4.4 NUMERICAL ERRORS

Numerical errors (also known as *procedural errors*) are introduced during the process of implementation of a numerical method. They come in two forms, *roundoff errors* and *truncation errors*. The total numerical error is the summation of these two errors. The total error can be reduced by devising suitable techniques for implementing the solution. We shall see in this section the magnitude of these errors.

Roundoff Errors

Roundoff errors occur when a fixed number of digits are used to represent exact numbers. Since the numbers are stored at every stage of computation, roundoff error is introduced at the end of every arithmetic operation. Consequently, eventhough an individual roundoff error could be very small, the cumulative effect of a series of computations can be very significant.

Rounding a number can be done in two ways. One is known as *chopping* and the other is known as *symmetric rounding*. Some systems use the chopping method while others use symmetric rounding.

Chopping

In chopping, the extra digits are dropped. This is called *truncating* the number. Suppose we are using a computer with a fixed word length of four digits. Then a number like 42.7893 will be stored as 42.78, and the digits 93 will be dropped. We can express the number 42.7893 in floating point form as

$$
\begin{aligned}
x &= 0.427893 \times 10^2 \\
&= (0.4278 + 0.000093) \times 10^2 \\
&= [0.4278 + (0.93 \times 10^{-4})] \times 10^2
\end{aligned}
$$

This can be expressed in general form as

$$
\begin{aligned}
\text{True } x &= (f_x + g_x \times 10^{-d})10^E \\
&= f_x \times 10^E + g_x \times 10^{E-d} \\
&= \text{approximate } x + \text{error.}
\end{aligned}
$$

where f_x is the mantissa, d is the length of the mantissa permitted and E is the exponent. In chopping, g_x is ignored entirely and therefore,

$$\text{Error} = g_x \times 10^{E-d},\ 0 \le g_x < 1$$

The absolute error introduced depends on the following:

1. the size of the digits dropped
2. number of digits in mantissa
3. the size of the number

Since the maximum value of g_x is less than 1.0,

$$\boxed{\text{Absolute error} \le 10^{E-d}}$$

Symmetric Roundoff

In the symmetric roundoff method, the last retained significant digit is "rounded up" by 1 if the first discarded digit is larger or equal to 5; otherwise, the last retained digit is unchanged. For example, the number 42.7893 would become 42.79 and the number 76.5432 would become 76.54.

As before, the value of unrounded number can be expressed as

$$\text{True } x = f_x \times 10^E + g_x \times 10^{E-d}$$

When $g_x < 0.5$, entire g_x is truncated and therefore,

$$\text{Approximate } x = f_x \times 10^E$$

and

$$\text{Error} = g_x \times 10^{E-d}, \qquad g_x < 0.5$$

When $g_x \ge 0.5$, the last digit in the mantissa is increased by 1 and therefore

$$\text{Approximate } x = (f_x + 10^{-d}) \times 10^E = f_x \times 10^E + 10^{E-d}$$

$$\begin{aligned}\text{Error} &= [f_x \times 10^E + g_x \times 10^{E-d}] - [f_x \times 10^E + 10^{E-d}] \\ &= (g_x - 1) \times 10^{E-d} \qquad g_x \ge 0.5\end{aligned}$$

In either case, 10^{E-d} is multiplied by factor whose absolute value is no greater than 0.5. Therefore, the value of the absolute error is

$$\boxed{\text{Absolute error} \le 0.5 \times 10^{E-d}}$$

Note that the symmetric rounding error is, at worst, one-half the chopping error.

Sometimes a slightly more refined rule is used when the g_x is exactly equal to 0.5. Here f_x is unchanged if its last digit is even and is increased by 1 if its last digit is odd.

Example 4.4

Find the roundoff error in storing the number 752.6835 using a four digit mantissa.

$$\text{True } x = 0.7526 \times 10^3 + 0.835 \times 10^{-1}$$

Chopping method

Approximate $x = 0.7526 \times 10^3$

Error $= 0.0835$

Symmetric rounding

$$\text{Error} = (g_x - 1) \times 10^{-1}$$

$$= -0.165 \times 10^{-1} = -0.0165$$

Approximate $x = 0.7527 \times 10^3$

Truncation Errors

Truncation errors arise from using an approximation in place of an exact mathematical procedure. Typically, it is the error resulting from the truncation of the numerical process. We often use some finite number of terms to estimate the sum of an infinite series. For example,

$$S = \sum_{i=0}^{\infty} a_i x^i \text{ is replaced by the finite sum } \sum_{i=0}^{n} a_i x^i$$

The series has been truncated.

