5.4.1 Gauss Elimination Method

This is one of the most widely used methods. This method is a systematic process of eliminating unknowns from the linear equations. This method is divided into two parts:

- Triangularization
- Back substitution

The system of n equations in n unknowns is reduced to an equivalent triangular system (an equivalent system is a system having identical solution) of equation of type

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n &= b_2 \\ a_{33}x_3 + \ldots + a_{3n}x_n &= b_3 \\ &\vdots \\ a_{nn}x_n &= b_n \end{aligned} \qquad \ldots (5.5)$$

Using the back substitution procedure can easily solve this new equivalent system of equations.

$R_2 \to R_{22} - \frac{a_{21}}{a_{11}}$

The details of the Gauss elimination method are given below.

Step 1. Eliminate x_1 from 2^{nd} equation onwards. This is done as follows:

- Subtract from the second equation $\frac{a_{21}}{a_{11}}$ times the first equation. This results in

$$\left[a_{21} - \frac{a_{21}}{a_{11}}a_{11}\right]x_1 + \left[a_{22} - \frac{a_{21}}{a_{11}}a_{12}\right]x_2 + \ldots + \left[a_{2n} - \frac{a_{21}}{a_{11}}a_{1n}\right]x_n = \left[b_2 - \frac{a_{21}}{a_{11}}b_1\right]$$

  which on simplifying gives equation of type

$$a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2$$

- Similarly, subtract from the third equation $\frac{a_{31}}{a_{11}}$ times the first equation. This will give equation of type

$$a_{32}x_2 + a_{33}x_3 + \ldots + a_{3n}x_n = b_3$$

- If we repeat this process till the n^{th} equation is operated, we get the new system of equation of type, as shown overleaf.

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\
a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n &= b_2 \\
a_{32}x_2 + a_{33}x_3 + \ldots + a_{3n}x_n &= b_3 \\
&\vdots \\
a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n &= b_n
\end{aligned} \qquad \ldots (5.6)
$$

The solution of these equations is same as that of the original equations.

Step 2. Eliminate x_2 from 3^{rd} equation onwards from the system of Eqs. (5.6). This is done as follows:

- Subtract from the third equation $\frac{a_{32}}{a_{22}}$ times the second equation.
- Subtract from the fourth equation $\frac{a_{42}}{a_{22}}$ times the second equation.
- and so on.

We will thus get the new system of equations of type

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\
a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n &= b_2 \\
a_{33}x_3 + \ldots + a_{3n}x_n &= b_3 \\
&\vdots \\
a_{n3}x_3 + \ldots + a_{nn}x_n &= b_n
\end{aligned} \qquad \ldots (5.7)
$$

This process will continue till the last equation contains only one unknown, namely x_n. The final form of equation will look like

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\
a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n &= b_2 \\
a_{33}x_3 + \ldots + a_{3n}x_n &= b_3 \\
&\vdots \\
a_{nn}x_n &= b_n
\end{aligned} \qquad \ldots (5.8)
$$

This process is called *triangularization*. In order to implement this method on computer, we consider the augmented matrix of Eqs. (5.4), which after elimination takes the following form as shown overleaf:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & . & . & . & a_{1n} & a_{1(n+1)} \\ 0 & a_{22} & a_{23} & . & . & . & a_{2n} & a_{2(n+1)} \\ 0 & 0 & a_{33} & . & . & . & a_{3n} & a_{3(n+1)} \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & . & a_{nn} & a_{n(n+1)} \end{pmatrix} \quad \ldots (5.9)$$

Algorithm 5.1: Triangularizing *n* equations in *n* unknowns

The given augmented matrix *A* of order $n \times (n+1)$ is stored using a two-dimensional array named *a* of size $n \times (n+1)$.

```
Begin
  read: n
  for i = 1 to n by 1 do
      for j = 1 to (n +1) by 1 do
            read: aij
      endfor
  endfor
  for k = 1 to (n -1) by 1 do
      for i = (k+1) to n by 1 do
            set temp = aik/akk
            for j = k to (n +1) by 1 do
                  set aij = aij - temp X akj
            endfor
      endfor
  endfor
End.
```

From the triangular system of linear equations, find the values of x_n from the *n*th equation as

$$x_n = \frac{a_{n(n+1)}}{a_{nn}} \quad \ldots (5.10)$$

Then the value of x_n is substituted back in the (*n*-1)th equation to obtain the value of x_{n-1}. This back substitution is continued till we get the solution for x_1. The general formula for back substitution is

$$x_i = \frac{\left[a_{i(n+1)} - \sum_{j=i+1}^{n} a_{ij} x_j \right]}{a_{ii}} \quad \text{for } i = n-1, n-2, \ldots, 3, 2, 1 \quad \ldots (5.11)$$

This process is called *back substitution.*

Algorithm 5.2: Back Substitution

The reduced augmented matrix A of order $n\times(n+1)$ is stored using a two-dimensional array named a of size $n\times(n+1)$. The solution of the system of linear equations is stored in one-dimensional array named x of size n.

```
Begin
    set x_n = a_n(n+1) / a_nn
    for i = (n -1) to 1 by -1 do
        set sum = 0
        for j = (i+1) to n by 1 do
            set sum = sum + a_ij x_j
        endfor
        set x_i = (a_i(n+1) - sum) / a_ii
    endfor
End.
```

If we combine the steps for triangularization and back substitution, the complete algorithm for the Gauss elimination method may look like as given below:

