2 Solution of Equations

In this section, we will study methods for finding solutions of functions of a single variable i.e. values of x such that $f(x) = 0$.

Consider, for example, the function $f(x) = x - \cos(x)$, and suppose we want to find x such that $f(x) = 0$. We could look directly for values of x where $f(x)$ crosses the x-axis. A rough estimate can be made by sketching the graph.

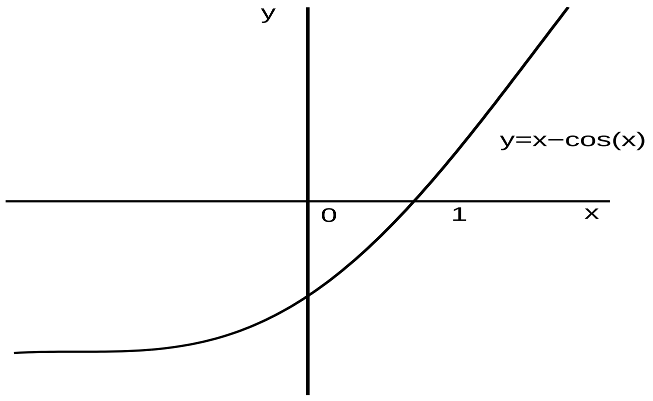

Figure 1: Graph of $y = x - \cos(x)$.

2.1 Bisection method

For the example above, it is easy to see that the solution to $f(x) = 0$ lies between $x = 0$ and $x = 1.5 \approx \pi/2$. Since $f(0) < 0$ and $f(1.5) > 0$ and the graph of $y = f(x)$ varies smoothly as x goes from 0 to 1.5, the graph must cross the x-axis somewhere in the interval $[0, 1.5]$. Using this starting information, we can obtain a refined interval by 'halving the interval' and looking at the sign of the function f evaluated at the mid-point $x = 0.75$.

$f < 0$ at $x =$	$f > 0$ at $x =$	mid point	f	approximate answer	Error Bound
0.0	1.5	0.75	+ve	0.75	± 0.75

$\therefore$ root lies between 0.0 and 0.75.

0.0	0.75	0.375	-ve	0.375	± 0.375

$\therefore$ root lies between 0.375 and 0.75.

0.375	0.75	0.5625	-ve	0.5625	± 0.1875

$\therefore$ root lies between 0.5625 and 0.75.

And continuing we get

$f < 0$ at $x =$	$f > 0$ at $x =$	mid point	f (mid point)
0.0	1.5	0.75	$+$
0.0	0.75	0.375	$-$
0.375	0.75	0.5625	$-$
0.5625	0.75	0.65625	$-$
0.65625	0.75	0.70313	$-$
0.70313	0.75	0.72656	$-$
0.72656	0.75	0.73828	$-$
0.73828	0.75	0.74414	$+$
0.73728	0.74414	0.74121	$+$

Once an interval has been located the method is very easy to implement, very reliable and has good error bounds. The only disadvantage is that the method is slow. (Note: the speed of a method for solving an equation is usually measured in the number of function evaluations required to obtain the answer).

The interval halving or 'bisection' method can be speeded up by making better use of the information computed. The method only uses the sign of the function f and not its value. Thus, if the absolute value of the function is much smaller at one end than at the other it is likely that the root will be close to the end where the function is smaller. This idea is exploited in the **Regula Falsi** method.

2.2 Fixed point iteration

Alternatively, to solve $x - \cos(x) = 0$ we can re-arrange the equation in the form $x = \cos(x)$. The value(s) of x we are seeking then occur where the curves $y = x$ and $y = \cos(x)$ intersect. From the figure below, we see there is only one point of intersection in this example, near $x = 0.7$.

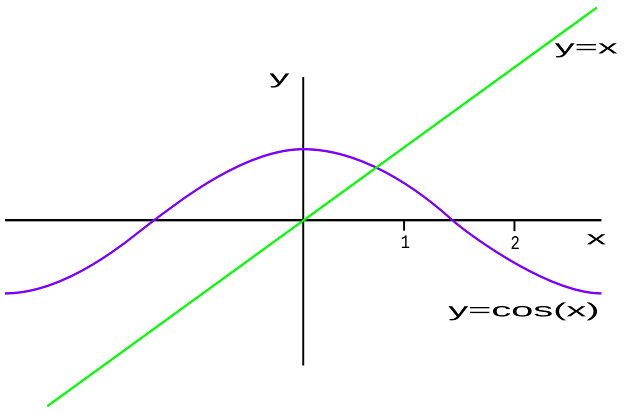

Figure 2: Intersection of $y = x$ and $y = \cos(x)$.