Another example is the use of a number of discrete steps in the solution of a differential equation. The error introduced by such discrete approximations is also called *discretisation error*. Consider the following infinite series:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

When we calculate the sine of an angle using this series, we cannot use all the terms in the series for computation. We usually terminate the process after a certain term is calculated. The terms "truncated" introduce an error which is called *truncation error*.

Many of the iterative procedures used in numerical computing are infinite and, therefore, a knowledge of this error is important. Truncation error can be reduced by using a better numerical model which usually increases the number of arithmetic operations. For example, in numerical integration, the truncation error can be reduced by increasing the number of points at which the function is integrated. But care should be exercised to see that the roundoff error which is bound to increase due to increase in arithmetic operations does not off-set the reduction in truncation error.

We often use library functions to compute logarithms, exponentials, trigonometric functions, hyperbolic functions, and so on. In all these cases, a series is used to evaluate these functions. It is important to know the truncation errors introduced by these library functions. Truncation errors are discussed in detail in many places in this book.

Example 4.5 ICE

Find the truncation error in the result of the following function for $x = 1/5$ when we use (a) first three terms, (b) first four terms, and (c) first five terms.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!}$$

(a) Truncation error when first three terms are added

$$\text{Truncation error} = + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!}$$

$$= + \frac{0.2^3}{6} + \frac{0.2^4}{24} + \frac{0.2^5}{120} + \frac{0.2^6}{720}$$

$$= 0.1402755 \times 10^{-2}$$

(b) Truncation error when first four terms are added

$$\text{Truncation error} = 0.694222 \times 10^{-4}$$

(c) Truncation error when first five terms are added

$$\text{Truncation error} = 0.275555 \times 10^{-5}$$

Example 4.6 ICE

Repeat the above example for $x = -1/5$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!}$$

$$e^{-0.2} = 1 - 0.2 + \frac{0.2^2}{2} - \frac{0.2^3}{6} + \frac{0.2^4}{24} - \frac{0.2^5}{120} + \frac{0.2^6}{720}$$

(a) Truncation error (three terms) $= -0.1279255 \times 10^{-2}$

(b) Truncation error (four terms) $= +0.6665556 \times 10^{-4}$

(c) Truncation error (five terms) $= -0.257777 \times 10^{-5}$

Note that

$$|\text{T.E.}_3| < \frac{x^3}{3!}$$

$$|\text{T.E.}_4| < \frac{x^4}{4!}$$

$$|\text{T.E.}_5| < \frac{x^5}{5!}$$

CHAPTER 6

Roots of Nonlinear Equations

ICE

6.1 INTRODUCTION

Mathematical models for a wide variety of problems in science and engineering can be formulated into equations of the form

$$f(x) = 0 \tag{6.1}$$

where x and $f(x)$ may be real, complex, or vector quantities. The solution process often involves finding the values of x that would satisfy the Eq. (6.1). These values are called the *roots* of the equation. Since the function $f(x)$ becomes zero at these values, they are also known as the zeros of the function $f(x)$.

Equation (6.1) may belong to one of the following types of equations:

1. Algebraic equations
2. Polynomial equations
3. Transcendental equations

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 to be *nonlinear*. Consider the function

$$y = f(x)$$

$f(x)$ 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* function.

On the other hand, $f(x)$ is said to be nonlinear, if the response of the dependent variable y is not in direct or exact proportion to the changes in the independent variable x. For example

$$y = x^2 + 1$$

is a nonlinear function.

There are many situations in science and engineering where the relationship between variables is *nonlinear*.

Algebraic Equations

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

$$f_n\, y_n + f_{n-1}\, y_{n-1} + \ldots + f_1\, y_1 + f_0 = 0 \tag{6.2}$$

where f_i is an ith order polynomial in x. Equation (6.2) can be thought of as having a general form

$$f(x, y) = 0 \tag{6.3}$$

This implies that Eq. (6.3) portrays a dependence between the variables x and y. Some examples are:

1. $3x + 5y - 21 = 0$ (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 of x and y which satisfy them.

Polynomial Equations

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

$$a_n\, x^n + a_{n-1}\, x^{n-1} + \ldots + a_1\, x + a_0 = 0 \tag{6.4}$$

This is called n^{th} degree polynomial and has n roots. The roots may be

1. real and different
2. real and repeated
3. complex numbers

Since complex roots appear in pairs, if n is odd, then the polynomial has at least one real root. For example, a cubic equation of the type

$$a_3x^3 + a_2x^2 + a_1x + a_0 = 0$$

will have at least one real root and the remaining two may be real or complex roots. Some specific examples of polynomial equations are:

1. $5x^5 - x^3 + 3x^2 = 0$
2. $x^3 - 4x^2 + x + 6 = 0$
3. $x^2 - 4x + 4 = 0$

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. $e^x \sin x - 1/2\, x = 0$
3. $\log x^2 - 1 = 0$
4. $x - e^{1/x} = 0$

A transcendental equation may have a finite or an infinite number of real roots or may not have real root at all.

