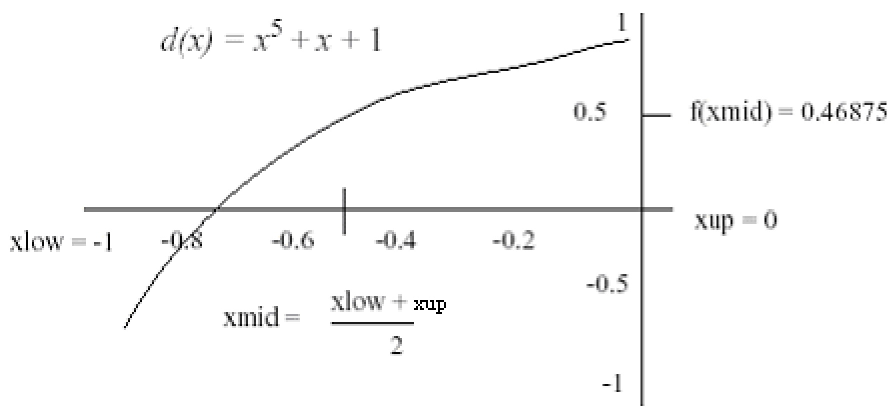
Brac University
Department of Electrical and Electronic Engineering
EEE282/ECE282 (V3)
Numerical Techniques
Experiment-08: Solution to Non-Linear Equations

Bisection method:

The Bisection method is one of the simplest procedures for finding the root of a function in a given interval.

The procedure is straightforward. The approximate location of the root is first determined by finding two values that bracket the root (a root is bracketed or enclosed if the function changes sign at the endpoints). Based on these a third value is calculated which is closer to the root than the original two values. A check is made to see if the new value is a root. Otherwise a new pair of brackets is generated from the three values, and the procedure is repeated.



Consider a function $d(x)$ and let there be two values of x, x_{low} and x_{up} ($x_{up} > x_{low}$), bracketing a root of $d(x)$.

Steps:

1. The first step is to use the brackets x_{low} and x_{up} to generate a third value that is closer to the root. This new point is calculated as the mid-point between x_{low} and

namely $x_{mid} = \frac{x_{low} + x_{up}}{2}$. The method therefore gets its name from this bisecting of two values. It is also known as interval halving method.

2. Test whether x_{mid} is a root of $d(x)$ by evaluating the function at x_{mid}.
3. If x_{mid} is not a root,
   a. If $d(x_{low})$ and $d(x_{mid})$ have opposite signs i.e. $d(x_{low})\,.\,d(x_{mid})$

<0, root is in left half of interval.

b. If $d\,(x_{low})$ and $d\,(x_{mid})$ have same signs i.e. $d\,(x_{low})\,.\,d\,(x_{mid})$ >0, root is in right half of interval.

4. Continue subdividing until interval width has been reduced to a size ≤ ε where ε = selected x tolerance

Algorithm: Bisection Method

```
Input xLower, xUpper, xTol
yLower = f(xLower) (* invokes fcn definition *)
xMid = (xLower + xUpper)/2.0
yMid = f(xMid)
iters = 0 (* count number of iterations *)
While ( (xUpper - xLower)/2.0 > xTol )
iters = iters + 1
if( yLower * yMid > 0.0) Then xLower = xMid
Else xUpper = xMid
Endofif
xMid = (xLower + xUpper)/2.0
yMid = f(xMid)
Endofwhile
Return xMid, yMid, iters (* xMid = approx to root *)
```

Exercise 1. Find the real root of the equation $d(x) = x^5 + x + 1$ using Bisection Method. x_{low} = -1, x_{up} =0 and ε = selected x tolerance $=10^{-4}$.