Algorithm 5.3: Gauss elimination method

The augmented matrix A of order $n\times(n+1)$ is stored using a two-dimensional array named a of size $n\times(n+1)$. The solution of the system of linear equations is stored in one-dimensional array named x of size n.

```
Begin
    read: n
    for i = 1 to n by 1 do
        for j = 1 to (n +1) by 1 do
            read: a_ij
        endfor
    endfor
    for k = 1 to (n -1) by 1 do
        for i = (k+1) to n by 1 do
            set temp = a_ik / a_kk
            for j = k to (n +1) by 1 do
                set a_ij = a_ij - temp X a_kj
            endfor
        endfor
    endfor
    set x_n = a_n(n+1) / a_nn
    for i = (n -1) to 1 by -1 do
        set sum = 0
        for j = (i+1) to n by 1 do
            set sum = sum + a_ij x_j
        endfor
        set x_i = (a_i(n+1) - sum) / a_ii
    endfor
    write: "Solution of equations"
```

```
for i = 1 to n by 1 do
    write: xi
endfor
End.
```

Example 5.1: Solve the following system of linear equations

$R_2 \rightarrow R_2 - \frac{a_{ik}}{a_{kk}} R_1$

$$2x_1 + 8x_2 + 2x_3 = 14$$
$$x_1 + 6x_2 - x_3 = 13$$
$$2x_1 - x_2 + 2x_3 = 5$$

$a_{ij} = a_{ij} - temp \times a_{kj}$

Solution: In order to eliminate x_1 from the second and third equation, first apply transformation

$a_{32} = a_{32} - ta/a_{22}$

$\frac{a_{ik}}{a_{kk}}$

$$R_2 - \frac{a_{21}}{a_{11}} R_1 \quad \Rightarrow \quad R_2 - \frac{1}{2} R_1$$

$\frac{a_{32}}{a_{22}}$

The coefficients of the second equation are computed as

$$a_{21} = a_{21} - \frac{1}{2} a_{11} = 1 - \frac{1}{2} \times 2 = 0 \qquad a_{22} = a_{22} - \frac{1}{2} a_{12} = 6 - \frac{1}{2} \times 8 = 2$$

$$a_{23} = a_{23} - \frac{1}{2} a_{13} = -1 - \frac{1}{2} \times 2 = -2 \qquad b_2 = b_2 - \frac{1}{2} b_1 = 13 - \frac{1}{2} \times 14 = 6$$

Now apply transformation

$$R_3 - \frac{a_{31}}{a_{11}} R_1 \quad \Rightarrow \quad R_3 - \frac{2}{2} R_1 \quad \Rightarrow \quad R_3 - R_1$$

The elements of the third equation are computed as

$$a_{31} = a_{31} - a_{11} = 2 - 2 = 0 \qquad a_{32} = a_{32} - a_{12} = -1 - 8 = -9$$

$$a_{33} = a_{33} - a_{13} = 2 - 2 = 0 \qquad b_3 = b_3 - b_1 = 5 - 14 = -9$$

Thus eliminating x_1 from second and third equation, we obtain the new system of linear equations as

$$2x_1 + 8x_2 + 2x_3 = 14$$
$$2x_2 - 2x_3 = 6$$
$$-9x_2 + 0x_3 = -9$$

Finally, in order to eliminate x_2 from the third equation, apply transformation

$$R_3 - \frac{a_{32}}{a_{22}} R_2 \quad \Rightarrow \quad R_3 - \frac{-9}{2} R_2 \quad \Rightarrow \quad R_3 + \frac{9}{2} R_2$$

Scanned with CamScanner

The elements of the third equation are computed as

$$a_{32} = a_{32} + \frac{9}{2}a_{22} = -9 + \frac{9}{2}\times 2 = 0 \qquad a_{33} = a_{33} + \frac{9}{2}a_{23} = 0 + \frac{9}{2}\times(-2) = -9$$

$$b_3 = b_3 + \frac{9}{2}b_2 = -9 + \frac{9}{2}\times 6 = 18$$

The final system will look as

$$\begin{aligned} 2x_1 + 8x_2 + 2x_3 &= 14 \\ 2x_2 - 2x_3 &= 6 \\ -9x_3 &= 18 \end{aligned}$$

Through back substitution, the following solution values are obtained:

$$x_3 = \frac{18}{-9} = -2 \qquad x_2 = \frac{(6+2x_3)}{2} = 1 \qquad x_1 = \frac{(14-2x_3-8x_2)}{2} = 5$$

Incorporating pivoting, the algorithm for the implementation of the modified Gauss elimination method may look like

Algorithm 5.3: Gauss elimination method with Pivoting

The augmented matrix A of order $n\times(n+1)$ is stored using a two-dimensional array named a of size $n\times(n+1)$. The solution of the system of linear equations is stored in one-dimensional array named x of size n.

```
Begin
   read: n
   for i = 1 to n by 1 do
       for j = 1 to (n +1) by 1 do
            read: aij
       endfor
   endfor
   for k = 1 to (n -1) by 1 do
       set max = | akk |
       set p = k
       for m = (k+1) to n by 1 do
           if (| amk | > max ) then
               set max = | amk |
               set p = m
           endif
       endfor
       if ( p ≠ k ) then
           for q = k to (n+1)by 1 do
               set temp = akq
               set akq = apq
               set apq = temp
```

max = a11
p = 1
m = 2 to n
7
a2'

```
        endfor
    endif
    for i = (k+1) to n by 1 do
        set temp = a_ik/a_kk
        for j = k to (n +1) by 1 do
            set a_ij = a_ij - temp × a_kj
        endfor
    endfor
endfor
set x_n = a_n(n+1)/a_nn
for i = (n -1) to 1 by -1 do
    set sum = 0
    for j = (i+1) to n by 1 do
        set sum = sum + a_ij x_j
    endfor
    set x_i = (a_i(n+1) - sum)/a_ii
endfor
write: "Solution of equations"
for i = 1 to n by 1 do
    write: x_i
endfor
End.
```

5.5.2 Gauss-Seidel Method

In the Jacobi's method, even though the new values of unknowns are computed in each iteration, but the values of unknowns in the previous iterations are used in the subsequent iterations. That is, although new value of x_1 is computed from the first equation in a current iteration, but it is not used to compute the new values of other unknowns in the current iteration. Note that the new values of the unknowns are better than the old values, and should be used in preference to the poorer values. This was the main objective of Gauss-Seidel method.