6.2 METHODS OF SOLUTION

There are a number of ways to find the roots of nonlinear equations such as those described in Section 6.1. They include:

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

In certain cases, roots can be found by using *direct analytical methods*. For example, consider a quadratic equation such as

$$ax^2 + bx + c = 0 \tag{6.5}$$

We know that the solution of this equation is

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{6.6}$$

Equation (6.6) gives the two roots of equation (6.5). 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 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 the x-axis. These points represent approximate values of the roots of the function.

Another approach to obtain approximate solution is the trial and error technique. 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 and 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. With the advent of computers, algorithmic approaches known as *iterative methods* have become popular. An iterative technique usually begins with an approximate value of the root, known as the *initial guess*, which is then successively corrected iteration by iteration. The process of iteration stops when the desired level of accuracy is obtained. Since iterative methods involve a large number of iterations and arithmetic operations to reach a solution, the use of computers has become inevitable to make the task simple and efficient.

In this chapter, we shall discuss a few iterative methods of solution that are commonly used. These methods are designed to determine the value of a single real root using some initial guess values. Later in the chapter, we shall also discuss methods to determine all the roots of a polynomial. Finally, we shall discuss the solution of a system of non-linear equations.

6.3 ITERATIVE METHODS

IVE

There are a number of iterative methods that have been tried and used successfully in various problem situations. All these methods typically generate a sequence of estimates of the solution which is expected to converge to the true solution. As mentioned earlier, all iterative methods begin their process of solution with one or more guesses at the solution being sought. Iterative methods, based on the number of guesses they use, can be grouped into two categories:

1. Bracketing methods
2. Open end methods

Bracketing methods (also known as *interpolation* methods) start with two initial guesses that 'bracket' the root and then systematically reduce the width of the bracket until the solution is reached. Two popular methods under this category are:

1. Bisection method
2. False position method

These methods are based on the assumption that the function changes sign in the vicinity of a root.

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:

1. Newton-Raphson method
2. Secant method
3. Muller's method
4. Fixed-point method
5. Bairstow's method

It may be noted that the bracketing methods require to find sign changes in the function during every iteration. Open end methods do not require this.

6.4 STARTING AND STOPPING AN ITERATIVE PROCESS

Starting the Process

Before an iterative process is initiated, we have to determine either an approximate value of root or a "search" interval that contains a root. One simple method of guessing starting points is to plot the curve of $f(x)$ and to identify a search interval near the root of interest. Graphical representation of a function cannot only provide us rough estimates of the roots, but also help us in understanding the properties of the function, thereby identifying possible problems in numerical computing. A plot of

$$f(x) = x^3 - x - 1$$

is shown in Fig. 6.1. Although $f(x)$ is a cubic function, it intersects the x-axis at only one point. This suggests that the remaining two roots are imaginary ones.

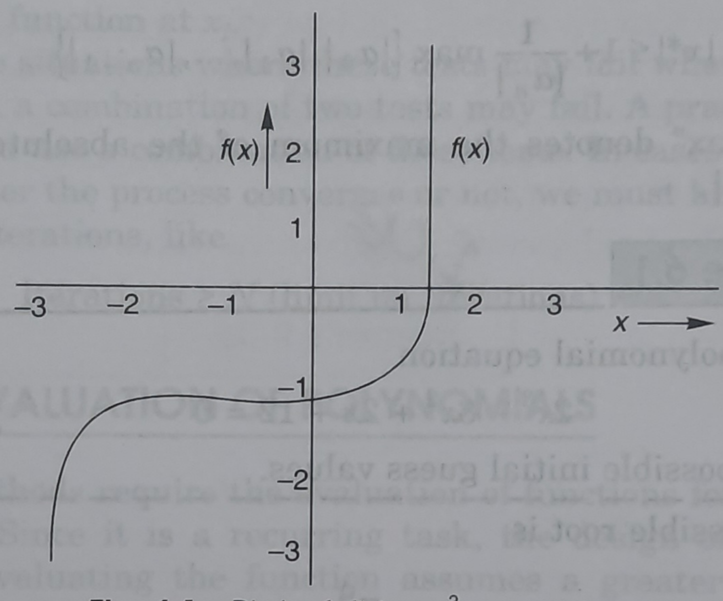


Fig. 6.1 Plot of $f(x) = x^3 - x - 1$

In the case of polynomials, many theoretical relationships between roots and coefficients are available. A few relations that might be useful for making initial guesses are described here.