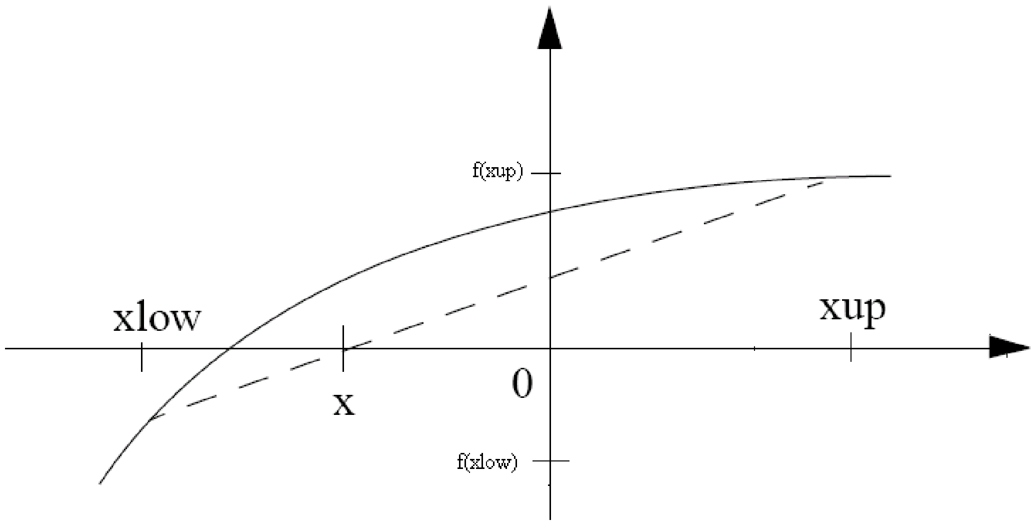
Note: For a given x tolerance (epsilon), we can calculate the number of iterations directly. The number of divisions of the original interval is the smallest value of n that satisfies: $\frac{x_{up} - x_{low}}{2^n} \langle \varepsilon$ or $2^n \rangle \frac{x_{up} - x_{low}}{\varepsilon}$

Thus $n \rangle \log 2\left(\frac{x_{up} - x_{low}}{\varepsilon}\right)$.

In our previous example, x_{low} = -1, x_{up} =0 and ε = selected x tolerance $=10^{-4}$.
So we have $n = 14$.

False-Position Method (Regula Falsi)

A shortcoming of the bisection method is that, in dividing the interval from x_{low} to x_{up} into equal halves, no account is taken of the magnitude of $f(x_{low})$ and $f(x_{up})$. For example, if $f(x_{low})$ is much closer to zero than $f(x_{up})$, it is likely that the root is closer to x_{low} than to x_{up}. An alternative method that exploits this graphical insight is to join $f(x_{low})$ and $f(x_{up})$ by a straight line. The intersection of this line with the x axis represents an improved estimate of the root. The fact that the replacement of the curve by a straight line gives the false position of the root is the origin of the name, method of false position, or in Latin, Regula Falsi. It is also called the Linear Interpolation Method.



Using similar triangles, the intersection of the straight line with the x axis can be estimated as

