Contents

page

CHAPTER TWO

Nonlinear equations

This chapter deals with way of solving nonlinear equations using different types of numerical methods.

The following equations are called non – linear equations

- ✓ Quadratic equations given by: $ax^2 + bx + c = 0, a \neq 0$
- ✓ Cubic equations given by: $ax^3 + bx^2 + cx + d = 0, a \neq 0$
- ✓ General polynomial equations of degree $n \geq 4$, given by

  $a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0, a_n \neq 0$
- ✓ Equations which contains transcedential functions such as $e^x, \ln x, \sin x, \cos x, \sinh x$ etc. that is: $\sin x + \cos x = 0$

  $e^{-x} + x^2 - x + 1 = 0$

2.1. Locating roots

The Root-Finding Problem: Given a function $f(x)$, find $x = \alpha$ such that $f(\alpha) = 0$. The number α is called a root of $f(x) = 0$ or a zero of the function $f(x)$.

The function of $f(x)$ may be algebraic or trigonometric functions. The well-known examples of algebraic functions are polynomials. In this chapter, we shall describe iterative methods, i.e. methods that generate successive approximations to the exact solution, by applying a numerical algorithm to the previous approximation. The indirect or iterative methods are further divided into two categories: bracketing and open methods. The bracketing methods require the limits between which the root lies, whereas the open methods require the initial estimation of the solution. Bisection and False position methods are two known examples of the bracketing methods. Among the open methods, the secant and Newton-Raphson is most commonly used.

2.2. Bisection method

Suppose we have an equation $f(x) = 0$, where f is continuous on $[a, b]$ and $f(a) < 0$ and $f(b) > 0$ or vice versa and now assume the first one. In order to find a root of $f(x) = 0$ lying in the interval $[a, b]$ we divide the interval into half. i.e.

$$m = \frac{a+b}{2} \tag{1}$$

and then

1. If $f(m) = 0$, then m is the exact root of the equation.
2. If $f(m) < 0$, then the root lies in the interval $[m, b]$.
3. If $f(m) > 0$, then the root lies in the interval $[a, m]$.

Then the newly reduced interval denoted by $[a_1, b_1]$ is again halved and the same investigation is made. Finally, at some stage in the process, we get either the exact root of $f(x) = 0$ or a finite sequence of intervals $[a, b] \supseteq [a_1, b_1] \supseteq \cdots \supseteq [a_n, b_n]$ such that $f(a_n) < 0$ and $f(b_n) > 0$. If $m_n = \frac{a_n+b_n}{2}$, the process terminates and the approximate root is m_n.

Stopping Criteria

Since in this an iterative method, we must determine some stopping criteria that will allow the iteration to stop. Here are some commonly used stopping criteria.

Let ε be the tolerance; that is, we would like to obtain the root with an error of at most of ε. Then Accept $x = m_k$ as a root of $f(x) = 0$ if any of the following criteria is satisfied:

1. $|f(m_k)| \leq \varepsilon$ (The functional value is less than or equal to the tolerance).

2.$|\, m_{k-1} - m_k| \leq \varepsilon$

2. $\frac{|m_{k-1}-m_k|}{|m_k|} \leq \varepsilon$ (The relative change is less than or equal to the tolerance).

3. $\frac{(b-a)}{2^k} \leq \varepsilon$ (The length of the interval after k iterations is less than or equal to the tolerance).

Number of Iterations Needed in the Bisection Method to Achieve Certain Accuracy

Let's now find out what is the minimum number of iterations *n* needed with the Bisection method to achieve a certain desired accuracy.

Length of the first interval $L_0 = b - a$, after 1 ite ration $L_1 = \frac{L_0}{2}$, after 2 iterations $L_2 = \frac{L_0}{4}$ ... ... , after n iterations $L_n = \frac{L_0}{2^n}$. Since the pre-determined error tolerance ε for the error cannot be larger than the length of the interval, then, knowing ε it is always easy to predict the number of iterations **n**: $\frac{L_0}{2^n} \leq \varepsilon \Rightarrow 2^n \geq \frac{L_0}{\varepsilon} \Rightarrow nln2 \geq \ln\left(\frac{L_0}{\varepsilon}\right) \Rightarrow n \geq \frac{[\ln(L_0)-\ln(\varepsilon)]}{ln2}$

Therefore, the number of iterations n needed in the Bisection Method to obtain an accuracy (error tolerance) of ε is given by

$$\boldsymbol{n \geq \frac{[\ln(b-a)-\ln(\varepsilon)]}{ln2}} \qquad \textbf{(2)}$$

Remarks: Since the number of iterations n needed to achieve a certain accuracy depends upon the initial length of the interval containing the root, it is desirable to choose the initial interval $[a\ \ b]$ as small as possible.

Example 2.2 Solve the equation $x^3 - 5x + 3 = 0$ using the Bisection method in the interval $[0,1]$ with error $\varepsilon = 0.001$.

Solution: Here $f(x) = x^3 - 5x + 3, a = 0$ and $b = 1$

f is continuous in the given interval $[0,1]$ and $f(0) = 3 > 0, f(1) = -1 < 0$. Hence, there is a root of the equation in the given interval.

$$m_1 = \frac{a+b}{2} = \frac{0+1}{2} = 0.5 \text{ and } f(0.5) = 0.625 > 0$$

Then, the root lies in the interval $[0.5,1]$ and $e_1 = |0.5 - 1| = 0.5 > 0.001$

$$m_2 = \frac{0.5+1}{2} = 0.75 \text{ and } f(0.75) = -0.328125 < 0$$

Then, the root lies in the interval $[0.5,0.75]$ and $e_2 = |0.5 - 0.75| = 0.25 > 0.001$

$$m_3 = \frac{0.5+0.75}{2} = 0.625 \text{ and } f(0.625) = 0.11914 > 0$$

Then, the root lies in the interval $[0.625,0.75]$ and $e_3 = |0.625 - 0.75| = 0.125 > 0.001$

$$m_4 = \frac{0.625+0.75}{2} = 0.6875 \text{ and } f(0.6875) = -0.112548 < 0$$

Then, the root is in the interval $[0.625,0.6875]$ & $e_4 = |0.625 - 0.6875| = 0.0.0625 > 0.001$

$$m_5 = \frac{0.625+0.6875}{2} = 0.65625 \text{ and } f(0.65625) = 0.001373 > 0$$