The sequence of steps constituting the Gauss-Seidel method is as follows:

Initialisation Step:

To begin the procedure, assign an initial value to each unknown, if you can make a reasonable guess. These initial values will not effect the convergence, but will effect the number of iterations required for convergence. Usually, we assign value zero to all unknowns if no better initial estimates are at a hand.

Iteration 1

i) Find the value of x_1 from the first equation by substituting the initial values of other unknowns.

ii) Find the value of x_2 from the second equation by substituting current value of x_1 and the initial values of other unknowns.

iii) Find the value of x_3 from the third equation by substituting the current values of x_1 and x_2 and the initial values of other unknowns.

⋮

And so on, till the value of x_n is computed from the nth equation using current values of $x_1, x_2, \ldots, x_{n-1}$.

Iteration 2

i) Find the value of x_1 from the first equation by substituting the values of other unknowns obtained in the first iteration.

ii) Find the value of x_2 from the second equation by substituting current values of other unknowns.

iii) Find the value of x_3 from the third equation by substituting the current values other unknowns.

⋮

And so on till the value of x_n is computed from the nth equation using current values of $x_1, x_2, \ldots, x_{n-1}$.

Iteration 3

i) Find the value of x_1 from the first equation by substituting the values of other unknowns obtained in the second iteration.

ii) Find the value of x_2 from the second equation by substituting the current values of other unknowns.

iii) Find the value of x_3 from the third equation by substituting the current values of other unknowns.

⋮ And so on till the value of x_n is computed from the nth equation using current values of $x_1, x_2, \ldots, x_{n-1}$.

This iterative procedure is continued until the successive values of each unknown differ only within the permissible limits. In other words, the error between the successive values of each unknown is less than or equal to the prescribed tolerance.

Example 5.6: Solve the following system of equations, accurate to four significant digits.

$$\begin{aligned} 10x_1 + x_2 + 2x_3 &= 44 \\ 2x_1 + 10x_2 + x_3 &= 51 \\ x_1 + 2x_2 + 10x_3 &= 61 \end{aligned}$$

Solution: As you can see the system is diagonal system, therefore the convergence is assured. Since we want the solution correct to four significant digits, therefore the iterative process will be terminated as soon as we find that successive iteration do not produce any change at first four significant positions. We rewrite the given system of equations as

$$x_1 = \frac{1}{10}(44 - x_2 - 2x_3) \qquad x_2 = \frac{1}{10}(51 - 2x_1 - x_3) \qquad x_3 = \frac{1}{10}(61 - x_1 - 2x_2)$$

We start with initial approximation as

$$x_1 = x_2 = x_3 = 0$$

Iteration 1: Substituting $x_2 = x_3 = 0$ in the first equation, we obtain

$$x_1 = 4.4$$

Substituting $x_1 = 4.4$ and $x_3 = 0$ in the second equation, we obtain

$$x_2 = 4.22$$

Substituting $x_1 = 4.4$ and $x_2 = 4.22$ in the third equation, we obtain

$$x_3 = 4.816$$

Thus, we obtain

$$x_1 = 4.4 \qquad x_2 = 4.22 \qquad x_3 = 4.816$$

as new approximations at the end of the first iteration.

Iteration 2: Now substituting $x_2 = 4.22$ and $x_3 = 4.816$ in the first equation, we obtain

$x_1 = 4.0154$

Next substituting $x_1 = 4.0154$ and $x_3 = 4.816$ in the second equation, we obtain

$x_2 = 3.0148$

Next substituting $x_1 = 4.0154$ and $x_2 = 3.0148$ in the third equation, we obtain

$x_3 = 5.0955$

Thus, we obtain

$x_1 = 4.0154 \qquad x_2 = 3.0148 \qquad x_3 = 5.0955$

as new approximations at the end of the second iteration.

Iteration 3: Now substituting $x_2 = 3.0148$ and $x_3 = 5.0955$ in the first equation, we obtain

$x_1 = 3.0794$

Next substituting $x_1 = 3.0794$ and $x_3 = 5.0955$ in the second equation, we obtain

$x_2 = 3.9746$

Next substituting $x_1 = 3.0794$ and $x_2 = 3.9746$ in the third equation, we obtain

$x_3 = 4.9971$

Thus, we obtain

$x_1 = 3.0794 \qquad x_2 = 3.9746 \qquad x_3 = 4.9971$

as new approximations at the end of the third iteration.

Iteration 4: Now substituting $x_2 = 3.0794$ and $x_3 = 4.9971$ in the first equation, we obtain

$x_1 = 3.0031$

Next substituting $x_1 =3.0031$ and $x_3 = 4.9971$ in the second equation, we obtain

$x_2 = 3.9997$

Next substituting $x_1 = 3.0031$ and $x_2 = 3.9997$ in the third equation, we obtain

$x_3 = 4.8001$

Thus, we obtain

$x_1 = 3.0031 \qquad x_2 = 3.9997 \qquad x_3 = 4.8001$

as new approximations at the end of the fourth iteration.

Iteration 5: Now substituting $x_2 = 3.9997$ and $x_3 = 4.8001$ in the first equation, we obtain

$$x_1 = 3.0400$$

Next substituting $x_1 = 3.0400$ and $x_3 = 4.8001$ in the second equation, we obtain

$$x_2 = 4.0120$$

Next substituting $x_1 = 3.0400$ and $x_2 = 4.0120$ in the third equation, we obtain

$$x_3 = 4.8360$$

Thus, we obtain

$$x_1 = 3.0400 \qquad x_2 = 4.0120 \qquad x_3 = 4.8360$$

as new approximations at the end of the fifth iteration.