Largest Possible Root For a polynomial represented by

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots a_1 x + a_0 \tag{6.7}$$

the largest possible root is given by

$$x_1^* = -\frac{a_{n-1}}{a_n} \tag{6.8}$$

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

Search Bracket Another relationship that might be useful for determining the search intervals that contain the real roots of a polynomial is

$$|x^*| \leq \sqrt{\left(\frac{a_{n-1}}{a_n}\right)^2 - 2\left(\frac{a_{n-2}}{a_n}\right)} \tag{6.9}$$

where x is the root of the polynomial. Then, the maximum absolute value of the root is

$$|x^*_{\max}| = \sqrt{\left(\frac{a_{n-1}}{a_n}\right)^2 - 2\left(\frac{a_{n-2}}{a_n}\right)} \tag{6.10}$$

This means that no root exceeds $x_{\max}$ in absolute magnitude and thus, all real roots lie within the interval $\left(-|x^*_{\max}|, |x^*_{\max}|\right)$.

There is yet another relationship that suggests an interval for roots. All real roots x satisfy the inequality

$$|x^*| \leq 1 + \frac{1}{|a_n|} \max\left\{|a_0|, |a_1|, \ldots, |a_{n-1}|\right\} \tag{6.11}$$

where the "max" denotes the maximum of the absolute values $|a_0|$, $|a_1|, \ldots, |a_{n-1}|$.

Example 6.1

ICE

Consider the polynomial equation

$$2x^3 - 8x^2 + 2x + 12 = 0$$

Estimate the possible initial guess values.

The largest possible root is

$$x_1^* = -\frac{-8}{2} = 4$$

That is, no root can be larger than the value 4.
All roots must satisfy the relation

$$|x^*| \leq \sqrt{\left(\frac{-8}{2}\right)^2 - 2\left(\frac{2}{2}\right)} = \sqrt{14}$$

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

m ## 6.6 BISECTION METHOD

ICE

The *bisection method* is one of the simplest and most reliable of iterative methods for the solution of nonlinear equations. This method, also known as *binary chopping* or *half-interval* method, relies on the fact that if $f(x)$ is real and continuous in the interval $a < x < b$, and $f(a)$ and $f(b)$ are of opposite signs, that is,

$$f(a)\, f(b) < 0$$

then there is at least one real root in the interval between a and b. (There may be more than one root in the interval).

Let $x_1 = a$ and $x_2 = b$. Let us also define another point x_0 to be the midpoint between a and b. That is,

$$x_0 = \frac{x_1 + x_2}{2} \tag{6.14}$$

Now, there exists the following three conditions:

1. if $f(x_0) = 0$, we have a root at x_0.

2. if $f(x_0)\, f(x_1) < 0$, there is a root between x_0 and x_1.
3. if $f(x_0) f(x_2) < 0$, there is a root between x_0 and x_2.

It follows that by testing the sign of the function at midpoint, we can deduce which part of the interval contains the root. This is illustrated in Fig. 6.2. It shows that, since $f(x_0)$ and $f(x_2)$ are of opposite sign, a root lies between x_0 and x_2. 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.

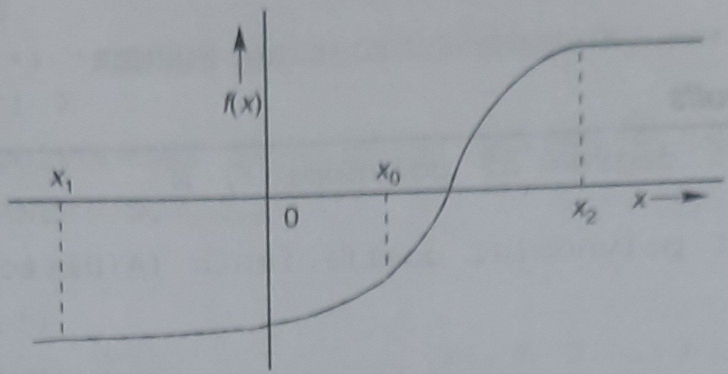


Fig. 6.2 Illustration of bisection method

Example 6.4

ICE

Find a root of the equation

$$x^2 - 4x - 10 = 0$$

using bisection method.