Then, the root lies in the interval $[0.65625,0.6875]$ and

$$e_5 = |0.65625 - 0.6875| = 0.03125 > 0.001$$

$$m_6 = \frac{0.65625+0.6875}{2} = 0.671875 \text{ and } f(0.671875) = -0.05607986 < 0$$

Then, the root lies in the interval $[0.65625,0.671875]$ and

$$e_6 = |0.65625 - 0.671875| = 0.015625 > 0.001$$

$$m_7 = \frac{0.65625+0.671875}{2} = 0.6640625 \text{ and } f(0.6640625) = -0.02747488 < 0$$

Then, the root lies in the interval $[0.65625, 0.6640625]$ and

$$e_7 = |0.65625 - 0.6640625| = 0.0078125 > 0.001$$

$$m_8 = \frac{0.65625+0.6640625}{2} = 0.66015625$$

$$f(0.66015625) = -0.01308 < 0$$

Then, the root lies in the interval $[0.65625, 0.66015625]$ and

$$e_8 = |0.65625 - 0.66015625| = 0.00390625 > 0.001$$

$$m_9 = \frac{0.65625+0.66015625}{2} = 0.658203125$$

$$f(0.658203125) = -0.00586 < 0$$

Then, the root lies in the interval $[0.65625, 0.658203125]$ and

$$e_9 = |0.65625 - 0.658203125| = 0.001953125 > 0.001$$

$$m_{10} = \frac{0.65625+0.658203125}{2} = 0.657226562$$

$$f(0.657226562) = -0.002246 < 0$$

Then, the root lies in the interval $[0.65625, 0.657226562]$ and

$$e_{10} = |0.65625 - 0.657226562| = 0.000976562 < 0.001$$

Since $|0.65625 - 0.657226562| = 0.000976562 < 0.001 = \varepsilon$, the process terminates and the approximate root of the equation is $m_{10} = 0.657226562$ Table 1 shows the entire iteration procedure of bisection method.

iter	A	b	m=(a+b)/2	f(a)	f(b)	f(m)	swap	|b-a|
1	0	1	0.5	3	-1	0.625	a=m	1
2	0.5	1	0.75	0.625	-1	-0.328125	b=m	0.5
3	0.5	0.75	0.625	0.625	-0.328125	0.11914	a=m	0.25

4	0.625	0.75	0.6875	0.11914	-0.328125	-0.112548	b=m	0.125
5	0.625	0.6875	0.65625	0.11914	-0.112548	0.001373	a=m	0.0625
6	0.65625	0.6875	0.671875	0.001373	-0.112548	-0.05607986	b=m	0.03125
7	0.65625	0.671875	0.6640625	0.001373	-0.056079	-0.02747488	b=m	0.015625
8	0.65625	0.6640625	0.66015625	0.001373	-0.027474	-0.01308	b=m	0.0078125
9	0.65625	0.66015625	0.658203125	0.001373	-0.01308	-0.00586	b=m	0.0039062
10	0.65625	0.658203125	0.657226562	0.001373	-0.00586	-0.002246	b=m	0.0019531
11	0.65625	0.657226562	0.656738281	0.001373	-0.002246	-0.00043678	b=m	0.0009765

From table 1 at the eleventh iteration we get $|b-a| = 0.0009765 \leq 0.001$ or

$\frac{|0.657226562 - 0.656738281|}{|0.656738281|} \leq \varepsilon \implies 0.00074349 \leq 0.001 = \varepsilon$, the process terminates and the approximate root of the equation is $m_{11} = 0.656738281$.

Exercise: Solve the equation $x^2 - 2 = 0$ using the Bisection method in the interval $[1,2]$ with $\varepsilon = 0.05$.

Advantages of bisection method

a) The iteration using bisection method always produces a root, since the method brackets the root between two values.

b) As iterations are conducted, the length of the interval gets halved. So one can guarantee the convergence in case of the solution of the equation.

c) The Bisection Method is simple to program in a computer.

d) The number of iterations needed to achieve a specific accuracy is known in advance.

Disadvantages of bisection method

a) The convergence of the bisection method is slow as it is simply based on halving the interval.

b) Bisection method cannot be applied over an interval where there is a discontinuity.

c) Bisection method cannot be applied over an interval where the function takes always values of the same sign.

d) The method fails to determine complex roots.

e) If one of the initial guesses a_o or b_o is closer to the exact solution, it will take larger number of iterations to reach the root.

Exercise 2.1 Solve the following equations using the bisection method

1. $x^2 - 2 = 0$ in the interval $[1,2]$ with $\varepsilon = 0.05$.
2. $e^x - 3x = 0$ in the interval $[1.5,1.6]$ correct to two decimal places. $ans\ \ x = 1.51$
3. $x^3 - 4x - 8.95 = 0$ in the interval 2, 3][accurate to three decimal places. $Ans\ \ x = 2.711$

2.3. Interpolation (false position) method

This method is also based on the intermediate value theorem. In this method also, as in bisection method, we choose two points a_n and b_n such that $f(a_n)$ and $f(b_n)$ are of opposite signs That is $f(a_n)f(b_n) < 0$). Then, intermediate value theorem suggests that a zero of f lies in between a_n and b_n, if f is a continuous function.

The curve $y = f(x)$ is not generally a straight line. Assume that f is continuous in the interval $[a, b]$ and $f(a)f(b) < 0$.