Iteration 6: Now substituting $x_2 = 4.0120$ and $x_3 = 4.8360$ in the first equation, we obtain

$$x_1 = 3.0316$$

Next substituting $x_1 = 3.0316$ and $x_3 = 4.8360$ in the second equation, we obtain

$$x_2 = 4.0101$$

Next substituting $x_1 = 3.0316$ and $x_2 = 4.0101$ in the third equation, we obtain

$$x_3 = 4.9948$$

Thus, we obtain

$$x_1 = 3.0316 \qquad x_2 = 4.0101 \qquad x_3 = 4.9948$$

as new approximations at the end of the sixth iteration.

Iteration 7: Now substituting $x_2 = 4.0101$ and $x_3 = 4.9948$ in the first equation, we obtain

$$x_1 = 3.0000$$

Next substituting $x_1 = 3.0000$ and $x_3 = 4.9948$ in the second equation, we obtain

$$x_2 = 4.0001$$

Next substituting $x_1 = 3.0000$ and $x_2 = 4.0001$ in the third equation, we obtain

$$x_3 = 5.0000$$

Thus, we obtain

$$x_1 = 3.0000 \qquad x_2 = 4.0001 \qquad x_3 = 5.0000$$

as new approximations at the end of the seventh iteration.

Iteration 8: Now substituting $x_2 = 4.0001$ and $x_3 = 5.0000$ in the first equation, we obtain

$$x_1 = 3.0000$$

Next substituting $x_1 = 3.0000$ and $x_3 = 5.0000$ in the second equation, we obtain

$$x_2 = 4.0000$$

Next substituting $x_1 = 3.0000$ and $x_2 = 4.0000$ in the third equation, we obtain

$$x_3 = 5.0000$$

Thus, we obtain

$$x_1 = 3.0000 \qquad x_2 = 4.0000 \qquad x_3 = 5.0000$$

as new approximations at the end of the eighth iteration.

By comparing the approximations of the seventh and eighth iterations, we find that there is no variation in first four significant digits; therefore we take the solution obtained at the end of the eighth iteration as the desired solution.

Hence, the solution correct to four significant digits (with truncation) is

$$x_1 = 3.000 \qquad x_2 = 4.000 \qquad x_3 = 5.000$$

Algorithm 5.10: Gauss-Seidel Method

The augmented matrix A of order $n\times(n+1)$ is stored using a two-dimensional array named a of size $n\times(n+1)$. The solution vector is stored in one-dimensional arrays named x of size n. The variable *epsilon* represents the prescribed tolerance and variable *maxiter* represents the number of permitted iterations.

```
Begin
   read: n
   for i = 1 to n by 1 do
        for j = 1 to (n+1) by 1 do
             read: a_ij
        endfor
   endfor
   read: maxiter, epsilon
   for i = 1 to n by 1 do
        set x_i = 0
   endfor
   for k = 1 to maxiter by 1 do
        set bigrelerror = 0
        for i = 1 to n by 1 do
             set sum = 0
             for j = 1 to n by 1 do
                  if ( i ≠ j ) then
                       set sum = sum + a_ij × x_j
                  endif
             endfor
```

```
            set temp = (a_i(n+1) - sum)/a_ii
            set relerror = |((temp - x_i)/temp)|
            set x_i = temp
            if ( relerror > bigrelerror ) then
               set bigrelerror = relerror
            endif
        endfor
        if ( bigrelerror ≤ epsilon) then
            write: "Solution converges in ", k, " iterations"
             for i = 1 to n by 1 do
                write: x_i
             endfor
            exit
        endif
    endfor
    write: "Solution does not converges in ", maxiter, " iterations"
End.
```

Chapter 7

Curve Fitting and Method of Least Squares

Learning Objectives

After reading this chapter, students will be able to explain the

- explain the process of curve fitting
- list various methods for curve fitting
- describe the method of least squares
- explain the procedure for curve fitting
- describe the steps for fitting various types of curves
- use these methods to fit a given curve through a set of given data
- write programs for fitting variety of curves

7.1 INTRODUCTION

Very often, in practice, a relationship is found to exist between two or more variables. It is frequently desirable to express this relationship in a mathematical form by determining an equation connecting these variables. The general problem of finding equations of approximating curves which fit given set of data is called *regression analysis* or *curve fitting*. There are various methods available for approximating curves, but the most popular and useful method is the *method of least squares*.

In general sense, the curve fitting involves the determination of the continuous function

$$y = f(x) \qquad \ldots (7.1)$$

which results in the most reasonable or best fit for the given set of values of (x_1, y_1), (x_2, y_2), (x_3, y_3), . . ., (x_n, y_n). The particular form of $f(x)$ sometimes is known in advance from the physical laws or theory associated with the x and y variables being measured. In other cases, the useful practice is to prepare a scatter diagram, i.e. plot all or some of the points if number of points is large, and the appearance of the plotted points may suggest the particular form of $f(x)$. Figure 7.1 depicts some popular curves drawn in this manner.

