Roots of Non Linear Equation

Mathematical models for a wide variety of problems in science and engineering can be formulated into equations of the form: $f(x) = 0$ .... (i)
where x and f (x) may be real, complex or vector quantities. The values of x for which the equation (i) satisfy are called the *roots* of the equation.

Equation (i) may belong to one of the following types of equations:
i) Algebraic equation ii) Polynomial equation iii) Transcendental equations

- y = f(x) is a *linear* function, if the dependent variable y changes in direct proportion to the change in independent variable x. For example, y = 3x + 5 is a linear equation.
- y = f(x) is said to be *nonlinear*, if the response of the dependent variable y is not direct or exact proportion to the changes in the independent variable x. For example, $y = x^2 + 1$ is a non linear equation.
- Any function of one variable which does not graph as a straight line in two dimensions, or any function of two variables which does not graph as a plane in three dimensions, can be said as *nonlinear*.

Algebraic Equation
An equation of type $y = f(x)$ is said to be *algebraic* if it can be expressed in the form

$$f_n.y_n + f_{n-1}y_{n-1} + \dots\dots\dots + f_1y_1 + f_0 = 0 \quad \dots\dots\dots\dots\text{(ii)}$$

where f_i is an *i*th order polynomial in x. Equation (ii) can be written as general form as,

$$f(x, y) = 0$$

- Some examples are:
  1. $3x + 5y – 2 = 0$ (linear)
  2. $2x + 3xy -25 = 0$ (non-linear)
  3. $x^3 – xy + 3y^3 = 0$ (non-linear)

  These equations have an infinite number of pairs of values and x and y which satisfy them.

Polynomial Equation
Polynomial equations are a simple class of algebraic equations that are represented as follows:

$$a_n x^n + a_{n-1} x^{n-1} + \dots\dots\dots + a_1 x + a_0 = 0$$

This is called n^{th} degree polynomial and has *n* roots. The roots may be:
i) real and different ii) real and repeated iii) complex numbers

- Since a complex roots appear in pairs, if n is odd, then the polynomial has at least one real root.
- Some specific examples of polynomial equations are:
  1. $5x^5 – x^3 + 3x^2 = 0$
  2. $x^3 – 4x^2 + x + 6 = 0$

Transcendental Equations
A non-algebraic equation is called a *transcendental* equation. These include trigonometric, exponential and logarithmic functions.

- Examples of transcendental equation are:
  1. $2 \sin x – x = 0$
  2. $e^x \sin x – \log x = 0$
- A transcendental equation may have a finite or an infinite number of real roots or may not have real root at all.

Methods of Solution

There are a number of ways to find the roots of non linear equation such as:

1. Direct analytical methods
2. Graphical methods
3. Trial and error methods
4. Iterative methods

Direct analytical methods

In certain cases, roots can be found by using direct analytical methods. For example, we can easily calculate the roots of a quadratic equation: $ax^2 + bx + c = 0$ from the following equation:

$$x = (- b \pm \sqrt{(b^2 - 4ac)}) / (2a)$$

However, there are equations that cannot be solved by analytical methods. For example, the simple transcendental equation $2 \sin x - x = 0$ cannot be solved analytically. Direct methods for solving non-linear equations do not exist except for certain simple cases.

Graphical methods

Graphical methods are useful when we are satisfied with approximate solution for a problem. This method involves plotting the given function and determining the points where it crosses x-axis. These points represent approximate values of the roots of the function.

Trial & Error methods

This method involves a series of guesses for x, each time evaluating the function to see whether it is close to zero. The value of x that causes the function value closer to zero is one of the approximate roots of the equation.

- Although graphical and trial error methods provide satisfactory approximations for many problem situations, they become cumbersome and time consuming. Moreover, the accuracy of the results are inadequate for the requirements of many engineering and scientific problems.

Iterative methods:

An iterative technique usually begin with an approximate value of the root, known as *initial guess*, which is then successively corrected iteration by iteration. The process of iteration stops when the desired level of accuracy is obtained.