In general, to obtain a fixed point iteration, we rearrange $f(x) = 0$ into the form $x = \phi(x)$ so that if x satisfies $f(x) = 0$ then it also satisfies $x = \phi(x)$. Then the fixed point iteration is the computation $x_{n+1} = \phi(x_n)$. We shall consider three rearrangements of $x - \cos(x) = 0$, namely:

1)	$x = \cos(x)$	:	$\phi(x) = \cos(x)$
2)	$x = x - \frac{1}{2}(x - \cos(x))$	:	$\phi(x) = \frac{1}{2}x + \frac{1}{2}\cos(x)$
3)	$x = x + \frac{1}{2}(x - \cos(x))$	:	$\phi(x) = \frac{3}{2}x - \frac{1}{2}\cos(x)$

Clearly, there are infinitely many possibilities!

Consider the first case with $\phi(x) = \cos(x)$. To perform the iteration $x_{n+1} = \phi(x_n) = \cos(x_n)$ we require a starting guess x_0. Then the computation proceeds as follows:

i)	choose	x_0	$=$	0.75,	(starting guess)		
ii)	compute	x_1	$=$	$\cos(x_0)$		$=$	0.73169
iii)	compute	x_2	$=$	$\cos(x_1)$		$=$	0.74405
iv)	compute	x_3	$=$	$\cos(x_2)$		$=$	0.73573
							etc.

A **fixed point** of the iteration $x_{n+1} = \phi(x_n)$ is any value α that satisfies

$$\alpha = \phi(\alpha).$$

Here, $\alpha = 0.73909\cdots$ is the only fixed point of the iteration $x_{n+1} = \phi(x_n)$, but in general, an iteration may have many fixed points. We say that an iteration $x_{i+1} = \phi(x_i)$ **converges** to β if

$$x_i \to \beta \text{ as } i \to \infty$$

Clearly, if the iteration converges to β then

$$\beta = \phi(\beta)$$

Thus, β is a fixed point of that iteration.

However the converse is not true. For some choices of $\phi(x)$, the fixed point iteration may not converge to the fixed point(s). For example, returning to the above three iterations

1)	$x_{i+1} = \cos(x_i)$	$\rightarrow$	fixed points satisfy $\alpha = \cos(\alpha)$
2)	$x_{i+1} = \frac{1}{2}x_i + \frac{1}{2}\cos(x_i)$	$\rightarrow$	fixed points satisfy $\alpha = \frac{1}{2}\alpha + \frac{1}{2}\cos(\alpha)$ ie. $\alpha = \cos(\alpha)$
3)	$x_{i+1} = \frac{3}{2}x_i - \frac{1}{2}\cos(x_i)$	$\rightarrow$	fixed points satisfy $\alpha = \frac{3}{2}\alpha - \frac{1}{2}\cos(\alpha)$ ie. $\alpha = \cos(\alpha)$

If these iterations converge, they converge to 0.73909.

	Iteration 1	Iteration 2	Iteration 3
i	x_i	x_i	x_i
1	0.73169	0.74084	0.75916
2	0.74405	0.73937	0.77602
3	0.73573	0.73913	0.80719
4	0.74134		0.86501
5	0.73757		0.97320
6	0.74011		1.17847
7	0.73840		
8	0.73955		
9	0.73877		
10	0.73930		
11	0.73891		
12	0.73918		

1) converges, took 12 iterations to reach $|x_i - 0.73909| < 10^{-4}$
2) converges, took 3 iterations to reach $|x_i - 0.73909| < 10^{-4}$
3) diverges.

This method is very easy to program. However, it is not always easy to know how accurate the result is or what is the best choice of $\phi(x)$. It can be very slow and often diverges. However, the following theorem gives us information about the convergence of an iteration.

Theorem: *Convergence of a Fixed Point Iteration*
Let α be a fixed point of the iteration $x_{i+1} = \phi(x_i)$ and let x_0 (the starting guess) lie in the range $(\alpha - h, \alpha + h)$ for some value h. Suppose that $\phi(x)$ is continuous in $[\alpha - h, \alpha + h]$. The iteration converges to α if there exists a positive number λ satisfying $0 < \lambda < 1$ such that

$$|\phi'(x)| \leq \lambda$$

for all x in $(\alpha - h, \alpha + h)$.