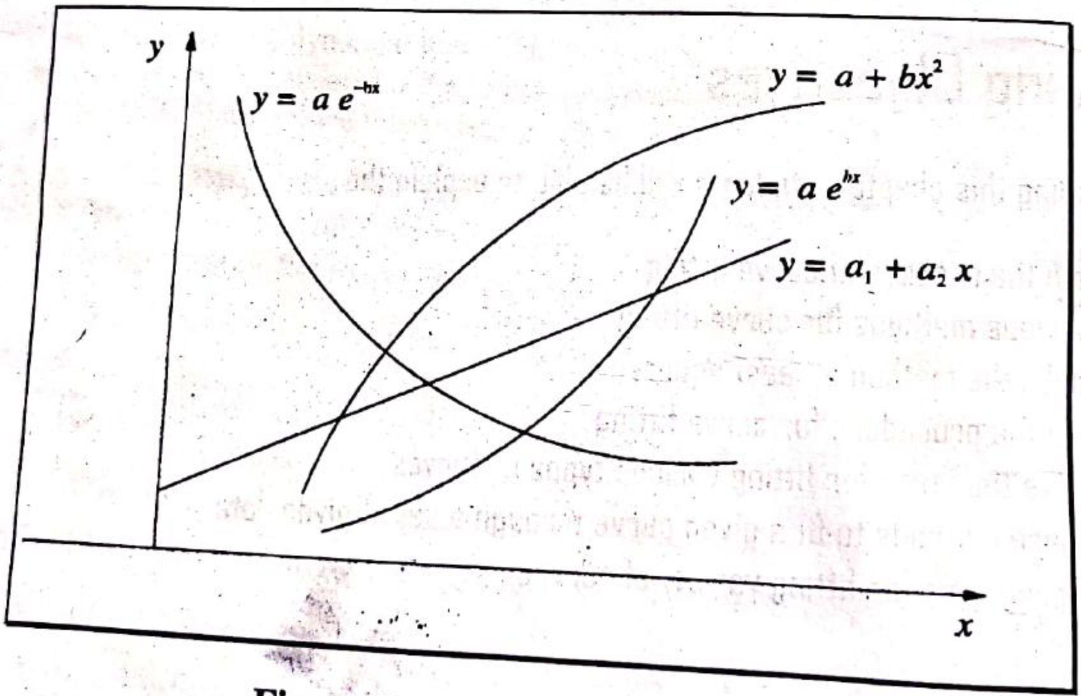

Figure 7.1: Some popular curves

Different methods available in literature to fit a curve to a given data are:

- Graphical Method
- Method of group averages
- Method of least squares
- Method of moments

The method of least squares is the best method to fit a curve to the given data.

In this chapter, we will focus our discussion on method of least squares for fitting variety of curves through a set of points given in the form of a table.

7.2 METHOD OF LEAST SQUARES

Let us suppose that $x_1, x_2, x_3, \ldots, x_n$ are the values of the independent variables x and $y_1, y_2, y_3, \ldots, y_n$ are values of the dependent variable y. The scatter diagram for this given data is:

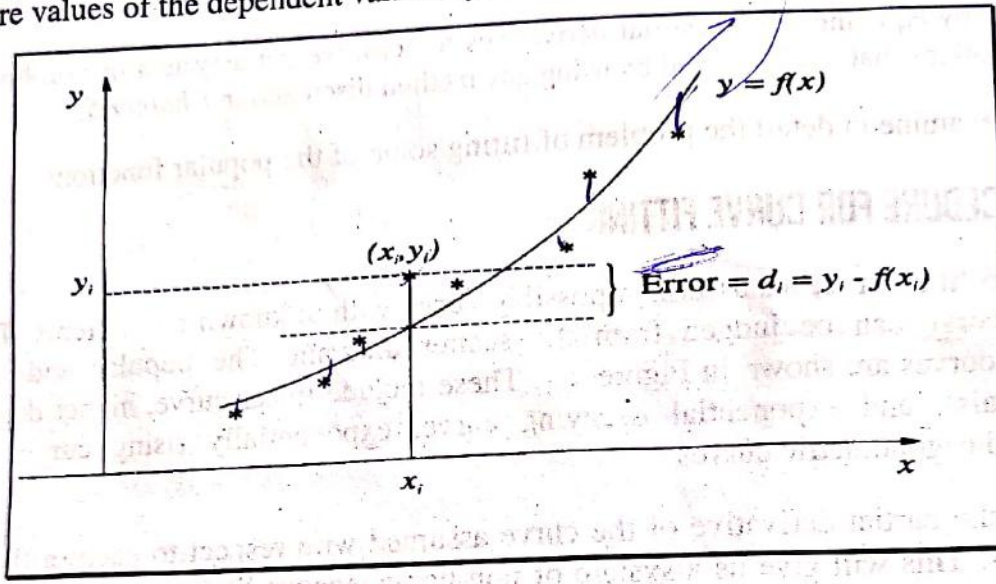

Figure 7.2: Scatter diagram

Let us suppose that

$$y = f(x)$$

be an approximation to the function.

The errors (i.e. deviations d_i's) between y's obtained from the above approximation and the actual tabulated values of y,s are

$$\begin{aligned} d_1 &= y_1 - f(x_1) \\ d_2 &= y_2 - f(x_2) \\ d_3 &= y_3 - f(x_3) \\ &\vdots \\ d_n &= y_n - f(x_n) \end{aligned}$$

The least square method states that

> *A curve is a best fit (i.e. a best curve), if the sum of the squares of deviations of the individual points from the curve is minimum.*

We form the sum, say S, of the squares of the deviations as

$$S = d_1^2 + d_2^2 + d_3^2 + ... + d_n^2 = \sum_{i=1}^{n} d_i^2 \qquad ...(7.2)$$

This sum will be either maximum or minimum if the partial derivatives of type $\partial S / \partial a = 0$, where a is one of the unknowns assumed in the approximation $y = f(x)$.

This way, by equating all the partial derivatives to zero, we get a system of non-homogeneous linear equations that can be solved by using any method discussed in *Chapter 5*.

We now examine in detail the problem of fitting some of the popular functions.

7.3 PROCEDURE FOR CURVE FITTING

In order to fit a curve, we assume a possible curve with unknown coefficients. The type of possible curve can be judged from the scatter diagram. The popular and most often occurring curves are shown in Figure 7.1. These include linear curve, higher degree curves (polynomials), and exponential decaying curve, exponentially rising curve, geometric curves, and trigonometric curves.

Then we take partial derivative of the curve assumed with respect to each of the unknown coefficients. This will give us a system of non-homogeneous linear equations with number of equations equal to number of unknowns. Then, this system of equation can be solved using any method of finding solution of system of non-homogeneous linear equations.