- Iterative methods can be divided into two categories:
  1. *Bracketing methods*: Bracketing methods (also known as *interpolation methods*) start with initial guesses that 'bracket' the root and then systematically reduce the width of the bracket until the solution is reached. Two popular method under this category are:
     - Bisection method
     - False position method
  2. *Open end methods*: Open end methods (also known as *extrapolation methods*) use a single starting value or two values that do not necessarily bracket the root. The following iterative methods fall under this category:
     - Newton-Raphson method
     - Secant method.
     - Muller's method.
     - Methods of successive approximation (Fixed-point method)

Largest Possible Root

For a polynomial represented by: $f(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0$ the largest possible root is given by: $x_{\cdot} = -\dfrac{a_{n-1}}{a_n}$.

- This value is taken as initial approximation when no other value is suggested by the knowledge of the problem at hand.

Search Bracket

Maximum absolute value of the root is,

$$| x_{\max} | = \sqrt{\{(\frac{a_{n-1}}{a_n})^2 - 2(\frac{a_{n-2}}{a_n})\}}$$

So, all real roots lie within the interval (- | x $_{\max}$ |, | x $_{\max}$ |).

Example: Consider the polynomial equation: $2x^3 - 8x^2 + 2x + 12 = 0$. Estimate the possible initial guess values.

Solution: The largest possible root is x = - (-8 / 2) = 4. No root can be larger than the value 4.

All roots must satisfy the relation

$$| x_{\max} | <= \sqrt{\{(\frac{-8}{2})^2 - 2.\frac{2}{2}\}} = \sqrt{14}$$

Therefore, all real roots lie in the interval (-$\sqrt{14}$, $\sqrt{14}$). We can use these two points as initial guesses for the bracketing methods and one of them as the open end methods.

Stopping Criterion

An iterative process must be terminated at some stage. We may use one (or combination) of the following tests, depending on the behavior of the function, to terminate the process:

1. $| x_{i+1} - x_i |$ $\le E_a$ (absolute error in x)
2. $| \dfrac{x_{i+1} - x_i}{x_{i+1}} |$ $\le E_r$ (relative error in x), $x \neq 0$
3. $| f(x_{i+1}) |$ $\le E$ (value of function at root)
4. $| f(x_{i+1}) - f(x_i) |$ $\le E$ (difference in function values)
5. $| f(x) |$ $\le F_{max}$ (large function value)
6. $| x_i |$ $\le XL$ (large value of x)

Here x_i represents the estimate of the root at i^{th} iteration and $f(x_i)$ is the value of the function at x_i.

- There may be situations where these tests may fail when used alone. Sometimes even a combination of two tests may fail. A practical convergence test should use a combination of these tests.
- In cases where we do not know whether the process converges or not, we must have a limit on the number of iterations, like Iterations $\ge$ N (limit on iterations).

Evaluation of Polynomials

The polynomial is a sum of n+1 terms and can be expressed as

$$f(x) = \sum_{i=0}^{n} a_i x^i = a_0 + \sum_{i=1}^{n} a_i x^i$$

This can be easily implemented using a loop. But this would require n(n+1)/2 multiplication and n additions.

Horner's Rule

We can write the polynomial $f(x)$ using *Horner's rule* (also known as *nested multiplication*) as follows:

$$f(x) = ((\ldots\ldots((a_n x + a_{n-1})x + a_{n-2})x + \ldots\ldots\ldots + a_1)x + a_0)$$

Here the innermost expression $a_n x + a_{n-1}$ is evaluated first. The resulting value constitutes a multiplicand for the expression at the next level. The number of level equals n, the degree of polynomial. This approach needs a total of n additions and n multiplications.

Algorithm: **Horner's Rule**

$p_n = a_n$
$p_{n-1} = p_{n * x} + a_{n-1}$
. . .
. . .
. . .
$p_j = p_{j+1}\, x + a_j$
. . .
. . .
$p_1 = p_2\, x + a_1$

$f(x) = p_0 = p_1\, x + a_0$

***Example**:* Evaluate the polynomial $f(x) = x^3 - 4x^2 + x + 6$ using Horner's rule at x = 2.

Solution: Here $n = 3$, $a_3 = 1$, $a_2 = -4$, $a_1 = 1$, $a_0 = 6$

$p_3 = a_3 = 1$
$p_2 = 1 * 2 + (-4) = -2$
$p_1 = (-2) * 2 + 1 = -3$
$p_0 = (-3) * 2 + 6 = 0$

$f(2) = 0$

Bisection method

The *bisection method* (also known as *binary chopping* or *half-interval method*) is one of the simplest and most reliable of iterative methods for the solution of nonlinear equations. This method based on the repeated application of the *intermediate value theorem*.