The first step is to guess two initial values that would bracket a root. Using Eq. (6.10), we can decide the maximum absolute of the solution. Thus

$$x_{max} = \sqrt{\left(\frac{-4}{1}\right)^2 - 2\left(\frac{-10}{1}\right)} = 6$$

Therefore, we have both the roots in the interval (–6, 6). The table below gives the values of $f(x)$ between –6 and 6 and shows that there is a root in the interval (–2, –1) and another in (5, 6).

x	–6	–5	–4	–3	–2	–1	0	1	2	3	4	5	6
$f(x)$	50	35	22	11	2	–4	–10	–13	–14	–13	–10	–5	2

Let us take $x_1 = -2$ and $x_2 = -1$.
Then

$$x_0 = \frac{-2-1}{2} = -1.5$$

$$f(-2) = 2 \text{ and } f(-1.5) = -1.75$$

Since $f(-2)\, f(-1.5) < 0$, the root must be in the interval (–2, –1.5). The next step begins.

$$x_1 = -2,\ x_2 = -1.5 \text{ and } x_0 = -1.75$$
$$f(-1.75) = 0.0625$$

Since $f(-1.75)$ and $f(-1.5)$ are of opposite sign, the root lies in the interval $(-1.75, -1.5)$. Another iteration begins.

$$x_1 = -1.75,\ x_2 = -1.5 \text{ and } x_0 = -1.625$$
$$f(-1.625) = -0.859$$

Now, the root lies in the interval $(-1.75, -1.625)$

$$x_0 = -1.6875$$
$$f(-1.6875) = -0.40$$

Next $$x_0 = -\frac{1.75 + 1.6875}{2} = -1.72$$
$$f(-1.72) = -0.1616$$

Next $$x_0 = -\frac{1.75 + 1.72}{2} = -1.735$$
$$f(-1.735) = -0.05$$

Next $$x_0 = -1.7425$$
$$f(-1.7425) = +0.0063$$

The root lies between -1.735 and -1.7425.
Approximate root is -1.7416.

An algorithm to achieve this is given in Algorithm 6.2.

Bisection Method

1. Decide initial values for x_1 and x_2 and stopping criterion, E.
2. Compute $f_1 = f(x_1)$ and $f_2 = f(x_2)$.
3. If $f_1 \times f_2 > 0$, x_1 and x_2 do not bracket any root and go to step 7; Otherwise continue.
4. Compute $x_0 = (x_1 + x_2)/2$ and compute $f_0 = f(x_0)$
5. If $f_1 \times f_0 < 0$ then
   set $x_2 = x_0$

   else

   set $x_1 = x_0$
   set $f_1 = f_0$
6. If absolute value of $(x_2 - x_1)/x_2$ is less than error E, then
   root $= (x_1 + x_2)/2$
   write the value of root
   go to step 7

   else

   go to step 4
7. Stop.

Algorithm 6.2

6.7 FALSE POSITION METHOD

In bisection method, the interval between x_1 and x_2 is 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 Fig. 6.3. Note that the root is closer to x_1. Let us join the points x_1 and x_2 by a 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* (because an approximate root is determined by linear interpolation).

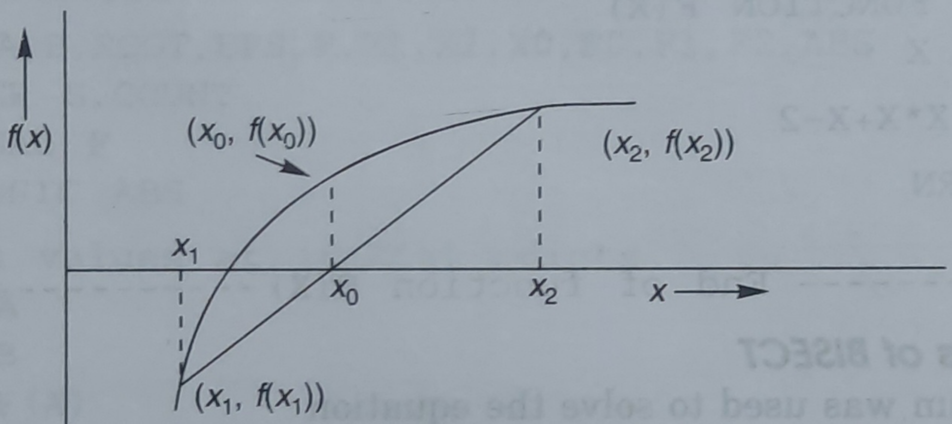


Fig. 6.3 Illustration of false position method

False Position Formula

A graphical depiction of the false position method is shown in Fig. 6.3. We know that equation of the line joining the points $(x_1, f(x_1))$ and $(x_2, f(x_2))$ is given by

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{y - f(x_1)}{x - x_1} \tag{6.16}$$

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

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

or

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

Then, we have

$$\boxed{x_0 = x_1 - \frac{f(x_1)(x_2 - x_1)}{f(x_2) - f(x_1)}} \tag{6.17}$$

This equation is known as the *false position formula*. Note that x_0 is obtained by applying a correction to x_1.