To prove this result we need the **mean value theorem** (see Standard Theorems and Results in Section 6).

The **proof** goes in four stages:-

1. Choose any $t \in (\alpha - h, \alpha + h)$ and define $s = \phi(t)$ then

$$s - \alpha = \phi(t) - \alpha = \phi(t) - \phi(\alpha) \qquad \because \alpha = \phi(\alpha).$$

   Now, by the mean value theorem,

$$\phi(t) = \phi(\alpha) + (t - \alpha)\phi'(\xi) \quad \text{with } \xi \text{ lying between } t \text{ and } \alpha.$$

$$\begin{aligned} \therefore\ s - \alpha &= (t - \alpha)\phi'(\xi) \\ |s - \alpha| &= |t - \alpha|\,|\phi'(\xi)|. \end{aligned}$$

Since both t and α lie in the interval $(\alpha - h, \alpha + h)$ then so does ξ and there exists a λ such that,

$$|\phi'(\xi)| \leq \lambda$$
$$\therefore \ |s - \alpha| \leq \lambda |t - \alpha| .$$

That is s is closer to α by at least a factor λ.

2. From part 1. we conclude that if the iterate $x_{i-1} \in (\alpha - h, \alpha + h)$ then so does x_i. Thus if we start with an initial iterate $x_0 \in (\alpha - h, \alpha + h)$ then x_i belongs to $(\alpha - h, \alpha + h)$ for all i.

3. From part 1. we also conclude that if $x_0 \in (\alpha - h, \alpha + h)$ then

$$|x_i - \alpha| \leq \lambda^i |x_0 - \alpha|$$

since $x_0, x_1, \cdots, x_i$ all belong to $(\alpha - h, \alpha + h)$.

4. Finally, since $0 < \lambda < 1$ then $\lambda^i \to 0$ as $i \to \infty$

$$\therefore |x_i - \alpha| \to 0 \text{ as } i \to \infty$$
$$\therefore \text{ convergence.}$$

2.3 Newton Raphson method

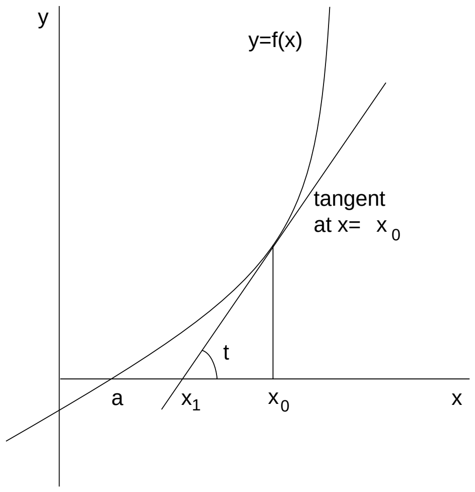

Figure 3: First step of Newton Raphson method.

From the figure, a is the point at which $f(x) = 0$ and x_0 is an estimate of a. The Newton Raphson method computes a new estimate, x_1 in the following way. Construct the tangent to $f(x)$ at $x = x_0$; the point x_1 is where that tangent crosses the axis. From the diagram

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

But from the definition of derivative

$$\tan(t) = \text{slope of tangent} = f'(x_0)$$

Thus

$$f'(x_0) = \frac{f(x_0)}{x_0 - x_1} \quad \therefore x_0 - x_1 = \frac{f(x_0)}{f'(x_0)} \quad \text{ie. } \; x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

The Newton Raphson iteration is

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

$$\text{where } \phi(x) = x - \frac{f(x)}{f'(x)}$$

Suppose we choose $x_0 = 0.75$ in the example above. The Newton Raphson method gives:

i	x_i	$f(x_i) = x_i - \cos(x_i)$	$f'(x_i) = 1 + \sin(x_i)$	x_{i+1}
0	0.75	0.01831	1.68164	0.73911
1	0.73911	0.00005	1.67363	0.73909
2	0.73909	0.00001	1.67361	0.73909

When the Newton Raphson method works it converges very rapidly!