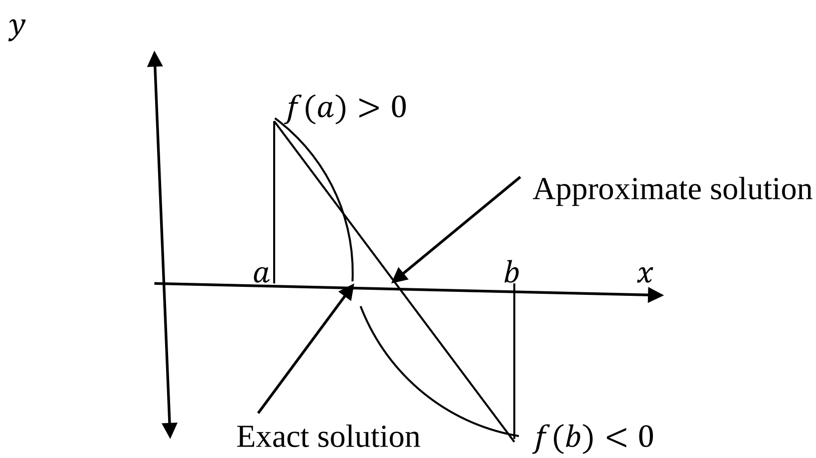

Fig.3 False position method

Now, the equation of the straight line which passes through the points $(a, f(a))$ and $(b, f(b))$ is given by:

$$\frac{y-f(a)}{x-a} = \frac{f(b)-f(a)}{b-a}$$

$$\Rightarrow (b-a)\big(y-f(a)\big) = (x-a)(f(b)-f(a))$$

$$\Rightarrow x = \frac{(b-a)\big(y-f(a)\big)+af(b)-af(a)}{f(b)-f(a)}$$

Now, if we find the x −intercept, then it is given as follows:

$$x = \frac{-bf(a)+af(a)+af(b)-af(a)}{f(b)-f(a)} = \frac{af(b)-bf(a)}{f(b)-f(a)}$$

Therefore, the approximate root of the equation $f(x) = 0$ is given by the formula

$$\tilde{x} = \frac{af(b)-bf(a)}{f(b)-f(a)}$$

Suppose that $f(a) < 0$ and $f(b) > 0$, then as in the Bisection method we follow three cases.

1. If $f(\tilde{x}) = 0$, then $\tilde{x}$ is the exact root of the equation.
2. If $f(\tilde{x}) < 0$, then the root lies in the interval $[\tilde{x}, b]$.
3. If $f(\tilde{x}) > 0$, then the root lies in the interval $[a, \tilde{x}]$.

✓ If either case 2 or 3 occurs, then the process is repeated until the root is obtained to the desired accuracy.

Algorithm: Given a function $f\ (x)$ continuous on an interval $[a_0\ , b_0]$ and satisfying

$$f(a_0)f(b_0) < 0$$

Step 1 set $n = 0$

Step 2 Compute $x_n = \frac{\begin{vmatrix} a_n & b_n \\ f(a_n) & f(b_n) \end{vmatrix}}{f(b_n)-f(a_n)}$ (3)

If $f(x_n) = 0$ accept x_n as a solution (root) and stop.
Else continue.

If $f(a_n\ ) f(\ x_n) < 0$, set $a_{n+1} = a_n, b_{n+1} = x_n$. Else set $a_{n+1} = x_n, b_{n+1} = b_n$.
Then $f\ (x) = 0$ for some x in $[\ a_{n+1}, b_{n+1}\ ]$.
In some cases the method of false position fails to give a small interval in which the zero is known to lie, we terminate the iterations

If $|f(x_n)| \leq \varepsilon$ or $|x_n - x_{n-1}| \leq \varepsilon$ or $\frac{|x_n - x_{n-1}|}{|x_{n-1}|} \leq \varepsilon$, where ε is a specified tolerance value.
The bisection method uses only the fact that f (a) f (b) < 0 for each new interval [a, b], but the false position method uses the values of f (a) and f (b). This is an example showing how one can include additional information in an algorithm to build a better one.

Example 1: find the roots of the equations by false position method for $f(x) = x^3 + x - 1$

Example 2: the function $f(x) = x^3 - x^2 - 1$ has exactly one zero in $[1,2]$. Use method of false position to approximate the zero of f to within the tolerance 10^{-4}.

Solution 2: a root lies in the interval [1,2] since $f(1) = -1\ and\ f(2) = 3$. Starting with $a_0 = 1$ and $b_0 = 2$, using the formula of false position method $x_n = \frac{\begin{vmatrix} a_n & b_n \\ f(a_n) & f(b_n) \end{vmatrix}}{f(b_n)-f(a_n)}$.

$x_0 = \frac{\begin{vmatrix} a_0 & b_0 \\ f(a_0) & f(b_0) \end{vmatrix}}{f(b_0)-f(a_0)} = \frac{\begin{vmatrix} 1 & 2 \\ -1 & 3 \end{vmatrix}}{3-(-1)} = 1.25$ *and* $f(x_0) = f(1.25) = -0.609375$. Here, $f(x_0)$ has the same sign as $f(a_0)$ and so the root must lie on the interval $[x_0, b_0] = [1.25,2]$. Next we set $a_1 = x_0$ and $b_1 = b_0$ to get the next approximation.

$$x_1 = \frac{\begin{vmatrix} a_1 & b_1 \\ f(a_1) & f(b_1) \end{vmatrix}}{f(b_1) - f(a_1)} = \frac{\begin{vmatrix} 1.25 & 2 \\ -0.609375 & 3 \end{vmatrix}}{3 - (-0.609375)} = 1.37662337 \ \ and \ f(x_1) = f(1.37662337) = -0.2862640.$$

Now $f(x)$ changes sign on $[a_1, b_1] = [1.37662337,2]$. Thus we set $a_2 = x_1$ and $b_2 = b_1$.