False Position Algorithm

Having calculated the first approximate to the root, the process is repeated for the new interval, as done in the bisection method, using Algorithm 6.3.

False Position Method

Let $x_0 = x_1 - f(x_1) \times \dfrac{x_2 - x_1}{f(x_2) - f(x_1)}$

If $f(x_0) \times f(x_1) < 0$

set $x_2 = x_0$

otherwise

set $x_1 = x_0$

Algorithm 6.3

A major difference between this algorithm and the bisection algorithm is the way x_0 is computed.

M **Example 6.5** ICE

Use the false position method to find a root of the function

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

in the range $1 < x < 3$

Iteration 1

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

$f(x_1) = f(1) = -2$

$f(x_2) = f(3) = 4$

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

$$= 1 + 2 \times \frac{3-1}{4+2} = 1.6667$$

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,

$$x_1 = x_0 = 1.6667$$

$$f(x_1) = f(1.6667) = -0.8889$$

$$f(x_2) = f(3) = 4$$

$$x_0 = 1.6667 + 0.8889 \times \frac{3 - 1.6667}{4 + 0.8889} = 1.909$$

Iteration 3

$$f(1.909)\, f(1.6667) = +0.2345$$

Root lies between $x_0(= 1.909)$ and $x_2(=3)$

Therefore,

$$x_1 = x_0 = 1.909$$

$$x_2 = 3$$

$$x_0 = 1.909 + 0.2647 \times \frac{3 - 1.909}{4 - 0.2647}$$

$$= 1.909 + 0.2647 \times \frac{1.091}{3.7353} = 1.986$$

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

See video and

6.8 NEWTON-RAPHSON METHOD

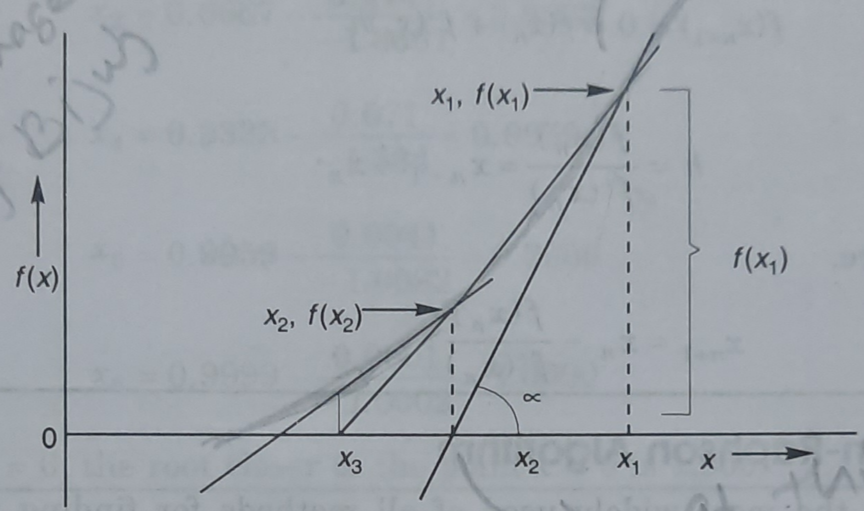
Consider a graph of $f(x)$ as shown in Fig. 6.5. Let us assume that x_1 is an approximate root of $f(x) = 0$. Draw a tangent at the curve $f(x)$ at $x = x_1$ as shown in the figure. The point of intersection of this tangent with the x-axis gives the second approximation to the root. Let the point of intersection be x_2. The slope of the tangent is given by

$$\tan \propto = \frac{f(x_1)}{x_1 - x_2} = f'(x_1) \tag{6.19}$$

where $f(x_1)$ is the slope of $f(x)$ at $x = x_1$. Solving for x_2 we obtain

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \tag{6.20}$$

This is called the *Newton-Raphson formula.*



at this point $x = x_2$, $y = 0$

Fig. 6.5 Newton-Raphson method

The next approximation would be

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

In general,

$$\boxed{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}} \tag{6.21}$$

This method of successive approximation is called the *Newton-Raphson method*. The process will be terminated when the difference between two successive values is within a prescribed limit.