- **Intermediate Value Theorem**: If f (x) is a continuous function is some interval [a, b] and f (a) and f (b) are of opposite signs, then the equation $f(x) = 0$ has at least one real root or an odd number of real roots in (a, b).

This theorem is evident from the Figure-6.2 [page-132, Balagurusamy] that if $f(x_1)$ and $f(x_2)$ have opposite signs the graph must cross the x-axis at least once between $x = x_1$ and $x = x_2$.

- Suppose f (x) has a real root in the interval [a, b]. The midpoint between a and b is
  $x_0 = (a+b)/2$.

  Now, there exist the following three conditions:
  1. if $f(x_0) = 0$, then x_0 is the root of the equation $f(x) = 0$.
  2. if $f(x_0) * f(a) < 0$, then the root lies between x_0 and a.
  3. if $f(x_0) * f(b) < 0$, then the root lies between x_0 and b.

  It follows that by testing the sign of the function at midpoint, we can deduce which part of the interval contains the root. We can further divide this subinterval into two halves to locate a new subinterval containing the root. This process can be repeated until the interval containing the root is as small as we desire.

Algorithm: Bisection method

1. Decide initial values for x_1 and x_2 and stopping criterion E.
2. Compute $f_1 = f(x_1)$ and $f2 = f(x_2)$.
3. If $f_1 * f_2 > 0$, x_1 and x_2 do not bracket any root and go to step 1.
4. Compute $x_0 = (x_1 + x_2) / 2$ and compute $f_0 = f(x_0)$.
5. If $f_1 * f_0 < 0$ then set $x_2 = x_0$ else set $x_1 = x_0$.
6. If absolute value of $(x_2 - x_1)$ is less then E, then root = $(x_1 + x_2) / 2$ and go to step 7
   Else go to step 4
7. Stop.

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

Solution: Since $f(1)$ is negative and $f(2)$ is positive a root lies between 1 and 2.

Hence, $x_0 = (1 + 2) / 2 = 1.5$.

Now, $f(x_0) = 7/8$, which is positive. Hence, the root lies between 1 and 1.5

Now, $x_1 = (1 + 1.5) / 2 = 1.25$ and $f(x_1) = -19/64$, which is negative.

Hence, the root lies between 1.25 and 1.5.

It follows that, $x_2 = (1.25 + 1.5)/2 = 1.375$

Repeating the procedure, we have the following successive approximation

$x_3 = 1.3125$

$x_4 = 1.3475$

$x_5 = 1.328125$ etc

Example 6.4: Page-132, Balagurusamy.

- The bisection method is *linearly convergent*. Since the convergence is slow to achieve a high degree of accuracy, a large number of iterations may be needed. However, the bisection method is guaranteed to converge.

False Position Method

In bisection method, the interval between x_1 and x_2 divided into two equal halves, irrespective of location of the root. It may be possible that the root is closer to one end than the other as shown in Figure 6.3 [page-138, Balagurusamy). In the figure, the root is closer to x_1. If we join the points x_1 and x_2 by straight line, the point of intersection of this line with the x axis (x_0) gives an improved estimate of the root and is called the *false position* of the root. This point then replaces one of the initial guesses that has a function value of the same sign as $f(x_0)$. The process is repeated with the new values of x_1 and x_2. Since this method uses the false position of the root repeatedly, it is called the *false position method* (or *regula falsi* in Latin). It is also called the *linear interpolation* method.

False Position Formula

The equation of the line joining the points $(x_1, f(x_1))$ and $(x_2, f(x_2))$ is given by

$$(f(x_2) - f(x_1)) / (x_2 - x_1) = (y - f(x_1)) / (x - x_1)$$

Since the line intersects the x-axis at x_0, when $x = x_0$, $y = 0$, we have,

$$(f(x_2) - f(x_1)) / (x_2 - x_1) = (-f(x_1)) / (x_0 - x_1)$$

or $$x_0 - x_1 = -f(x_1)(x_2 - x_1) / (f(x_2) - f(x_1))$$

Then we have,

$$x_0 = x_1 - (f(x_1)(x_2 - x_1)) / (f(x_2) - f(x_1))$$

This equation is known as the *false position formula*.

