right] - \left[0.02^3 - 0.165(0.02)^2 + 3.993\times10^{-4}\right]}$$

$$= 0.06461$$

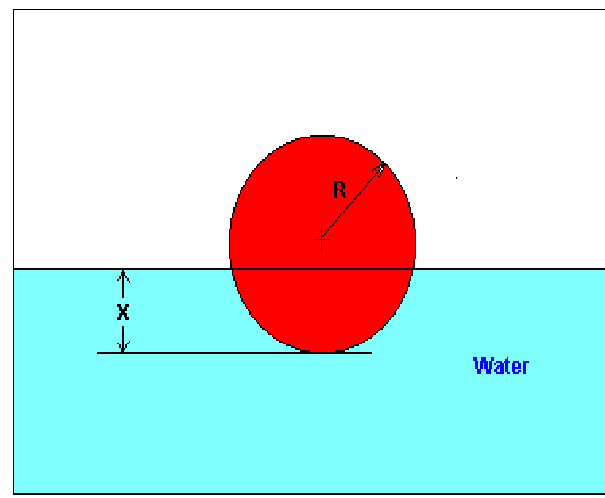


Figure 2 Floating ball problem

The absolute relative approximate error $\left|\in_a\right|$ at the end of Iteration 1 is

$$\left|\in_a\right| = \left|\frac{x_1 - x_0}{x_1}\right| \times 100 \qquad = \left|\frac{0.06461 - 0.05}{0.06461}\right| \times 100 \qquad = 22.62\%$$

Iteration 2

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

$$= x_1 - \frac{\left(x_1^3 - 0.165x_1^2 + 3.993\times10^{-4}\right)\times\left(x_1 - x_0\right)}{\left(x_1^3 - 0.165x_1^2 + 3.993\times10^{-4}\right) - \left(x_0^3 - 0.165x_0^2 + 3.993\times10^{-4}\right)}$$

$$= 0.06461 - \frac{\left[0.06461^3 - 0.165(0.06461)^2 + 3.993\times10^{-4}\right]\times(0.06461 - 0.05)}{\left[0.06461^3 - 0.165(0.06461)^2 + 3.993\times10^{-4}\right] - \left[0.05^3 - 0.165(0.05)^2 + 3.993\times10^{-4}\right]} \qquad = 0.06241$$

The absolute relative approximate error $\left|\in_a\right|$ at the end of Iteration 2 is

$$\left|\in_a\right| = \left|\frac{x_2 - x_1}{x_2}\right| \times 100 \qquad = \left|\frac{0.06241 - 0.06461}{0.06241}\right| \times 100 \qquad = 3.525\%$$

The number of significant digits at least correct is 1, as you need an absolute relative approximate error of 5% or less.

Iteration 3

$$x_3 = x_2 - \frac{f(x_2)(x_2 - x_1)}{f(x_2) - f(x_1)}$$

$$= x_2 - \frac{\left(x_2^3 - 0.165x_2^2 + 3.993\times10^{-4}\right)\times\left(x_2 - x_1\right)}{\left(x_2^3 - 0.165x_2^2 + 3.993\times10^{-4}\right) - \left(x_1^3 - 0.165x_1^2 + 3.993\times10^{-4}\right)}$$

$$= 0.06241 - \frac{\left[0.06241^3 - 0.165(0.06241)^2 + 3.993\times10^{-4}\right]\times(0.06241 - 0.06461)}{\left[0.06241^3 - 0.165(0.06241)^2 + 3.993\times10^{-4}\right] - \left[0.06461^3 - 0.165(0.06461)^2 + 3.993\times10^{-4}\right]} = 0.06238$$

The absolute relative approximate error $\left|\in_a\right|$ at the end of Iteration 3 is

$$\left|\in_a\right| = \left|\frac{x_3 - x_2}{x_3}\right|\times100 = \left|\frac{0.06238 - 0.06241}{0.06238}\right|\times100 = 0.0595\%$$

Table Secant method results as a function of iterations.

Iteration Number, i	x_{i-1}	x_i	x_{i+1}	$\left\|\in_a\right\|\%$	$f(x_{i+1})$
1	0.02	0.05	0.06461	22.62	-1.9812×10^{-5}
2	0.05	0.06461	0.06241	3.525	-3.2852×10^{-7}
3	0.06461	0.06241	0.06238	0.0595	2.0252×10^{-9}
4	0.06241	0.06238	0.06238	-3.64×10^{-4}	-1.8576×10^{-13}