Continuing in this converges to the root ($r = 1.465558$).

iter	A	B	C	f(c)	|b-a|
0	1.0000	2.0000	1.250000	-0.609375	1.000000
1	1.2500	2.0000	1.376623	-0.286264	0.750000
2	1.3766	2.0000	1.430925	-0.117660	0.623377
3	1.4309	2.0000	1.452402	-0.045671	0.569075
4	1.4524	2.0000	1.460613	-0.017331	0.547598
5	1.4606	2.0000	1.463712	-0.006520	0.539387
6	1.4637	2.0000	1.464875	-0.002445	0.536288
7	1.4649	2.0000	1.465310	-0.000916	0.535125
8	1.4653	2.0000	1.465474	-0.000343	0.534690
9	1.4655	2.0000	1.465535	-0.000128	0.534526
10	1.4655	2.0000	1.465558	-0.000048	0.534465

Table 2 solution $x^3 - x^2 - 1 = 0$ using false position method with $a_0 = 1$ and $b_0 = 2$.

2.4. Iteration Methods

A number p is a fixed point for a given function g if $g(p) = p$.

Given a root-finding problem $f(p) = 0$, we can define functions g with a fixed point at p in a number of ways.

If the function g has a fixed point at p, then the function defined by $f(x) = x - g(x)$ has a zero at p.

To find the roots of the function $f(x) = 0$ by approximating the fixed point of a function $g(x)$, then the formula $x = g(x)$ can be expressed by the iterative formula, for $n \geq 0$

$$x_{n+1} = g(x_n) \quad n = 0,1,2, \ldots \qquad (4)$$

One start with initial guess x_0 calculate $x_1 = g(x_0)$. The iterative method continues with repeated substitutions into the g(x) function to obtain the values

$$x_2 = g(x_1)$$
$$x_3 = g(x_2)$$
$$\vdots$$
$$x_{n+1} = g(x_n)$$

If the sequence of values $\{x_n\}^{\infty}_{n=1}$ converges to p , then

$$\lim_{x\to\infty} x_{n+1} = p = \lim_{x\to\infty} g(x_n)$$

and p is called a fixed point of the mapping.

Terminate the solution if $|x_{n+1} - x_n| \leq \varepsilon$ (the required accuracy).

Example The function $g(x) = x^2 - 2$, for $-2 \leq x \leq 3$, has fixed points at $x = -1$ and $x = 2$ since $g(-1) = (-1)^2 - 2 = -1$ and $g(2) = (2)^2 - 2 = 2$.
Example 2.4 Solve the equation $x^3 - 2x - 5 = 0$ using trial and error method.

Solution: Let $x^3 - 2x - 5 = 0 \Rightarrow x(x^2 - 2) = 5 \Rightarrow x^2 - 2 = \frac{5}{x} \Rightarrow x = \sqrt{\frac{5}{x} - 2} = g(x)$

Or $x^3 - 2x - 5 = 0 \Rightarrow 2x = x^3 - 5 \Rightarrow \frac{x^3-5}{2} = g(x)$

Or $x^3 - 2x - 5 = 0 \Rightarrow x^3 = 2x + 5 \Rightarrow x = \sqrt[3]{2x+5} = g(x)$

Now, let us use the first equation and initial value $x_0 = 0.5$, then

$$x_1 = \sqrt{\frac{5}{x_0} + 2} = \sqrt{\frac{5}{0.5} + 2} = 3.4641$$

$$x_2 = \sqrt{\frac{5}{x_1} + 2} = \sqrt{\frac{5}{3.4641} + 2} = 1.8556$$

$$x_3 = \sqrt{\frac{5}{x_2} + 2} = \sqrt{\frac{5}{1.8556} + 2} = 2.1666$$

$x_4 = 2.0754, x_5 = 2.0997, x_6 = 2.0949, x_7 = 2.0949$ etc.

Convergence condition for the fixed-point iteration scheme

Consider the equation $f(x) = 0$, which has the root a and can be written

as the fixed-point problem $g(x) = x$. If the following conditions hold which has the root a and can be written as the fixed-point problem $g(x) = x$. If the following conditions hold

1. $g(x)$ and $g'(x)$ are continuous functions;
2. $|g'(a)| < 1$

then the fixed-point iteration scheme based on the function g will converge to a.

Alternatively, if $|g'(a)| > 1$ then the iteration will not converge to a.

(Note that when $|g'(a)| = 1$ no conclusion can be reached.)

Exercise 2.3 Solve the following equations using the method of successive approximation.

1. $x^3 - 2x - 3 = 0$ with $x_0 = 1$ correct to three decimal places. $ans\ \ x = 1.324$
2. $2x - \log_{10} x - 9$ with $x_0 = 4$ correct to four decimal places. $ans\ \ x = 4.8425$
3. $cos\, x - 3x + 5 = 0$. Correct to four decimal places. $ans\ \ x = 1.6427$

2.5. Newton-Raphson Method

most nonlinear functions can be approximated by a set of tangents over small intervals. Consider $f\ (x) = 0$, where f has continuous derivative f'. Assume that a is a solution to the equation $f\ (x) = 0$. In order to find the value of a, we start with any arbitrary point x_0. We have slopes, that is tangent to the curve f at $(\ x_0, f\ (x_0\ ))$ (with slope $f'(x_0\ )$) touches the x-axis at x_1.

