3. Solution of functions

We will look at here one of the most basic issues of numerical analysis, the problem of finding the value of x such that $f(x) = 0$ for a given function f.

For example suppose we had the equation

$$x^3 + 4x^2 = 10$$

We could write the function $f(x)$ as

$$f(x) = x^3 + 4x^2 - 10 = 0$$

Solving for x to give $f(x) = 0$ solves our original equation.

Often the function may not have an analytical solution (unlike the one above.) In which case resort is made to solving the equation numerically. There are very many ways to do this and we shall look at a few of them here.

3.1 Bisection Method

We can plot our function $f(x)$ which may look something like that shown here for a certain range of x:

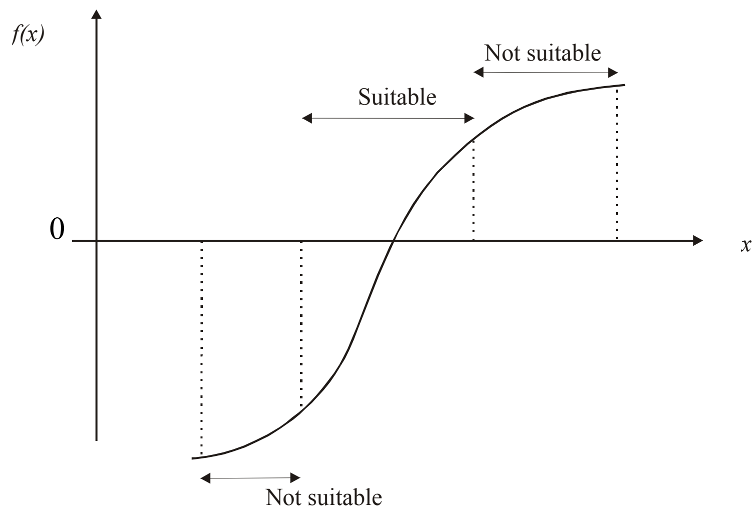

Figure 8: A function

So long as the function is continuous, the solution to the equation is the value of x at the point where the graph crosses the x axis (where $f(x)=0$). This solution is known as a **root** or a **zero** of the function.

We will look in this section at a method called the **bisection** method or **interval-halving** method. The general idea with this method is to first obtain two values of x between which the graph crosses the axis. Then to narrow down the difference between the two x values until we have one which gives $f(x)$ near enough to 0.

We don't have to plot a graph of every function to use this method. What we do is guess two *suitable* values of x, say x_1 and x_2 and calculate the $f(x_1)$ and $f(x_2)$. If the sign of the function change then a solution exists in the interval between x_1 and x_2.

If the sign doesn't change then choose a different x_1 or x_2. A problem may occur if x_1 and x_2 are two far apart and more than one solution occurs in this interval – if this is the case then the method will only calculate one of the solutions.

The next step is to calculate the function value mid-way between our x_1 and x_2 i.e. at $x_3 = (x_1+x_2)/2$. We then look at the sign of $f(x_3)$ and decide if the solution lies between x_1 and x_3 or between x_3 and x_2.

This process is repeated until f(x) ≈ 0 or

$$|f(x)| < \varepsilon$$

Where ε is a predefined error (or tolerance).

This process is shown graphically in the figures below.