$$\frac{f(x_{low})}{x - x_{low}} = \frac{f(x_{up})}{x - x_{up}}$$

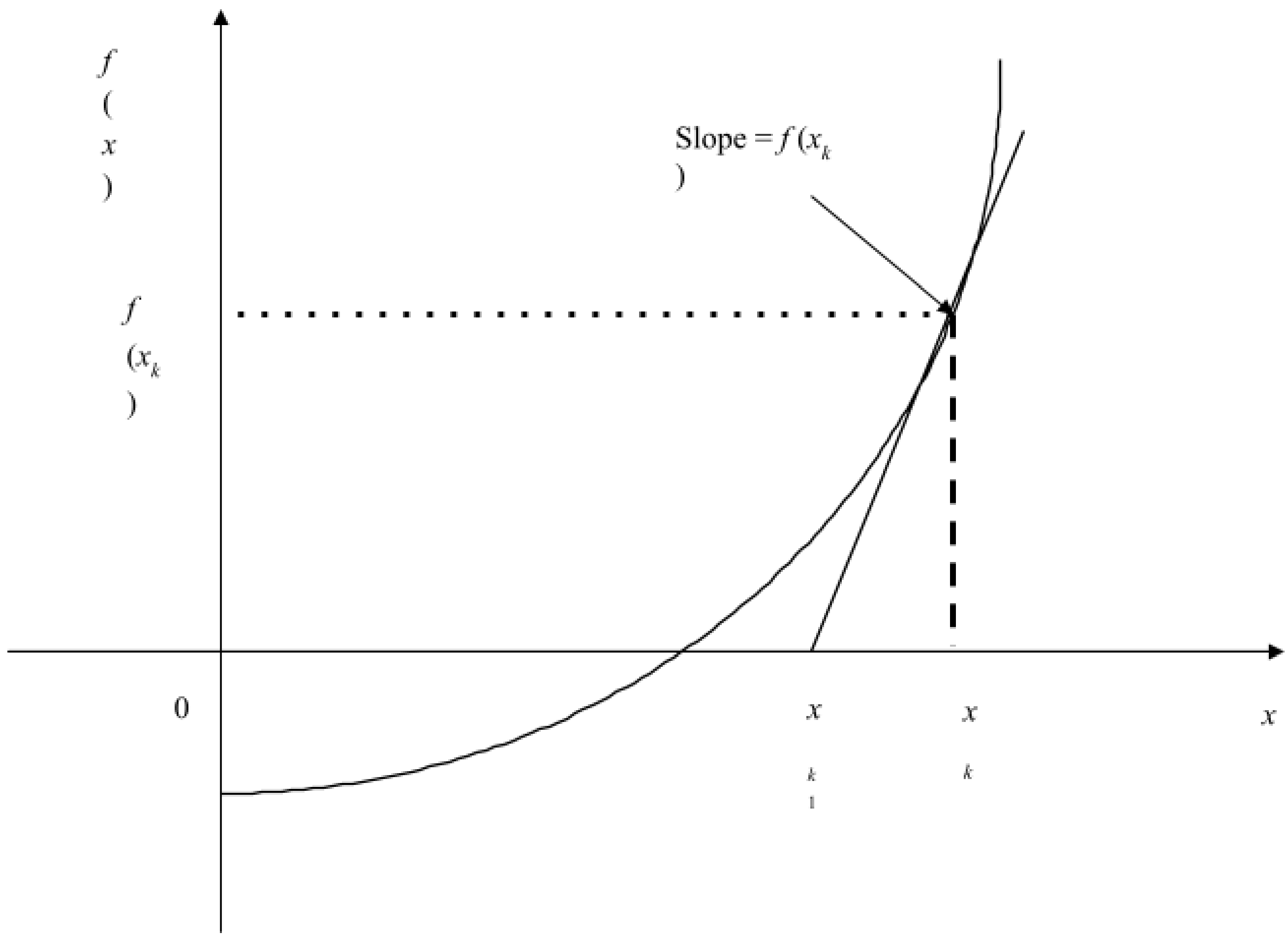
That is $x = x_{up} - \dfrac{f(x_{up})(x_{low} - x_{up})}{f(x_{low}) - f(x_{up})}$

This is the False Position formulae. The value of x then replaces whichever of the two initial guesses, x_{low} or x_{up}, yields a function value with the same sign as $f(x)$. In this way, the values of x_{low} and x_{up} always bracket the true root. The process is repeated until the root is estimated adequately.

Newton Raphson Method:

If $f(x)$, $f'(x)$ and $f''(x)$ are continuous near a root x, then this extra information regarding the nature of $f(x)$ can be used to develop algorithms that will produce sequences $\{x_k\}$ that converge faster to x than either the bisection or false position method. The Newton-Raphson (or simply Newton's) method is one of the most useful and best-known algorithms that relies on the continuity of $f'(x)$ and $f''(x)$.

The attempt is to locate root by repeatedly approximating $f(x)$ with a linear function at each step. If the initial guess at the root is x_k, a tangent can be extended from the point $[x_k, f(x_k)]$. The point where this tangent crosses the x axis usually represents an improved estimate of the root.



The Newton-Raphson method can be derived on the basis of this geometrical interpretation. As in the figure, the first derivative at x is equivalent to the slope:

$$f'(x_k) = \frac{f(x_k) - 0}{x_k - x_{k+1}}$$

which can be rearranged to yield

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

which is called the Newton Raphson formulae.

So the Newton-Raphson Algorithm actually consists of the following steps:

1. Start with an initial guess 0 x and an x-tolerance ε.

2. Calculate $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ $k = 0,1,2,\ldots$

Algorithm - Newton's Method

Input x0, xTol
iters = 1
dx = -f(x0)/fDeriv(x0) (* fcns f and fDeriv *)
root = x0 + dx
While (Abs(dx) > xTol)
dx = -f(root)/fDeriv(root)
root = root + dx
iters = iters + 1
End of while
Return root, iters

Exercise 2. Use the Newton Raphson method to estimate the root of $f(x) = e^{-x} - 1$, employing an initial guess of $x_0 = 0$. The tolerance is = 10^{-8} .