False Position Algorithm

1. Decide initial values for x_1 and x_2 and stopping criterion E.
2. Compute $x_0 = x_1 - (f(x_1)(x_2 - x_1)) / (f(x_2) - f(x_1))$
3. If $f(x_0) * f(x_1) < 0$ set $x_2 = x_0$ otherwise set $x_1 = x_0$
4. If the absolute difference of two successive x_0 is less then E, then root = x_0 and stop. Else go to step 2.

- False position method converges *linearly*.

Example: Use the false position method to find a root of the function $f(x) = x^2 - x - 2 = 0$ in the range $1 < x < 3$.

Solution: Given $x_1 = 1$ and $x_2 = 3$

$$f(x_1) = f(1) = -2 \qquad f(x_2) = f(3) = 4$$

$$x_0 = x_1 - (f(x_1)(x_2 - x_1)) / (f(x_2) - f(x_1))$$
$$= 1 + 2 * (3 - 1) / (4 + 2)$$
$$= 1.6667$$

Iteration 2: $f(x_0) * f(x_1) = f(1.6667) * f(1) = 1.7778$

Therefore, the root lies in the interval between x_0 and x_2. Then,

$$x_1 = x_0 = 1.6667$$
$$f(x_1) = f(1.6667) = -0.8889$$
$$f(x_2) = f(3) = 4$$
$$x_0 = 1.6667 + 0.8889 * (3 - 1.6667) / (4 + 0.8889) = 1.909$$

Iteration 3: $f(x_0) * f(x_1) = f(1.909) * f(1.6667) = 0.2345$

Therefore, the root lies in the interval between $x_0 = 1.909$ and $x_2 = 3$. Then,

$$x_1 = x_0 = 1.909$$
$$x_0 = 1.909 - 0.2647 * (3 - 1.909) / (4 - 0.2647) = 1.986$$

The estimated root after third iteration is 1.986. The interval contains a root x = 2. We can perform additional iterations to refine this estimate further.

Fixed Point method

To find the root of the equation $f(x) = 0$, we rewrite this equation in this way

$$x = g(x)$$

Let x_0 be an approximate value of the desire root. Substituting it for x as the right side of the equation, we obtain the first approximation $x_1 = g(x_0)$. Further approximation is given by $x_2 = g(x_1)$. This iteration process can be expressed in general form as

$$x_{i+1} = g(x_i) \quad i = 0, 1, 2, \ldots.$$

which is called the *fixed point iteration formula*. The iteration process would be terminated when two successive approximations agree within some specified error.

- This method of solution is also known as the *method of successive approximations* or *method of direct substitution*.

Example: Locate the root of the equation $f(x) = x^3 + x^2 - 1 = 0$.
Solution: The given equation can be expressed as $x = 1 / \sqrt{(x+1)}$.
Let us start with an initial value of $x_0 = 1$.

$x_1 = 0.7071 \quad x_2 = 0.7654 \quad x_3 = 0.7526 \quad x4 = 0.7554 \ldots$

Example 6.11 & 6.12: Page – 161-62, Balagurusamy.

- The iteration function g (x) can be formulated in different forms. But, not all forms result in convergence of solution. Convergence of the iteration process depends on the nature of g (x). The necessary condition for convergence is $g'(x) < 1$.

Newton-Raphson Method

Let x_0 be an approximate root of $f(x) = 0$ and let $x_1 = x_0 + h$ be the correct root so that $f(x_1) = 0$. Expanding $f(x_0 + h)$ by Taylor's series, we obtain

$$f(x_0) + h f'(x_0) + h^2/2!\ f''(x_0) + \ldots.. = 0$$

Neglecting the second and higher order derivatives, we have

$$f(x_0) + h f'(x_0) = 0$$

which gives $h = -f(x_0) / f'(x_0)$

A better approximation than x0 is therefore given by x1, where

$$x_1 = x_0 - f(x_0) / f'(x_0)$$

Successive approximations are given by $x_2, x_3, x_4, \ldots, x_{n+1}$, where

$$X_{n+1} = x_n - f(x_n) / f'(x_n)$$

which is called the *Newton-Raphson formula*.

The process will be continued till the absolute difference between two successive approximations is less than the specified error E i,e, $| x_{n+1} - x_n | < E$.