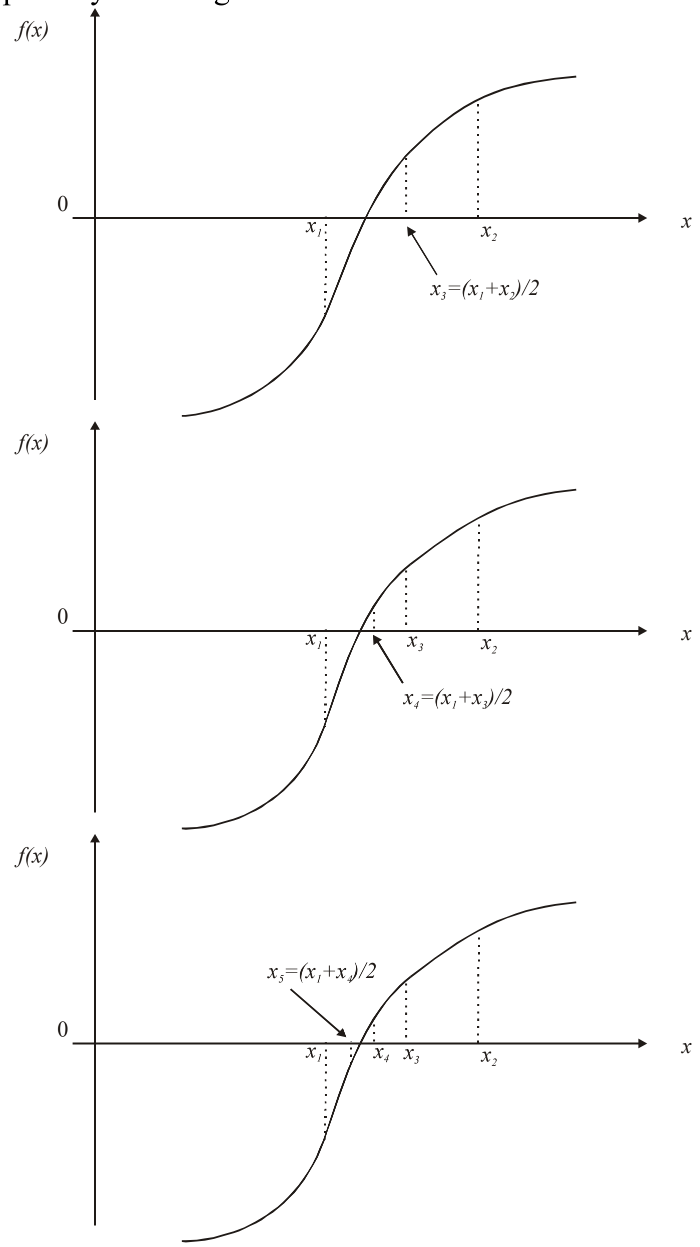

Figure 9: Bisection Method

Example

Use the Bisection method to determine the solution of $x^3 + 4x^2 = 10$ between the values x_1=1 and x_2=2. to a tolerance of $\varepsilon = 0.001$.

We write the function *f(x)* as

$$f(x) = x^3 + 4x^2 - 10 = 0$$

Then for the bisection method we calculate

$$f(1) = -5$$
$$f(2) = 14$$

The sign changes so the solution lies between 1 and 2.

Calculate the function at half way between the points (1.5)

$$f((1+2)/2) = f(1.5) = 2.37$$

We can now see the sign changes between x = 1.5 and 1.0, so calculate the function midway between these to points,

$$f\left(\frac{1+1.5}{2}\right) = f(1.25) = -1.7969$$

The solution can now be seen to lie between 1.25 and 1.5. Repeat this process until the modulus of *f(x)* is below the tolerance. All the iteration steps are shown in the table below, together with a plot of the iteration convergence history.

x	f(x)
1.0000	-5.0000
2.0000	14.0000
1.5000	2.3750
1.2500	-1.7969
1.3750	0.1621
1.3125	-0.8484
1.3438	-0.3510
1.3594	-0.0964
1.3672	0.0324
1.3633	-0.0321
1.3652	0.0001

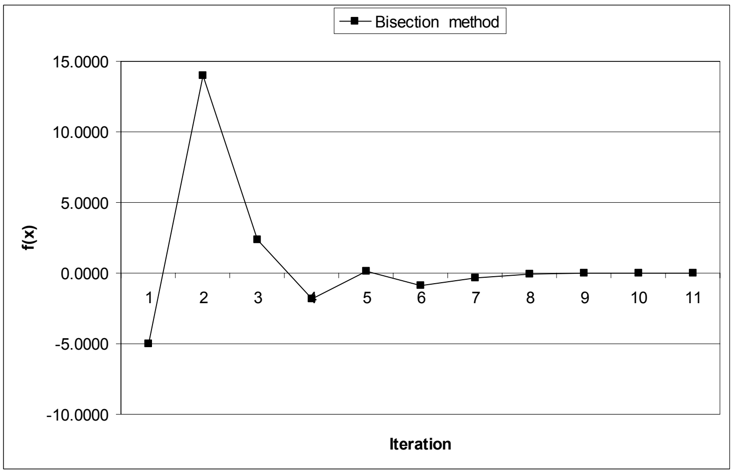

Figure 10: Table of iteration steps and graph of convergence history for Bisection Method

3.2 Method of false-position

A refinement to the bisection method is the **method of false position** (also known as **regula falsi**)

Instead of choosing point mid-way between the two *x* values to get a more accurate solution, this method approximates the function with a straight line, and chooses the point where this line crosses the x-axis as the next point. This can be seen graphically in the figure below:

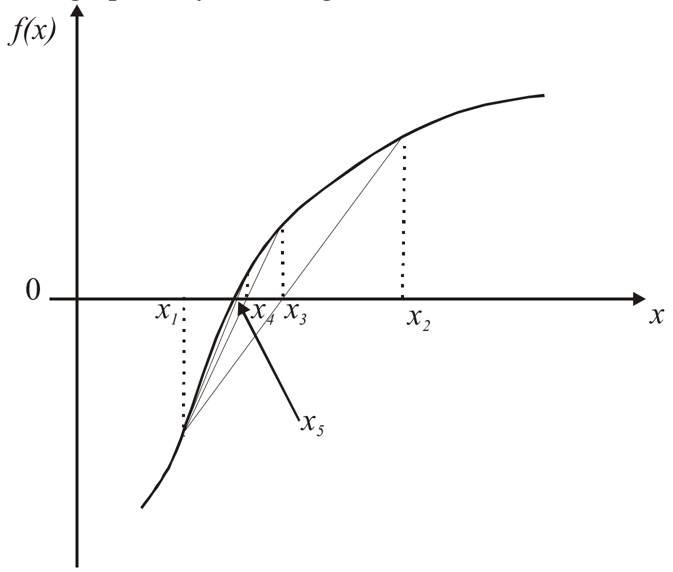

Figure 11: Method of False Position

We can formulate this graphical representation by looking at the equation of the line joining the two points ($x_1, f(x_1)$) and ($x_2, f(x_2)$), which is given by

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

Equation 22

The line cuts the x-axis where

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

Equation 23

so this is the new estimate of the root.

This method usually converges more rapidly than the bisection method as can be seen for this example in the graph of convergence history below:

x	fx
1.0000	-5.0000
2.0000	14.0000
1.2632	-1.6023
1.3388	-0.4304
1.3666	0.0229
1.3652	-0.0003
1.3652	0.0000

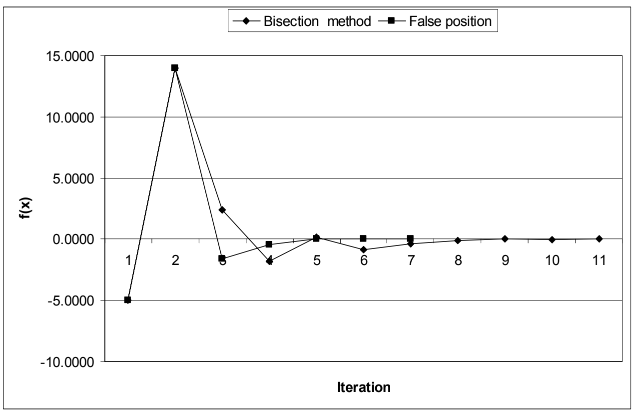

Figure 12: Table of iteration steps for False Position method and convergence history graph for Bisection and False Position methods

These two methods are known as bracketing methods as the solution is bracketed somewhere between the two x values and this fact remains the same until the solution is found. They are guaranteed to converge to a solution.

3.3 Fixed point methods

A different approach to calculating a root is to devise a scheme that produces a convergent sequence whose limit is the root of the function.

For these we take the function $f(x) = 0$ and from it we produce an equation of the form

$$x = g(x)$$

or

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

Equation 24

where the subscript n means the iteration count, such that repeated application of this converges to the root.

The easiest way to do this is to rearrange the original equation. As illustrated by this example.

Example

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

this can be rearranged in many ways, for example:

(a) $x = x^2 - 2 = g_a(x)$

(b) $x = \sqrt{x+2} = g_b(x)$

(c) $x = 1 - \frac{2}{x} = g_c(x)$

Each of these $g(x)$'s are called **iteration functions** of $f(x)$. Once we have chosen the iteration function we just keep repeating the calculation of the $g(x)$ with the new x_{n+1}, until the value of x changes are below the error tolerance chosen,

Take iteration function (b) for example. We need to choose a starting value, so take $x_n=1.1$.

$$x_{n+1} = g(x_n) = \sqrt{x+2}$$
$$x_{n+1} = \sqrt{1.100+2} = 1.761$$
$$x_{n+2} = \sqrt{1.761+2} = 1.939$$
$$x_{n+3} = \sqrt{1.939+2} = 1.985$$
$$x_{n+4} = \sqrt{1.985+2} = 1.996$$

We can see that this converging. In fact it converges to 2 (the two root of the equation are –1 and 2).

What would have happened had we chosen a different iteration function?

The solutions from the other iteration functions are shown in the table and graph below.