Once unknown coefficients are determined, we substitute their values in assumed curve, and thus the procedure for curve fitting reaches the logical end.

7.4 FITTING A STRAIGHT LINE

Let us suppose that $x_1, x_2, x_3, \ldots, x_n$ are the values of the independent variable x and $y_1, y_2, y_3, \ldots, y_n$ are values of the dependent variable y. These are plotted as shown in figure 7.3.

The simplest type of the approximating curve is a straight line, whose equation can be written as

$$y = f(x) = a_2 x + a_1 \qquad ...(7.3)$$

where a_2 and a_1 represent the slope and the intercept of the line, respectively, and are known as *regression coefficients*, and the line is known as the regression line of the y on x. Now to find the equation of the best line we have to compute these regression coefficients.

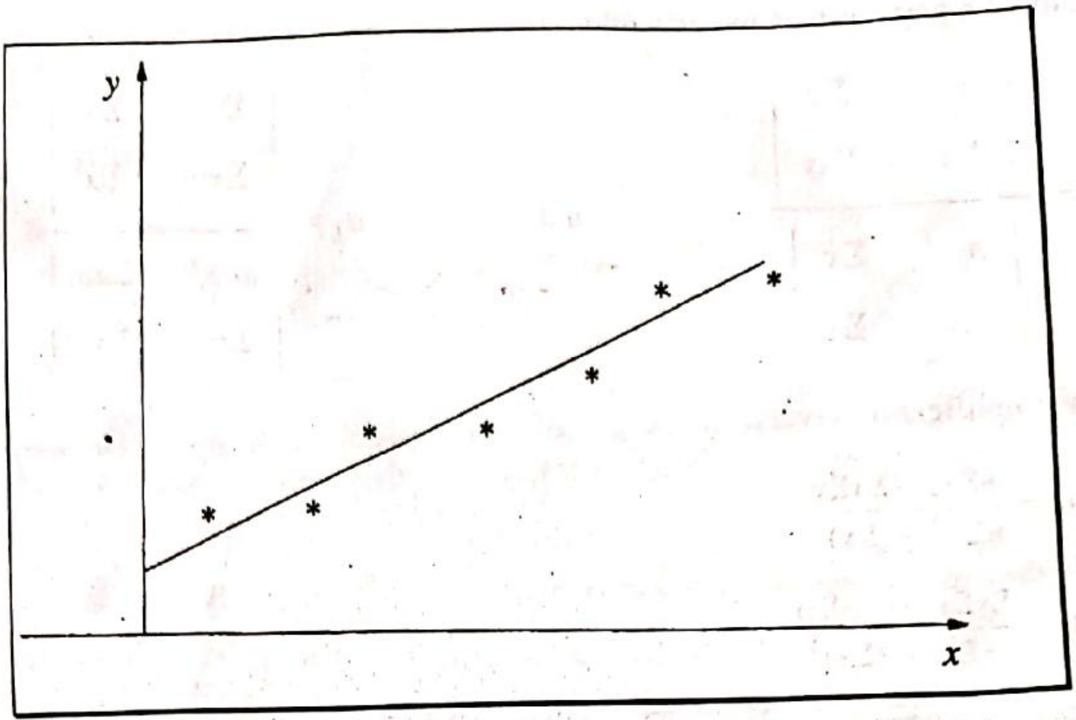

Figure 7.3: Fitting a Straight Line

The procedure used is described below:

The sum of the squares of the deviations is

$$S = \sum_{i=1}^{n} d_i^2 = \sum_{i=1}^{n} (y - f(x))^2 = \sum_{i=1}^{n} (y - a_2 x - a_1)^2$$

Therefore

$$\frac{\partial S}{\partial a_1} = 2\sum_{i=1}^{n} (y - a_2 x - a_1)(-1) \quad \ldots (7.4)$$

$$\frac{\partial S}{\partial a_2} = 2\sum_{i=1}^{n} (y - a_2 x - a_1)(-x) \quad \ldots (7.5)$$

According to the principle of least squares, these partial derivatives must be equal to zero. Hence, equating them to zero and simplifying, we get the following linear equations

$$na_1 + a_2 \Sigma x = \Sigma y \quad \ldots (7.6)$$

$$a_1 \Sigma x + a_2 \Sigma x^2 = \Sigma xy \quad \ldots (7.7)$$

These equations are called *normal equations* for the least square line. In the above normal equations we have used Σ instead of $\sum_{i=1}^{n}$ for simplicity and will continue to use this notation in rest of the chapter.

Solving these equations by Cramer's rule

$$a_2 = \frac{\begin{vmatrix} n & \Sigma y \\ \Sigma x & \Sigma xy \end{vmatrix}}{\begin{vmatrix} n & \Sigma x \\ \Sigma x & \Sigma x^2 \end{vmatrix}} \quad \text{and} \quad a_1 = \frac{\begin{vmatrix} \Sigma y & \Sigma x \\ \Sigma xy & \Sigma x^2 \end{vmatrix}}{\begin{vmatrix} n & \Sigma x \\ \Sigma x & \Sigma x^2 \end{vmatrix}}$$

which on simplification gives

$$a_2 = \frac{n\Sigma xy - \Sigma x\Sigma y}{n\Sigma x^2 - (\Sigma x)^2} \qquad \ldots (7.8)$$

$$a_1 = \frac{\Sigma y\Sigma x^2 - \Sigma x\Sigma xy}{n\Sigma x^2 - (\Sigma x)^2} \qquad \ldots (7.9)$$

Once a_2 is computed, the value of a_1 can also be computed as

$$a_1 = \frac{\Sigma y}{n} - a_2 \frac{\Sigma x}{n} \qquad \ldots (7.10)$$

Proceeding on the same lines, we can find the regression line of x on y, whose general form may be represented as

$$x = g(y) = b_2 y + b_1 \qquad \ldots (7.11)$$

whose normal equations will be

$$\begin{cases} nb_1 + b_2\Sigma y = \Sigma x \\ b_1\Sigma y + b_2\Sigma y^2 = \Sigma yx \end{cases} \qquad \ldots (7.12)$$