We have $f'(x_0) = \frac{f\ (x_0\ )-0}{x_0 - x_1} \quad \Longrightarrow \quad x_1 = x_0 - \frac{f\ (x_0\ )}{f'(x_0)}$, similarly $x_2 = x_1 - \frac{f\ (x_1\ )}{f'(x_1)}$

More generally, we write x_{n+1} in terms of $x_n, f\ (x_n\ )$ and $f'(x_n)$ for $n = 0,1,2, \dots$. by means of the **Newton-Raphson** formula

$$\boldsymbol{x_{n+1} = x_n - \frac{f\ (x_n\ )}{f'(x_n)}} \qquad \textbf{(5)}$$

Termination criteria for newton-Raphson method

$|f(x_n)| \le \varepsilon$, $|x_{n+1} - x_n| \le \varepsilon$ or $\frac{|x_{n+1} - x_n|}{x_{n+1}} \le \varepsilon$ where ε is a specified tolerance value, Also, check if the number of iterations has exceeded the maximum number of iterations allowed. If so, one needs to terminate the algorithm and notify the user.

Example: use newton's method to compute the root of the function $f(x) = x^3 - x^2 - 1$ to an accuracy (tolerance) of 10^{-4} use $x_0 = 1$.

Solution: the derivatives of the function is $f'(x) = 3x^2 - 2x$ then $f'(x_0) = f'(1) = 1$, $f(x_0) = f(1) = -1$

The first approximation (iterations), $x_1 = x_0 - \frac{f\ (x_0\ )}{f'(x_0)} = 1 - \frac{-1}{1} = 2.$ then $f'(x_1) = f'(2) = 8$,

$f(x_1) = f(2) = 3.$

The second iteration, $x_2 = x_1 - \frac{f\,(x_1\,)}{f'(x_1)} = 2 - \frac{3}{8} = 1.625$. then $f'(x_2) = f'(1.625) = 4.671875$, $f(x_2) = f(1.625) = 0.650391$. $|x_2 - x_1| = |1.625 - 2| = 0.375 > 10^{-4}$.

See table 4 for the iterations upto approximation of roots.

iter	x_n	$f(x_n)$	$df(x_n)$	$\|x_{n+1}-x_n\|$
0	1.000000	-1.000000	1.000000	
1	2.000000	3.000000	-3.609375	0.750000
2	1.625000	0.650391	4.671875	0.375000
3	1.485786	0.072402	3.651108	0.139214
4	1.465956	0.001352	3.515168	0.019830
5	1.465571	0.000001	3.512556	0.000385
6	1.465571	0.000000	3.512555	0.000000

The root of the equation at 6th iteration is root =1.4656

Table 4 solution of $(x) = x^3 - x^2 - 1 = 0$ using newton's method with $x_0 = 1$.

Some Familiar Computations Using the Newton-Raphson Method

1. Computing the Square Root of a Positive Number A

Computing $\sqrt{A}$ is equivalent to solving $x^2 - A = 0$. the number $\sqrt{A}$, may be computed by applying the Newton-Raphson Method to $f(x) = x^2 - A$, $f'(x) = 2x$, we have the following Newton iterations to generate the root of A.

$$x_{n+1} = x_n - \frac{{x_n}^2 - A}{2x_n} = \frac{{x_n}^2 + A}{2x_n} \qquad (6)$$

for $n = 0,1,2, \ldots$ until convergence.

Compute $\sqrt{2}$

2. Computing the nth Root A

It is easy to see that the above Newton-Raphson Method to compute $\sqrt{A}$ can be easily generalized to compute $\sqrt[n]{A}$. In this case $f(x) = x^n - A$, $f'(x) = nx^{n-1}$.

Thus Newton's Iterations in this case are:

$$x_{k+1} = x_k - \frac{{x_k}^n - A}{n{x_k}^{n-1}} = \frac{(n-1){x_k}^n + A}{n{x_k}^{n-1}} \qquad (7)$$

Compute $\sqrt[3]{3}$

3. Finding the Reciprocal of a Nonzero Number A

The problem here is to compute $\frac{1}{A}$, where $A \neq 0$. Again, the newton-raphson method can be applied with $f(x) = \frac{1}{x} - A$. We have $f'(x) = -\frac{1}{x^2}$. The sequence $x_{n+1}$is given by

$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{\frac{1}{x_n} - A}{-\frac{1}{x_n^2}}$ This leads to the general formula

$$x_{n+1} = x_n(2 - Ax_n) \tag{8}$$

In order to get convergence choose $x_0 < \frac{2}{A}$, we can get x_0 satisfies this equation by linear interpolation of $x_0 = p_1(A) = 3 - 2A$.

Example: find the reciprocal of 0.8.

Solution: let A=0.8 so we need to approximate $\frac{1}{0.8}$ use the initial guess of $x_0 = 1.4 = 3 - 2x0.8$

The newton's method gives the solution as

$x_1 = x_0(2 - (0.8)x_0) = (1.4)\big(2 - (0.8)(1.4)\big) = 1.231$

$x_2 = x_1(2 - (0.8)x_1) = (1.232)\big(2 - (0.8)(1.232)\big) = 1.2497408$

$x_3 = x_2(2 - (0.8)x_2) = (1.2497408)\big(2 - (0.8)(1.2497408)\big) = 1.249999946$

$x_4 = x_3(2 - (0.8)x_3) = (1.249999946)\big(2 - (0.8)(1.249999946)\big) = 1.25$

$x_5 = x_4(2 - (0.8)x_4) = (1.25)\big(2 - (0.8)(1.25)\big) = 1.25$

Thus it converge in only four iterations.

Advantages of the Newton Method

a) Converges faster than other methods, if it converges at all.

b) Requires only one guess

c) Allows for finding complex roots of an equation

d) The error decreases rapidly with each iteration

Disadvantages (Drawbacks) of the Newton Method

a) Unfortunately, for bad choices of x_0 (the initial guess) the method can fail to converge! Therefore the choice of x_0 is very important!

b) Diverges at inflection points: Selection of the initial guess or an iteration value of the root that is close to the inflection (concavity change) point of the function may start diverging away from the root in the Newton-Raphson method.

c) Oscillations near local maximum and minimum: Results obtained from the Newton-Raphson method may oscillate about the local maximum or minimum without converging on a root but converging on the local maximum or minimum. Eventually, it may lead to division by a number close to zero and may diverge.

d) Root Jumping: In some cases where the function is oscillating and has a number of roots, one may choose an initial guess close to a root. However, the guesses may jump and converge to some other root.