2.4 Fixed point iterations to solve simultaneous linear equations

Consider the system of equations

$$\begin{aligned} 5x_1 + x_2 - x_3 &= 5 \\ -x_1 + 10x_2 - 3x_3 &= 6 \\ -2x_1 - x_2 + 10x_3 &= 7. \end{aligned}$$

We can rearrange these equations into the form

$$\begin{aligned} x_1 &= (5 - x_2 + x_3)/5 \\ x_2 &= (6 + x_1 + 3x_3)/10 \\ x_3 &= (7 + 2x_1 + x_2)/10 \end{aligned}$$

and make this rearrangement the basis for an iterative scheme. We consider the so-called Jacobi and Gauss-Seidel methods.

For the **Jacobi** method, we choose an initial approximation $\underline{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, x_3^{(0)}]^T$ and iterate as follows:

$$\begin{aligned} x_1^{(i+1)} &= \left(5 - x_2^{(i)} + x_3^{(i)}\right)/5 \\ x_2^{(i+1)} &= \left(6 + x_1^{(i)} + 3x_3^{(i)}\right)/10 \\ x_3^{(i+1)} &= \left(7 + 2x_1^{(i)} + x_2^{(i)}\right)/10. \end{aligned}$$

That is, all the old x-values are frozen on the right hand side and used to update the x-values on the left hand side simultaneously. Starting with $\underline{x}^{(0)} = \underline{0}$ gives:

iteration	x_1	x_2	x_3
0	0.0000	0.0000	0.0000
1	1.0000	0.6000	0.7000
2	1.0200	0.9100	0.9600
3	1.0100	0.9900	0.9950
4	1.0010	0.9995	1.0010
exact	1	1	1

The **Gauss Seidel** method is subtly different. The iteration is:

$$\begin{aligned} x_1^{(i+1)} &= \left(5 - x_2^{(i)} + x_3^{(i)}\right)/5 \\ x_2^{(i+1)} &= \left(6 + x_1^{(i+1)} + 3x_3^{(i)}\right)/10 \\ x_3^{(i+1)} &= \left(7 + 2x_1^{(i+1)} + x_2^{(i+1)}\right)/10. \end{aligned}$$

That is, new values are used to update x-values as soon as they become available. When x_1 is updated, it is used in the calculation of the new value of x_2 and so on. Notice that the three equations have to be solved in order for the Gauss-Seidel method. Starting with $\underline{x}^{(0)} = \underline{0}$ gives the iteration:

iteration	x_1	x_2	x_3
0	0.0000	0.0000	0.0000
1	1.0000	0.7000	0.9700
2	1.0054	0.9964	1.0104
3	1.0028	1.0034	1.0009
4	0.9995	1.0002	0.9999
exact	1	1	1

Normally, when both methods work (i.e. converge) the Gauss Seidel method converges faster than the Jacobi method.

2.5 Fixed point iterations to solve simultaneous equations

To give a simple illustration of iterative methods for solving a system of simultaneous equations, we consider the following very simple system of two equations:

$$\begin{aligned} f(x,y) &= x + x^2y^2 - 2y + 3 = 0 \\ g(x,y) &= y - x^3 + 2xy^2 - 2 = 0. \end{aligned}$$

We consider three basic types of iteration

1) Jacobi type iteration
2) Gauss Seidel type iteration
3) Newton iteration

For Jacobi and Gauss Seidel type iterations, the first step is to rearrange the equations into the form:

$$\begin{aligned} x &= \phi(x,y) \\ y &= \psi(x,y). \end{aligned}$$

For example, we can write:

$$\begin{aligned} x &= 2y - x^2y^2 - 3 \\ y &= x^3 - 2xy^2 + 2. \end{aligned}$$

Next, we require an initial guess (x_0, y_0) for the root (x, y).

Jacobi iteration yields:

$$\begin{aligned} x_{i+1} &= 2y_i - x_i^2y_i^2 - 3 \\ y_{i+1} &= x_i^3 - 2x_iy_i^2 + 2. \end{aligned}$$

Gauss Seidel iteration gives:

$$\begin{aligned} x_{i+1} &= 2y_i - x_i^2 y_i^2 - 3 \\ y_{i+1} &= x_{i+1}^3 - 2x_{i+1} y_i^2 + 2. \end{aligned}$$

In practice it can be quite difficult to get these iterations to converge! Both the Jacobi and Gauss Seidel iterations above diverge from the root near $(0.1, 1.5)$. (Try it!) In this example it is indeed difficult to find a rearrangement that leads to a convergent iteration.