Geometrical interpretation of Newton-Raphson Method

Let us consider the curve $y = f(x)$. Let this curve meets x-axis at $(\varepsilon, 0)$. Thus ε is the solution of the equation $f(x) = 0$. We have to find the value of x near to ε. [Fig]

Let us take a point $(x_n, f(x_n))$ on $y = f(x)$ where x_n is not far from ε. Let us draw a tangent to the curve $y = f(x)$ at the point $(x_n, f(x_n))$. The equation of the tangent is

$$y - f(x_n) = m (x - x_n)$$

where $m = [d/dx(f(x))]_{x = xn} = f'(x_n)$ is the slope of the tangent.

Hence the equation of the tangent is

$$y - f(x_n) = f'(x_n)(x - x_n)$$

If this tangent meets x-axis at $(x_{n+1}, 0)$ then

$$0 - f(x_n) = f'(x_n)(x_{n+1} - x_n)$$

Hence, $x_{n+1} = x_n - f(x_n) / f'(x_n)$

Algorithm: Newton-Raphson Method

1. Assign an initial value for x, say x_0 and stopping criterion E.
2. Compute $f(x_0)$ and $f'(x_0)$.
3. Find the improved estimate of x_0
   $x_1 = x_0 - f(x_0) / f'(x_0)$
4. Check for accuracy of the latest estimate.
   If $|x_1 - x_0| < E$ then stop; otherwise continue.
5. Replace x_0 by x_1 and repeat steps 3 and 4.

- Newton-Raphson method is said to have *quadratic convergence*.

Example: Find the root of the equation $f(x) = x^2 - 3x + 2$ using Newton-Raphson method.
Solution: Here, $f'(x) = 2x - 3$
Let $x_1 = 0$ (First approximation)
$x_2 = x_1 - f(x_1) / f'(x_1) = 0 - 2 / (-3) = 2/3 = 0.6667$
Similarly,
$x_3 = 0.6667 - 0.4444/-1.6667 = 0.9333$
$x_4 = 0.9333 - 0.071/-1.334= 0.9959$
$x_5 = 0.6667 - 0.0041/-1.0082 = 0.9999$
$x_6 = 0.6667 - 0.0001/-1.0002 = 1.0000$
Since $f(1.0) = 0$, The root closer to the point x = 0 is 1.0000.

Limitations of Newton-Raphson method
The Newton-Raphson method has certain limitations and pitfalls. The method will fail in the following situations.

1. Division by zero may occur if $f'(x_i)$ is zero or very close to zero.
2. If the initial guess is too far away from the required root, the process may converge to some other root.
3. A particular value in the iteration sequence may repeat, resulting in an infinite loop. This occurs when the tangent to the curve f (x) at $x = x_{i+1}$ cuts the x-axis again at $x = x_i$.

Secant method

Secant method, like the False Position & Bisection methods, uses two initial estimates but does not require that they must bracket the root.
Let us consider two points x_1 and x_2 in Fig 6.6 [page-152, Balagurusamy] as starting values. Slope of the secant line passing through x_1 and x_2 is given by,

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

$$f(x_1)(x_2 - x_3) = f(x_2)(x_1 - x_3)$$

$$x_3[f(x_2) - f(x_1)] = f(x_2).x_1 - f(x_1).x_2$$

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

By adding and subtracting $f(x_2).x_2$ to the numerator and remaining the terms we have,

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

which is called the *secant formula*.
The approximate value of the root can be refined by repeating this procedure by replacing x_1 and x_2 by x_2 and x_3, respectively. That is, the next approximate value is given by,

$$x_4 = x_3 - \frac{f(x_3)(x_3 - x_2)}{f(x_3) - f(x_2)}$$

This procedure is continued till the desired level of accuracy is obtained.
We can express the Secant formula as follows:

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

- The formula for false position method and secant method are similar and both of them use two initial estimates. However there is a major difference in their algorithms of implementation. In false position method, the last estimate replaces one of the end points of the interval such that new interval brackets the root. But, in secant method, the values are prefaced in strict sequence, i.e. x_{i-1} is replaced by x_i and x_i by x_{i+1}. The points may not bracket the root.

Algorithm: **Secant Method**

1. Decide two initial points x_1 and x_2 and required accuracy level E.
2. Compute $f_1 = f(x_1)$ and $f_2 = f(x_2)$
3. Compute $x_3 = (f_2 x_1 - f_1 x_2) / (f_2 - f_1)$
4. If $|x_3 - x_2| > E$, then
   set $x_1 = x_2$ and $f_1 = f_2$
   set $x_2 = x_3$ and $f_2 = f(x_3)$
   go to step 3

   Else
   set root = x_3
   print results
5. Stop.