Note: A good strategy for avoiding failure to converge would be to use the bisection method for a few steps (to give an initial estimate and make sure the sequence of guesses is going in the right direction) followed by Newton's method, which should converge very fast at this point.

2.5.1. Multiple roots and modified newton's method

A multiple root (double, triple, etc.) occurs where the function is tangent to the x axis. if **m** is the multiplicity of the root, then the modified newton method can be given by the formula

$$\boldsymbol{x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)}} \qquad \textbf{(9)}$$

A more general modified Newton-Raphson method for multiple roots let us define a new function $u(x) = \frac{f(x)}{f'(x)} \Rightarrow \boldsymbol{x_{n+1} = x_n - \frac{u(x_n)}{u'(x_n)}}$. a new function $u(x)$ Contains only single roots even though $f(x)$ may have multiple roots

$$\boldsymbol{x_{n+1} = x_n - \frac{f'(x_n)f(x_n)}{[f'(x_n)]^2 - f(x_n)f''(x_n)}} \qquad \textbf{(10)}$$

Example1: The function $f(x) = x^5 - 11x^4 + 46x^3 - 90x^2 + 81x - 27 = (x-1)^2(x-3)^3$ has root of multiplicity 2 at $x = 1$ and 3 at $x = 3$ use modified newton's method to approximate the roots with $x_0 = 10$, and allowable tolerance of 0.000001. (Exercise)

Example2: The function $\boldsymbol{f(x) = x^3 - 3x + 1 = (x+2)(x-1)^2}$ has root of multiplicity 2 at $x = 1$ and 1 at $x = -2$ use modified newton's method to approximate the roots with $x_0 = 1.3$, and allowable tolerance of 0.000001. (Exercise)

Example3: The function $\boldsymbol{f(x) = x^3 - 7x^2 + 11x - 5}$ has root of multiplicity 2 at $x = 1$. use modified newton's method to approximate the roots with $x_0 = 0$, and allowable tolerance of 0.000001. Again use newton's method to approximate the roots with $x_0 = 0$, and allowable tolerance of 0.000001. Compare the two approximated roots.

Solution: for modified newton's methods $\boldsymbol{f'}(x) = 3x^2 - 14x + 11$, multiplicity (m=2), then $x_{n+1} = x_n - 2\frac{x^3{}_n - 7x^2{}_n + 11x_n - 5}{3x^2{}_n - 14x_n + 11}$ see table 5.1 for the iteration.

iter	**X**	**f(x)**	**df(x)**	**$\|x_{n+1}-x_n\|$**
0	0.000000	-5.000000	11.000000	
1	0.909091	-0.033809	0.752066	0.909091
2	0.999001	-0.000004	0.007995	0.089910

3	1.000000	-0.000000	0.000001	0.000999
4	1.000000	0.000000	0.000000	0.000000

The root of the equation at 4th iteration is 1 (root =1.0000).

Table 5.1 solution of $x^3 - 7x^2 + 11x - 5 = 0$ using modified newton's method $x_0 = 0$ and m=2.

For newton's method $x_{n+1} = x_n - \frac{x^3{}_n - 7x^2{}_n + 11x_n - 5}{3x^2{}_n - 14x_n + 11}$ see table 5.2 for the iterations.

iter	X	f(x)	df(x)	$\lvert x_{n+1} - x_n \rvert$
0	0.000000	-5.000000	11.000000	
1	0.454545	-1.352367	5.256198	0.454545
2	0.711835	-0.356084	2.554434	0.257290
3	0.851234	-0.091818	1.256523	0.139399
4	0.924307	-0.023352	0.622734	0.073073
5	0.961805	-0.005891	0.309935	0.037498
6	0.980813	-0.001480	0.154603	0.019008
7	0.990384	-0.000371	0.077209	0.009571
8	0.995186	-0.000093	0.038582	0.004802
9	0.997592	-0.000023	0.019285	0.002406
10	0.998795	-0.000006	0.009641	0.001204
11	0.999398	-0.000001	0.004820	0.000602
12	0.999699	-0.000000	0.002410	0.000301
13	0.999849	-0.000000	0.001205	0.000151
14	0.999925	-0.000000	0.000602	0.000075
15	0.999962	-0.000000	0.000301	0.000038
16	0.999981	-0.000000	0.000151	0.000019

The root of the equation at 16th iteration is 1 (root = 1.0000).

Table 5.2 the solution of $x^3 - 7x^2 + 11x - 5 = 0$ using newton's method $x_0 = 0$ and m=1.

Note that it took 16 iterations for newton's method to converge and only 4 iterations for the modified newton's method with the same initial guess and tolerance.

2.6. The Secant Method

Most nonlinear functions can be approximated by a set of straight lines over small intervals. The secant method reduces finding a root of a nonlinear equation to finding a point where some linear equation determined by the approximating line equation has a root as close as possible to the root we are looking for and it requires two initial estimates of x: x_0, x_1.

Geometric interpretation of secant method

- x_2 is the x-intercept of the secant line passing through $(x_0, f(x_0))$ and $(x_1, f(x_1))$.

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

- x_3 is the *x*-intercept of the secant-line passing through $(x_1, f(x_1))$ and $(x_2, f(x_2))$.

and so on. That is why the method is called the secant method.

Generally secant method can be expressed as

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

$$\Longrightarrow x_{n+1} = \frac{\begin{vmatrix} x_n & x_{n-1} \\ f(x_n) & f(x_{n-1}) \end{vmatrix}}{f(x_{n-1}) - f(x_n)} \qquad (11)$$

This is of course, the same formula as for the method of false position.