The Newton-Raphson method approximates the curve of $f(x)$ by tangents. Complications will arise if the derivative $f'(x_n)$ is zero. In such cases, a new initial value for x must be chosen to continue the procedure.

eq^n of tangent at $\{x_1, f(x_1)\}$

$$y - f(x_1) = \frac{d f(x_1)}{dx}(x - x_1)$$

$\Rightarrow$ ~~y~~

$$y - f(x_1) = f'(x_1)(x - x_1)$$

at point $(x_2, 0)$ the tangent becomes

$$0 - f(x_1) = f'(x_1)(x_2 - x_1) \quad \text{as } x = x_2$$

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

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

$$y - f(x_2) = f'(x_2)(x - x_2)$$

$\Rightarrow$ at point $(x_3, 0)$ the tangent becomes

$$0 - f(x_2) = f'(x_2)(x_3 - x_2)$$

as $x = x_3$

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

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

Example 6.6

Derive the Newton-Raphson formula using the Taylor series expansion.

Assume that x_n is an estimate of a root of the function $f(x)$. Consider a small interval h such that

$$h = x_{n+1} - x_n$$

We can express $f(x_{n+1})$ using Taylor series expansion as follows:

$$f(x_{n+1}) = f(x_n) + f'(x_n)h + f''(x_n)\frac{h^2}{2!} + \ldots$$

If we neglect the terms containing the second order and higher derivatives, we get

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

If x_{n+1} is a root of $f(x)$, then

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

Then,

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

Therefore,

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

Newton-Raphson Algorithm

Perhaps the most widely used of all methods for finding roots is the Newton-Raphson method. Algorithm 6.4 describes the steps for implementing Newton-Raphson method iteratively.

Newton-Raphson Method

1. Assign an initial value to x, say x_0.
2. Evaluate $f(x_0)$ and $f(x_0)$
3. Find the improved estimate of x_0

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

4. Check for accuracy of the latest estimate.

   Compare relative error to a predefined value E. If $\left|\frac{x_1 - x_0}{x_1}\right| \le E$ stop; Otherwise continue.
5. Replace x_0 by x_1 and repeat steps 3 and 4.

Algorithm 6.4

Example 6.7

Find the root of the equation

$$f(x) = x^2 - 3x + 2$$

in the vicinity of $x = 0$ using Newton-Raphson method.

$$f'(x) = 2x - 3$$

Let $x_1 = 0$ (first approximation)

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

$$= 0 - \frac{2}{-3} = \frac{2}{3} = 0.6667$$

Similarly,

$$x_3 = 0.6667 - \frac{0.4444}{-1.6667} = 0.9333$$

$$x_4 = 0.9333 - \frac{0.071}{-1.334} = 0.9959$$

$$x_5 = 0.9959 - \frac{0.0041}{-1.0082} = 0.9999$$

$$x_6 = 0.9999 - \frac{0.0001}{-1.0002} = 1.0000$$

Since $f(1.0) = 0$, the root closer to the point $x = 0$ is 1.000.

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 $x = x_i$.

6.9 SECANT METHOD

m I CE

Secant method, like the false position and bisection methods, uses two initial estimates but does not require that they must bracket the root. For example, the secant method can use the points x_1 and x_2 in Fig. 6.6 as starting values, although they do not bracket the root. Slope of the secant line passing through x_1 and x_2 is given by

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

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

or

$$x_3\,[f(x_2) - f(x_1)] = f(x_2)\,x_1 - f(x_1)x_2$$

Then

$$\boxed{x_3 = \frac{f(x_2)x_1 - f(x_1)x_2}{f(x_2) - f(x_1)}} \qquad (6.26)$$

By adding and subtracting $f(x_2)x_2$ to the numerator and rearranging the terms we get

$$x_3 = x_2 - \frac{f(x_2)(x_2 - x_1)}{f(x_2) - f(x_1)} \tag{6.27}$$

Equation (6.27) is known as the *secant formula*. If the secant line represents the linear interpolation polynomial of the function $f(x)$ (with the interpolating points x_1 and x_2) then x_3, which intercepts the x-axis, represents the approximate root of $f(x)$.