Solving these equations by Cramer's rule, we get ... (7.13)

$$b_2 = \frac{\begin{vmatrix} n & \Sigma x \\ \Sigma y & \Sigma yx \end{vmatrix}}{\begin{vmatrix} n & \Sigma y \\ \Sigma y & \Sigma y^2 \end{vmatrix}} \quad \text{and} \quad a_1 = \frac{\begin{vmatrix} \Sigma x & \Sigma y \\ \Sigma yx & \Sigma y^2 \end{vmatrix}}{\begin{vmatrix} n & \Sigma y \\ \Sigma y & \Sigma y^2 \end{vmatrix}}$$

which on simplification gives

$$b_2 = \frac{n\Sigma yx - \Sigma y \Sigma x}{n\Sigma y^2 - (\Sigma y)^2} \qquad \ldots (7.14)$$

$$b_1 = \frac{\Sigma x \Sigma y^2 - \Sigma y \Sigma yx}{n\Sigma y^2 - (\Sigma y)^2} \qquad \ldots (7.15)$$

Once b_2 is computed, the value of b_1 can also be computed as

$$b_1 = \frac{\Sigma x}{n} - b_2 \frac{\Sigma y}{n} \qquad \ldots (7.16)$$

An important property of these regression lines is that they intersect at a point $\left(\frac{\Sigma x}{n}, \frac{\Sigma y}{n}\right)$.

Example 7.1: Given a table of values for the function as

x:	0.1	0.2	0.3	0.4	0.5	0.6
y:	5.1	5.3	5.6	5.7	5.9	6.1

Determine both the regression lines, and also prove that they intersect at $\left(\frac{\Sigma x}{n}, \frac{\Sigma y}{n}\right)$.

Solution: The values of Σx, Σx^2, Σy, Σy^2, Σxy are computed as shown in the following table.

i	x_i	y_i	x_i^2	y_i^2	$x_i y_i$
1	0.1	5.1	0.01	26.01	0.51
2	0.2	5.3	0.04	28.09	1.06
3	0.3	5.6	0.09	31.36	1.68
4	0.4	5.7	0.16	32.49	2.28
5	0.5	5.9	0.25	34.81	2.95
6	0.6	6.1	0.36	37.21	3.66
$n = 6$	$\Sigma x = 2.1$	$\Sigma y = 33.7$	$\Sigma x^2 = 0.91$	$\Sigma y^2 = 189.97$	$\Sigma xy = 12.14$

Regression line of y on x

Let the regression line of y on x be of type

$$y = a_2 x + a_1$$

Substituting the values computed in the above table in Eqs. (7.8) and (7.9), and simplifying we get

$$a_2 = \frac{n\Sigma xy - \Sigma x \Sigma y}{n\Sigma x^2 - (\Sigma x)^2} = \frac{6 \times 12.14 - 2.1 \times 33.7}{6 \times 0.91 - (2.1)^2} = 1.97$$

$$a_1 = \frac{\Sigma y \Sigma x^2 - \Sigma x \Sigma xy}{n\Sigma x^2 - (\Sigma x)^2} = \frac{33.7 \times 0.91 - 2.1 \times 12.14}{6 \times 0.91 - (2.1)^2} = 4.93$$

The final value of a_2 and a_1 is rounded to two decimal places.

Thus the regression line of y on x is

$$y = 1.97x + 4.93$$

Regression line of x on y

Let the regression line of x on y be of type

$$x = b_2 y + b_1$$

Substituting the values computed in the above table in Eqs. (7.8) and (7.9), and simplifying we get

$$b_2 = \frac{n\Sigma xy - \Sigma y \Sigma x}{n\Sigma y^2 - (\Sigma y)^2} = \frac{6 \times 12.14 - 33.7 \times 2.1}{6 \times 189.97 - (33.7)^2} = 0.50$$

$$b_1 = \frac{\Sigma x \Sigma y^2 - \Sigma y \Sigma xy}{n\Sigma y^2 - (\Sigma y)^2} = \frac{2.1 \times 189.97 - 33.7 \times 12.14}{6 \times 189.97 - (33.7)^2} = -2.46$$

The final value of b_2 and b_1 is rounded to two decimal places.

Thus the regression line of x on y is

$$x = 0.50y - 2.46$$

Now we would see that they intersect at $\left(\frac{\Sigma x}{n}, \frac{\Sigma y}{n}\right)$, i.e., (0.35, 5.62). This we can be done if we can prove that both of these regression lines pass through it. Let us first check for the regression line of y on x, *i.e.*,

$$y = 1.97x + 4.93$$

Substituting $x = 0.35$ on the right hand side, we get

$$y = 1.97 * 0.35 + 4.93 = 0.69 + 4.93 = 5.62$$

which shows that the regression line of y on x passes through (0.35, 5.62).

Now check for the regression line of x on y i.e.

$$x = 0.50y - 2.46$$

Substituting $y = 5.62$ on the right hand side, we get

$$x = 0.50 * 5.62 - 2.46 = 2.81 - 2.46 = 0.35$$

which shows that the regression line of x on y also passes through (0.35, 5.62).

Since both the regression lines passes through the point (0.35, 5.62), thus we conclude that they intersect at the point $\left(\frac{\Sigma x}{n}, \frac{\Sigma y}{n}\right)$.