Algorithm for secant method

Let x_0 and x_1 be two initial approximations, For $n = 1,2, \ldots$,maximum iterations

$$x_{n+1} = \frac{\begin{vmatrix} x_n & x_{n-1} \\ f(x_n) & f(x_{n-1}) \end{vmatrix}}{f(x_{n-1}) - f(x_n)}$$

A suitable stopping criteria is

$|f(x_n)| \leq \varepsilon$, $|x_{n+1} - x_n| \leq \varepsilon$ or $\frac{|x_{n+1} - x_n|}{x_{n+1}} \leq \varepsilon$ where ε is a specified tolerance value.

Example 1: use the secant method with $x_0 = 1$ and $x_1 = 2$ to solve $f(x) = x^3 - x^2 - 1 = 0$ to within the tolerance 10^{-4}.

Solution: With $x_0 = 1$, $f(x_0) = -1$ and $x_1 = 2$, $f(x_1) = 3$, we have

$x_2 = \frac{\begin{vmatrix} x_1 & x_0 \\ f(x_1) & f(x_0) \end{vmatrix}}{f(x_0) - f(x_1)} = \frac{\begin{vmatrix} 2 & 1 \\ 3 & -1 \end{vmatrix}}{-1-3} = 1.25$, From which $f(x_2) = f(1.25) = -0.609375$. the next iterate is $x_3 = \frac{\begin{vmatrix} x_2 & x_1 \\ f(x_2) & f(x_1) \end{vmatrix}}{f(x_1) - f(x_2)} = \frac{\begin{vmatrix} 1.25 & 1 \\ -0.609375 & 3 \end{vmatrix}}{3-(-0.609375)} = 1.3766234$.

Continuing this process the root converges to $(r = 1.4655713)$, See table 6 below

iter	x_n	$f(x_n)$	$f(x_{n+1})$-$f(x_n)$	$\lvert x_{n+1}-x_n \rvert$
0	1.000000	-1.000000		
1	1.250000	-0.609375	-3.609375	0.750000
2	1.376623	-0.286264	0.323111	0.126623
3	1.488807	0.083463	0.369727	0.112184
4	1.463482	-0.007322	-0.090786	0.025325
5	1.465525	-0.000163	0.007160	0.002043
6	1.465571	0.000000	0.000163	0.000046

Table 6 solution of $x^3 - x^2 - 1 = 0$ using secant method with $x_0 = 1$ and $x_1 = 2$.

Hence, the approximate root is 1.4656, at the seventh iteration.

Advantages of secant method

1. It converges at faster than a linear rate, so that it is more rapidly convergent than the bisection method.

2. It does not require use of the derivative of the function, as newton's method.

3. It requires only one function evaluation per iteration, as compared with Newton's method which requires two.

Disadvantages of secant method

1. It may not converge.

2. The method fails to converge when $f(x_n) = f(x_{n-1})$

3. There is no guaranteed error bound for the computed iterates.

4. It is likely to have difficulty if $f'(\alpha) = 0$. This means the x-axis is tangent to the graph of $f(x)$ at $x = \alpha$, where α is a root of $f(x)$.

2.7. Conditions for convergence (Convergence of the Iteration Methods)

We now study the rate at which the iteration methods converge to the exact root, if the initial approximation is sufficiently close to the desired root. Define the error of approximation at the n^{th} iterations for $n = 0, 1, 2, \ldots$as

$$\varepsilon_n = x_n - \alpha \tag{12}$$

Definition An iterative method is said to be of *order* p or has the *rate of convergence* p, if p is the largest positive real number for which there exists a finite constant $C \neq 0$, such that

$$|\varepsilon_{n+1}| \leq C|\varepsilon_n|^p \tag{13}$$

The constant C, which is independent of n is called the *asymptotic error constant* and it depends on the derivatives of $f(x)$ at $x = \alpha$.

Let us now obtain the orders of the methods that were derived earlier.

Method of false position

Substituting $x_n = \varepsilon_n + \alpha,\ x_{n+1} = \varepsilon_{n+1} + \alpha,\ x_0 = \varepsilon_0 + \alpha$, we expand each term in Taylor's Series and simplify using t he fact that $f\ (\alpha) = 0$. we obtain the error equation as

$\varepsilon_{n+1} = C\varepsilon_0\varepsilon_n$, where $C = \frac{f''(\alpha)}{2f'(\alpha)}$ Since ε_0 is finite and fixed, the error equation becomes

$|\ \varepsilon_{n+1}\ | = |\ C*\ |\,|\ \varepsilon_n\ |, where\ C* = C\varepsilon_0.$

Hence, the method of false position has order 1 or has linear rate of convergence.

Method of successive approximations or fixed point iteration method

We have $x_{n+1} = g(x_n)$, and $\alpha = g(\alpha)$ Subtracting, we get

$x_{n+1} - \alpha = g(x_n) - g(\alpha) = g(\alpha + x_n - \alpha) - g(\alpha) = [g(\alpha) + (x_n - \alpha)g'(\alpha) + \cdots] - g(\alpha)$

or $\varepsilon_{n+1} = \varepsilon_n g'^{(\alpha)} + O(\varepsilon_n)$.

Therefore, $|\,\varepsilon_{n+1}| = C|\,\varepsilon_n|,\ x_n < t_n < \alpha, and\ C = |\,g'(\alpha)\,|$.

Hence, the fixed point iteration method has order 1 or has linear rate of convergence.