(a)	(b)	[c)
1.100	1.100	1.100
-0.790	1.761	-0.818
-1.376	1.939	3.444
-0.107	1.985	0.419
-1.989	1.996	-3.769
1.954	1.999	1.531
1.820	2.000	-0.307

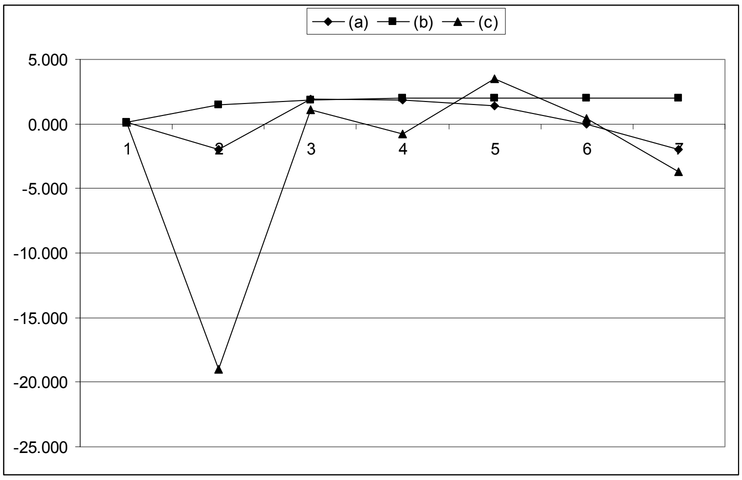

Figure 13: Solution table and Convergence history for functions (a), (b) and (c)

It can be clearly seen that (a) and (c) do not converge to a solution. Thus, we can conclude that it cannot be guaranteed that any particular iteration function will converge to a solution of the original equation.

It may also be the case that the function is undefined (e.g. if we had chosen $x_1 = 0$).

There are techniques available to determine if the iteration function is convergent to a solution and the detail may be found in many general maths text books, in this course we will not go into that detail.

3.4 Newton-Raphson Iteration

A very popular technique used by engineers for solving nonlinear equations is the **Newton-Raphson** method - often called just the Newton method. The method solves the function $f(x) = 0$. And that function **must** be **differentiable**.

The basic idea is that if we have a function value at x_o then if we can take a tangent to the solution and find where this cuts the x-axis then that point is a better solution. This is shown graphically in the figure 14 below.

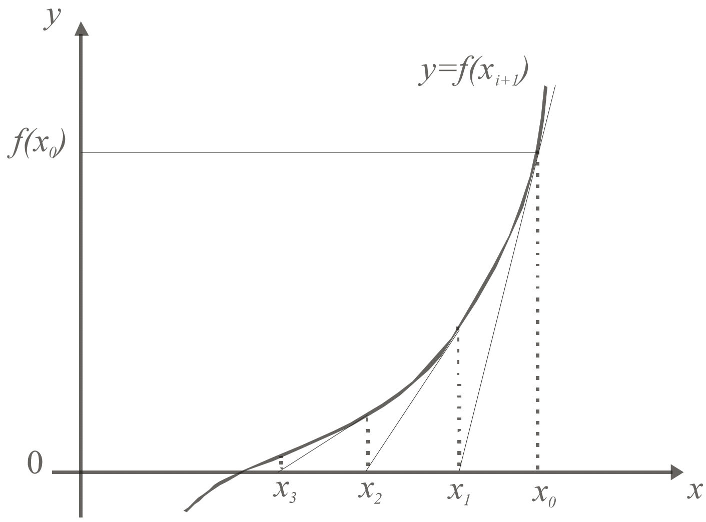

Figure 14: Newton's Method

IF we start with x_0 and calculate $f(x_0)$ we then draw the tangent to the solution at this point $(x_0, f(x_0))$. The point where the tangent crosses the x-axis is then x_1. We then calculate the solution here, $f(x_1)$ and the procedure continues until we have a solution below some error tolerance ε.

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

Equation 25

Now we know that the first derivative of the function at any point is equal to the gradient of the function at that point. This is equal to the angle that the tangent makes with the axis, *tan* β - see figure 14.

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

Equation 26

We then repeat this for the next point, replacing x_0 with the just calculated x_1

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

and so on until the error tolerance is satisfied.

The general Newton formula for the *n*th iteration is

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

Equation 27

An alternative derivation can be obtained by writing the Taylor expansion about x_0, from equation 17 we get:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2$$

If we are *near* to the solution then we can ignore the second derivative term and the equation will approximate the solution. As $f(x) = 0$ we have

$$0 = f(x_0) + f'(x_0)(x - x_0)$$

which can be rearranged to give

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

Which is Newton's iteration formula, equation 27.

Example

Apply Newton's method to the equation $f(x) = x^3 + x - 1 = 0$ starting from a initial value of $x = 1$.

First write down the first derivative

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

The table below shows each of the first 5 steps of the calculation. After 4 steps the calculation is accurate to 5dp.

x	f	f'	x_n+1
1.0000	1.0000	4.0000	0.7500
0.7500	0.1719	2.6875	0.6860
0.6860	0.0089	2.4120	0.6823
0.6823	0.0000	2.3968	0.6823
0.6823	0.0000	2.3967	0.6823

In general Newton's method is faster that the other fixed point methods and the bracketing methods. Formally it has second order convergence – that is it converges at the rate of h^2. However it has the limitation that the function must be differentiable.