- ***Comparison between the secant formula and the Newton-Raphson formula for estimating a root***
  Newton-Raphson formula: $x_{n+1} = x_n - f(x_n)/f'(x_n)$
  Secant formula: $x_{n+1} = x_n - ((f(x_n) * (x_n - x_{n-1})) / (f(x_n) - f(x_{n-1})))$

This shows that the derivative of the function in the Newton formula $f'(x_n)$, has been replaced by the term $(f(x_n) - f(x_{n-1})) / (x_n - x_{n-1})$ in the secant formula.
This is a major advantage because there is no need for the evaluation of derivatives. There are many functions whose derivatives may be extremely difficult to evaluate.
However, one drawback of the secant formula is that the previous two iterates are required for estimating the new one. Another drawback of the secant method is its slower rate of convergence. It is proved that the rate of convergence of secant method is 1.618 while that of the Newton method is 2.

- The convergence of secant method is referred as *superlinear convergence.*

Example: Use the secant method to estimate the root of the equation $f(x) = x^2 - 4x - 10 = 0$ with the initial estimates of $x_1 = 4$ and $x_2 = 2$.
Solution: Given $x_1 = 4$ and $x_2 = 2$

$f(x_1) = f(4) = -10 \qquad f(x_2) = f(2) = -14$

$x_3 = x_2 - ((f(x_2) * (x_2 - x_1)) / (f(x_2) - f(x_1)))$
$= 2 - (-14) * (2 - 4)/((-14) - (-10))$
$= 9$

For second iteration,

$x_1 = x_2 = 2 \qquad x_2 = x_3 = 9$

$f(x_1) = f(2) = -14$ $f(x_2) = f(9) = 95$

$x_3 = 9 – 35 * (9 – 2) / (35 + 14) = 4$

For third iteration,

$x_1 = x_2 = 9$ $x_2 = x_3 = 4$

$f(x_1) = f(9) = 95$ $f(x_2) = f(4) = -10$

$x_3 = 4 – (-10) * (4 – 9) / ((-10) – (-35)) = 5.1111$

For fourth iteration,

$x_1 = x_2 = 4$ $x_2 = x_3 = 5.1111$

$f(x_1) = f(4) = -10$ $f(x_2) = f(5.1111) = - 4.3207$

$x_3 = 5.1111 – (-4.3207) * (5.1111 – 4) / ((- 4.3207) – (-10)) = 5.9563$

For fifth iteration,

$x_1 = x_2 = 5.1111$ $x_2 = x_3 = 5.9563$

$f(x_1) = f(5.1111) = - 4.3207$ $f(x_2) = f(5.9563) = 5.0331$

$x_3 = 5.9563– 5.0331 * (5.9563– 5.1111)/(5.0331 + 4.3207) = 5.5014$

For sixth iteration,

$x_1 = x_2 = 5.9563$ $x_2 = x_3 = 5.5014$

$f(x_1) = f(5.9563) = 5.0331$ $f(x_2) = f(5.5014) = -1.7392$

$x_3 = 5.5014– (-1.7392)* (5.5014– 5.9563) / (-1.7392- 5.0331) = 5.6182$

The value can be further refined by continuing the process, if necessary.

Determining All Possible Roots:

All the methods discussed so far estimate only one root. If we want to locate all the roots in the given interval then one option is to *plot a graph of the function* and then identify various independent intervals that bracket the roots. These intervals can be used to locate the various roots.

Another approach is to use an *incremental search* technique covering the entire interval containing the roots. This means that search for a root continues even after the first root is found. The procedure consists of starting at one end of the interval, say at point **a**, and then searching for a root at every incremental interval till the other end, say point **b**, is searched. [Fig-6.9, page-167, Balagurusamy]. The following algorithm describes the steps for implementing an incremental search technique using the bisection method for locating all roots.

Algorithm

1. Choose lower limit **a** and upper limit **b** of the interval covering all the roots.
2. Decide the size of the increment interval Δx
3. set $x_1 = a$ and $x_2 = x_1 + \Delta x$
4. Compute $f_1 = f(x_1)$ and $f_2 = f(x_2)$
5. If $(f_1 * f_2)>0$, then the interval does not bracket any root and go to step 9
6. Compute $x_0 = (x_1 + x_2)/2$ and $f_0 = f(x_0)$
7. If $(f_1 * f_2)<0$, then set $x_2 = x_0$
   Else set $x_1 = x_0$ and $f_1 = f_0$
8. If $| x_2 – x_1 | < E$, then
   root = $(x_1 - x_2) / 2$
   write the value of root
   go to step 9
   Else
   go to step 6
9. If $x_2 < b$, then set $a = x_2$ and go to step 3
10. Stop.