Newton-Raphson method

The method is given by, $x_{n+1} = x_n - \frac{f\,(x_n\,)}{f'(x_n)}, f'(x_n) \neq 0$. Substituting $x_n = \varepsilon_n + \alpha,\ x_{n+1} = \varepsilon_{n+1} + \alpha,$ we obtain $\varepsilon_{n+1} + \alpha = \varepsilon_n + \alpha - \frac{f\,(\varepsilon_n+\alpha\,)}{f'(\varepsilon_n+\alpha)}$. Expand the terms in Taylor's series. Using the fact that $f(\alpha) = 0$, and canceling $f\,'(\alpha)$, we Obtain

$$\varepsilon_{n+1} = \varepsilon_n - \frac{\left[\varepsilon_n f'(\alpha) + \frac{1}{2}\varepsilon_n{}^2 f''(\alpha) + \cdots\right]}{f'(\alpha) + \varepsilon_n f''(\alpha)}$$

$$= \varepsilon_n - \left[\varepsilon_n + \frac{f''(\alpha)}{2f'(\alpha)}\varepsilon_n{}^2 + \cdots\right]\left[1 + \frac{f''(\alpha)}{f'(\alpha)}\varepsilon_n + \cdots\right]^{-1}$$

$$= \varepsilon_n - \left[\varepsilon_n + \frac{f''(\alpha)}{2f'(\alpha)}\varepsilon_n{}^2 + \cdots\right]\left[1 - \frac{f''(\alpha)}{f'(\alpha)}\varepsilon_n + \cdots\right]$$

$$= \varepsilon_n - \left[\varepsilon_n - \frac{f''(\alpha)}{2f'(\alpha)}\varepsilon_n{}^2 + \cdots\right] = \frac{f''(\alpha)}{2f'(\alpha)}\varepsilon_n{}^2 + \cdots$$

Neglecting the terms containing $\varepsilon_n{}^3$ and higher power of ε_n, we get

$\varepsilon_{n+1} = C\varepsilon_n{}^2, where\ \ C = \frac{f''(\alpha)}{2f'(\alpha)}$ and $|\varepsilon_{n+1}| = |C||\varepsilon_n|^2$

Therefore, Newton's method is of order 2 or has quadratic rate of convergence.

Or we can also see in a similar way as

$$f(x) = f(x_n) + (x - x_n)f'(x_n) + \frac{1}{2}(x - x_n)f''(\eta)$$

$$f(x^*) = 0 = f(x_n) + (x^* - x_n)f'(x_n) + \frac{1}{2}(x^* - x_n)^2 f''(\eta)$$

$\Longrightarrow x^* = x_n - \frac{f(x_n)}{f'(x_n)} - (x^* - x_n)^2\frac{f''(\eta)}{2f'(x_n)}$ But $x_n - \frac{f(x_n)}{f'(x_n)} = x_{n+1}$

$\frac{x^* - x_{n+1}}{(x^* - x_n)^2} = \frac{-f''(x^*)}{2f'(x^*)} = \frac{\varepsilon_{n+1}}{\varepsilon_n{}^2}$ Quadratic convergence

Remark: *What is the importance of defining the order or rate of convergence of a method*?

Suppose that we are using Newton's method for computing a root of $f(x) = 0$. Let us assume that at a particular stage of iteration, the error in magnitude in computing the root is $10^{-1} = 0.1$. We observe from Newton's method which has quadratic rate of convergence, that in the next iteration, the error behaves like $C(0.1)^2 = C(10^{-2})$.

That is, we may possibly get an accuracy of two decimal places. Because of the quadratic convergence of the method, we may possibly get an accuracy of four decimal places in the next iteration. However, it also depends on the value of *C*. From this discussion, we conclude that both fixed point iteration and regula-falsi methods converge slowly as they have only linear rate of convergence. Further, Newton's method converges at least twice as fast as the fixed point iteration and regula-falsi methods.

```
>> f=@(x)x.^2+4*x-10; df=@(x)2*x+4; x0=1; tol=0.001; n=5;
>> root= newton1(f,df,x0,tol,n)
--------------------------------------------------------------
iter    x          f(x)        df(x)        |xn+1-xn|
--------------------------------------------------------------
 0    1.000000   -5.000000    6.000000
 1    1.833333    0.694444    7.666667    0.833333
 2    1.742754    0.008205    7.485507    0.090580
 3    1.741658    0.000001    7.483315    0.001096
 4    1.741657    0.000000    7.483315    0.000000
the root of the equation at 4th iteration is
root =

    1.7417

>> f=@(x)x.^2+4*x-10; a=1; b=2; tol=0.001; n=5;
>> root= false1(f,a,b,tol,n)
--------------------------------------------------------------
iter    a         b         c         f(c)     |b-a|
--------------------------------------------------------------
```

```
 0      1.0000      2.0000     1.714286  -0.204082   1.000000
 1      1.7143      2.0000     1.740741  -0.006859   0.285714
 2      1.7407      2.0000     1.741627  -0.000229   0.259259
the root of the equation at 3th iteration is
root =

    1.7417

>> f=@(x)x.^2-2; df=@(x)2*x; x0=1; tol=0.001; n=5;
>> root= newton1(f,df,x0,tol,n)
--------------------------------------------------------------
iter     x           f(x)         df(x)         |xn+1-xn|
--------------------------------------------------------------
 0    1.000000    -1.000000     2.000000
 1    1.500000     0.250000     3.000000     0.500000
 2    1.416667     0.006944     2.833333     0.083333
 3    1.414216     0.000006     2.828431     0.002451
 4    1.414214     0.000000     2.828427     0.000002
the root of the equation at 4th iteration is
root =

    1.4142

>> f=@(x)x.^2-2; a=1; b=1.5; tol=0.001; n=5;
>> root= false1(f,a,b,tol,n)
--------------------------------------------------------------
iter     a           b           c           f(c)      |b-a|
--------------------------------------------------------------
 0      1.0000      1.5000     1.400000  -0.040000   0.500000
 1      1.4000      1.5000     1.413793  -0.001189   0.100000
 2      1.4138      1.5000     1.414201  -0.000035   0.086207
```

the root of the equation at 3th iteration is

root =

1.4142