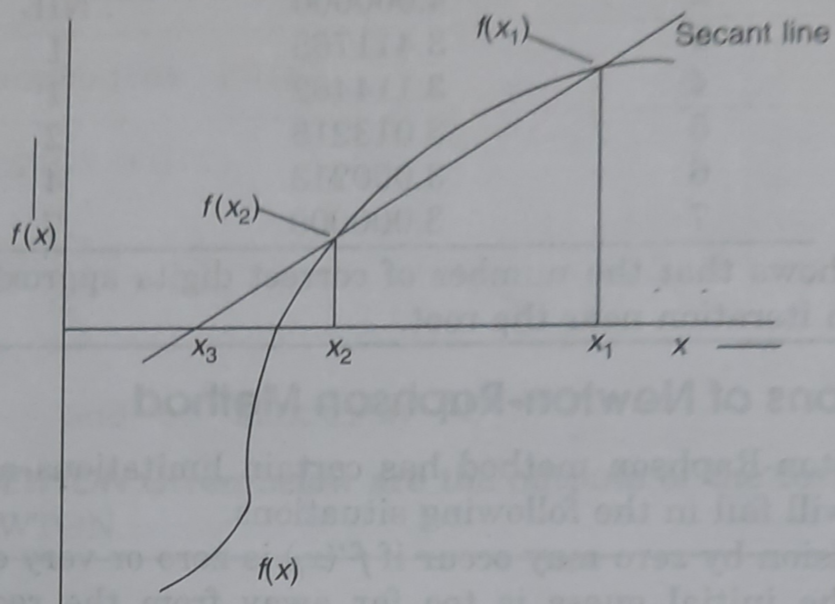


Fig. 6.6 Graphical depiction of secant method

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, in Eq. (6.27). That is, 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 in general form as follows:

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} \tag{6.28}$$

Note that Eqs (6.17) and (6.28) are similar and both of them use two initial estimates. However, there is a major difference in their algorithms of implementation. In Eq. (6.17), the latest estimate replaces one of the end points of the interval such that the new interval brackets the root. But, in Eq. (6.28) 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.

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

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

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

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

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

Secant Algorithm

ICE

Note that the value of new approximation of the root depends on the previous two approximations and corresponding functional values. Algorithm 6.5 illustrates how this procedure is implemented to estimate a root with a given level of accuracy.

Secant Method

1. Decide two initial points x_1 and x_2, accuracy level required, E.
2. Compute $f_1 = f(x_1)$ and $f_2 = f(x_2)$
3. Compute $x_3 = \dfrac{f_2 x_1 - f_1 x_2}{f_2 - f_1}$
4. Test for accuracy of x_3.

   If $\left|\dfrac{x_3 - x_2}{x_3}\right| > E$, 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

   otherwise,

   set root $= x_3$

   print results
5. Stop

Algorithm 6.5

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

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

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

Example 6.9

ICE

Use the secant method to estimate the root of the equation

m.

$$x^2 - 4x - 10 = 0$$

with the initial estimates of $x_1 = 4$ and $x_2 = 2$

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

Given $x_1 = 4$ and $x_2 = 2$

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

(Note that these points do not bracket a root)

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

$$= 2 - \frac{-14(2-4)}{-14-(-10)} = 9$$

For second iteration,

$$x = x_2 = 2$$

$$x_2 = x_3 = 9$$
$$f(x_1) = f(2) = -14$$
$$f(x_2) = f(9) = 95$$
$$x_3 = 9 - \frac{35(9-2)}{35+14} = 4$$

For third iteration,

$$x_1 = 9$$
$$x_2 = 4$$
$$f(x_1) = f(9) = 95$$
$$f(x_2) = f(4) = -10$$
$$x_3 = 4 - \frac{-10(4-9)}{-10-35} = 5.1111$$

For fourth iteration,

$$x_1 = 4$$
$$x_2 = 5.1111$$
$$f(x_1) = f(4) = -10$$
$$f(x_2) = f(5.1111) = -4.3207$$
$$x_3 = 5.1111 - \frac{-4.3207\,(5.1111-4)}{-4.3207-10} = 5.9563$$

For fifth iteration,

$$x_1 = 5.1111$$
$$x_2 = 5.9563$$
$$f(x_1) = f(5.1111) = -4.3207$$
$$f(x_2) = f(5.9563) = 5.0331$$
$$x_3 = 5.9563 - \frac{5.0331\,(5.9563-5.1111)}{5.0331+4.3207} = 5.5014$$

For sixth iteration,

$$x_1 = 5.9563$$
$$x_2 = 5.5014$$
$$f(x_1) = f(5.9539) = 5.0331$$
$$f(x_2) = f(5.5014) = -1.7392$$
$$x_3 = 5.5014 - \frac{-1.7392(5.5014-5.9563)}{-1.7392+5.0331} = 5.6182$$

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