- A major problem is to decide the increment size. A small size may mean more iterations and more execution time. If the size is large, then there is a possibility of missing the closely spaced roots.

Roots of Polynomials

The number of real roots can be obtained using Descarte's rule of sign. This rule states that

1. The number of positive real roots is equal (or less than by an even integer) to the number of sign changes in the co-efficient of the equation.
2. The number of negative real roots is equal (or less than by an even integer) to the number of sign changes in the co-efficient if x is replaced by –x.

Deflation & Synthetic Division

A polynomial of degree n can be expressed as

$$p(x) = (x - x_r)\, q(x)$$

where x_r is a root of the polynomial p(x) and q(x) is the *quotient. Polynomial* of degree n-1. Once a root is found, we can use this fact to obtain a lower degree polynomial q(x) by dividing p(x) by $(x - x_r)$ using a process known as *Synthetic division*. The name "Synthetic" is used because the quotient polynomial q(x) is obtained without actually performing the division. The activity of reducing the degree of a polynomial is referred to as *deflation*.

The quotient polynomial q(x) can be used to determine the other roots of p(x), because the remaining roots of p(x) are the roots of q(x). When a root of q(x) is found, a further deflation can be performed and the process can be continued until the degree is reduced to one.
Synthetic division is preformed as follow:

Let, $p(x) = \sum_{i=0}^{n} a_i x^i$

$q(x) = \sum_{i=0}^{n-1} b_i x^i$

If $p(x) = (x - x_r)\ q(x)$, then

$$a_n x^n + a_{n-1} x^{n-1} + \ldots \ldots + a_1 x + a_0 = (x - x_r)(b_{n-1} x^{n-1} + b_{n-2} x^{n-2} + \ldots \ldots + b_1 x + b_0)$$

By comparing the coefficients of like powers of x on both the sides of equation, we get the following relation between them:

$a_n = b_{n-1}$

$a_{n-1} = b_{n-2} - x_r b_{n-1}$

. . .

. . .

$a_1 = b_0 - x_r b_1$

$a_0 = - x_r b_0$

That is $a_i = b_{i-1} - x_r b_i$ where, $b_n = 0$ and $i = n, n-1, \ldots\ldots 0$

Then $b_{i-1} = a_i + x_r b_i$ where, $b_n = 0$ and $i = n, n-1, \ldots\ldots 1$

The above Equation suggests that we can determine the co-efficient of q(x) [i.e, b_{n-1}, b_{n-2}, …….. , b_0] from the co-efficient of p(x) [i.e a_n, a_{n-1}, ………… , a_1] recursively. Thus we have obtained the polynomial q(x) without performing any division operation.

Example: The polynomial $p(x) = x^3 - 7x^2 + 15x - 9 = 0$ has a root at x = 3. Find the quotient polynomial q(x) such that p(x) = (x – 3) q(x).

Solution: Here, $a_3 = 1$, $a_2 = -7$, $a_1 = 15$, $a_0 = -9$

$b_3 = 0$

$b_2 = a_3 + b_3 * 3 = 1 + 0 = 1$

$b_1 = a_2 + b_2 * 3 = -7 + 3 = -4$

$b_0 = a_1 + b_1 * 3 = 15 + (-12) = 3$

Thus the polynomial q(x) is

$x^2 - 4x + 3 = 0$

Algorithm: Multiple Roots by Newton-Raphson Method

1. Obtain degree and co-efficient of polynomial (n and a_i).
2. Decide an initial estimate for the first root (x_0) and error criterion, E.
   Do while n >1
3. Find the root using Newton-Raphson algorithm
   $x_r = x_0 - f(x_0) / f'(x_0)$
4. Root (n) = x_r
5. Deflate the polynomial using synthetic division algorithm and make the factor polynomial as the new polynomial of order n-1.
6. Set $x_0 = x_r$ [Initial value of the new root]
   End of Do
7. Root (1) = $- a_0 / a_1$
8. Stop.