However, we can rearrange $g(x, y) = 0$ into the form:

$$x = (y - 2) \,/\, \left(x^2 - 2y^2\right)$$

and $f(x, y) = 0$ into the form:

$$y = (x + 3) \,/\, \left(2 - x^2 y\right).$$

This gives the Jacobi iteration:

$$\begin{aligned} x_{i+1} &= (y_i - 2) \,/\, \left(x_i^2 - 2y_i^2\right) \\ y_{i+1} &= (x_i + 3) \,/\, \left(2 - x_i^2 y_i\right) \end{aligned}$$

Starting with $x_0 = 0.1$, $y_0 = 1.5$ we obtain:

i	$f(x_{i-1}, y_{i-1})$	$g(x_{i-1}, y_{i-1})$	x_i	y_i
1	0.1225	−0.0510	0.1114	1.5617
2	0.0182	0.1035	0.0901	1.5709
3	−0.0317	0.0147	0.0871	1.5550
4	−0.0045	−0.0246	0.0922	1.5527
5	0.0073	−0.0036	0.0929	1.5564
6	0.0011	0.0058	0.0917	1.5569
7	0.0017	0.0009	0.0916	1.5561
8	−0.0003	−0.0014	0.0918	1.5559
9	0.0004	−0.0002	0.0919	1.5561
10	0.0001	0.0003	0.0918	1.5562
etc.				

The process eventually converges to $(x, y) = (0.0918, 1.5561)$.

The corresponding Gauss Seidel iteration is:

$$\begin{aligned} x_{i+1} &= (y_i - 2) \,/\, \left(x_i^2 - 2y_i^2\right) \\ y_{i+1} &= (x_{i+1} + 3) \,/\, \left(2 - x_{i+1}^2 y_i\right). \end{aligned}$$

Using the same initial guess, we obtain,

i	$f(x_{i-1}, y_{i-1})$	$g(x_{i-1}, y_{i-1})$	x_i	y_i
1	0.1225	−0.1058	0.1114	1.5703
2	0.0014	0.1181	0.0874	1.5530
3	0.0002	−0.2630	0.0928	1.5568
4	0.0000	0.0060	0.0916	1.5560
5	0.0000	0.0013	0.0919	1.5562
etc.				

This process converges to $(0.0918, 1.5561)$ faster than the equivalent Jacobi iteration.

The third type of iteration we consider, **Newton iteration**, is closely related to the Newton Raphson method. To define the method, we first have to introduce the **Jacobian matrix** J. This is the 2×2 matrix of first partial derivatives of the functions f and g, defined by:

$$J = \begin{bmatrix} \dfrac{\partial f}{\partial x} & \dfrac{\partial f}{\partial y} \\ \dfrac{\partial g}{\partial x} & \dfrac{\partial g}{\partial y} \end{bmatrix}.$$

Note, in the example above, we have:

$$\frac{\partial f}{\partial x} = 1 + 2xy^2,\ \frac{\partial f}{\partial y} = 2x^2y - 2,\ \frac{\partial g}{\partial x} = -3x^2 + 2y^2,\ \frac{\partial g}{\partial y} = 1 + 4xy.$$

Now, having chosen an initial guess (x_0, y_0) we compute:

1. $f_0 = f(x_0, y_0)$, $g_0 = g(x_0, y_0)$
2. J_0, the Jacobian matrix J evaluated at (x_0, y_0).

The first step of the Newton iteration is then:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} - J_0^{-1} \begin{bmatrix} f_0 \\ g_0 \end{bmatrix}.$$

In other words, we have to solve the linear system:

$$J_0 \begin{bmatrix} r_0 \\ s_0 \end{bmatrix} = \begin{bmatrix} f_0 \\ g_0 \end{bmatrix}$$

and then perform a correction step:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} - \begin{bmatrix} r_0 \\ s_0 \end{bmatrix}.$$

The process is then repeated with (x_1, y_1) replacing (x_0, y_0) and a further estimate (x_2, y_2) is computed. The general iteration is:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} - J_i^{-1} \begin{bmatrix} f_i \\ g_i \end{bmatrix}.$$

The iteration is continued until convergence. For the above example with $x_0 = 0.1$, $y_0 = 1.5$ we obtain:

i	$f(x_{i-1}, y_{i-1})$	$g(x_{i-1}, y_{i-1})$	x_i	y_i
1	-0.1225	0.0510	0.0914	1.5559
2	-0.0001	-0.0023	0.0918	1.5561
3	4×10^{-7}	6×10^{-7}	0.0918	1.5561

This is rapidly converging to the root.