Algorithm 7.1: Fitting a regression line of Y on X

Two one-dimensional arrays named x and y, each of size n, are used to store the data points at which the function is tabulated.

```
Begin
    read: n
    for i = 1 to n by 1 do
        read: x_i, y_i
    endfor
    set sx  = 0
    set sx2 = 0
    set sy  = 0
    set sxy = 0
    for i = 1 to n by 1 do
        set sx  = sx + x_i
        set sx2 = sx2 + x_i × x_i
        set sy  = sy + y_i
        set sxy = sxy + x_i × y_i
    endfor
    set denom = n×sx2 - sx×sx
    set a2 = (n×sxy - sx×sy)/ denom
    set a1 = (sy×sx2 - sx×sxy)/ denom
    write: "Regression coefficients are"
    write: "Slope of the regression line = ", a2
    write: "Y-intercept for the line = ", a1
End.
```

7.5 FITTING A POLYNOMIAL

In general it may be necessary to fit a higher degree polynomial rather a straight line. In this section, we will see how to fit a polynomial of a specific form.

The actual degree of the polynomial will be described by the physical consideration. However, we can utilize the property of difference tables. Recall that we have proved in

Chapter 6 that if a polynomial happens to be of m^{th} degree, then the m^{th} order differences will be constant and the higher order differences vanishes. Therefore by generating a difference table, we can find the degree of the polynomial.

Let us suppose that $x_1, x_2, x_3, \ldots, x_n$ are the values of the independent variable x and $y_1, y_2, y_3, \ldots, y_n$ are the values of the dependent variable y. And further suppose that we want to fit a polynomial of degree m of the form

$$y = f(x) = a_{m+1}x^m + a_m x^{m-1} + a_{m-1}x^{m-2} + \ldots + a_2 x + a_1 \quad \ldots (7.17)$$

where $a_{m+1}, a_m, a_{m-1}, \ldots, a_2, a_1$ are to be computed to give the best polynomial fit.

The sum of squares of the deviations is given by

$$S = \sum_{i=1}^{n}(y - f(x))^2 = \sum_{i=1}^{n}\left[y - (a_{m+1}x^m + a_m x^{m-1} + a_{m-1}x^{m-2} + \ldots + a_2 x + a_1)\right]^2$$

Differentiating S with respect to $a_1, a_2, a_3, \ldots, a_{m+1}$, and equating each one to zero we obtain the following normal equations

$$\left.\begin{aligned}
a_1 n + a_2\Sigma x + a_3\Sigma x^2 + \ldots + a_{m+1}\Sigma x^m &= \Sigma y \\
a_1\Sigma x + a_2\Sigma x^2 + a_3\Sigma x^3 + \ldots + a_{m+1}\Sigma x^{m+1} &= \Sigma xy \\
a_1\Sigma x^2 + a_2\Sigma x^3 + a_3\Sigma x^4 + \ldots + a_{m+1}\Sigma x^{m+2} &= \Sigma x^2 y \\
\vdots \qquad\qquad &\qquad \vdots \\
a_1\Sigma x^m + a_2\Sigma x^{m+1} + a_3\Sigma x^{m+2} + \ldots + a_{m+1}\Sigma x^{m+m} &= \Sigma x^m y
\end{aligned}\right\} \quad \ldots (7.18)$$

These are $(m+1)$ non-homogeneous equations in $(m+1)$ unknowns, and are called *normal equations* for a polynomial fit.

$$\begin{pmatrix}
n & \Sigma x & \Sigma x^2 & \cdots & \Sigma x^m \\
\Sigma x & \Sigma x^2 & \Sigma x^3 & \cdots & \Sigma x^{m+1} \\
\Sigma x^2 & \Sigma x^3 & \Sigma x^4 & \cdots & \Sigma x^{m+2} \\
\vdots & \vdots & \vdots & & \vdots \\
\Sigma x^m & \Sigma x^{m+1} & \Sigma x^{m+2} & \cdots & \Sigma x^{m+m}
\end{pmatrix}
\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{m+1} \end{pmatrix}
=
\begin{pmatrix} \Sigma y \\ \Sigma xy \\ \Sigma x^2 y \\ \vdots \\ \Sigma x^m y \end{pmatrix}$$

The above system can be written as $CA = B$, where

$$C = \begin{pmatrix} n & \Sigma x & \Sigma x^2 & \cdots & \Sigma x^m \\ \Sigma x & \Sigma x^2 & \Sigma x^3 & \cdots & \Sigma x^{m+1} \\ \Sigma x^2 & \Sigma x^3 & \Sigma x^4 & \cdots & \Sigma x^{m+2} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \Sigma x^m & \Sigma x^{m+1} & \Sigma x^{m+2} & \cdots & \Sigma x^{m+m} \end{pmatrix} \quad A = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \cdot \\ \cdot \\ \cdot \\ a_{m+1} \end{pmatrix} \quad B = \begin{pmatrix} \Sigma y \\ \Sigma xy \\ \Sigma x^2 y \\ \cdot \\ \cdot \\ \cdot \\ \Sigma x^m y \end{pmatrix}$$

The elements of the matrix C are computed as

$$c_{ij} = \sum_{k=1}^{n} x_k^{i+j-2} \qquad \text{for } i = 1, 2, 3, \ldots, m+1$$
$$\text{for } j = 1, 2, 3, \ldots, m+1 \quad \text{(for each value of } i) \qquad \ldots (7.19)$$

and the elements of the vector B are computed as

$$b_i = \sum_{k=1}^{n} x_k^{i-1} y_k \qquad \text{for } i = 1, 2, 3, \ldots, m+1 \qquad \ldots (7.20)$$

Before we can solve these equations, we have to compute the above summations, and then using any of the methods to find the solution of a system of non-homogeneous equations discussed in *Chapter 5*, we can find the unknown coefficients and this is how the polynomial relationship is established.

In the following algorithm, we will be using the Gauss elimination method with pivoting. For computer implementation, we will be storing the constants of the normal equations as the last column of the augmented matrix (i.e. as $(m+2)^{\text{th}}$ column). The step required to compute these coefficients are summarised in the following algorithm.

Example 7.2: Given a table of values for the function as

x:	1.0	1.5	2.0	2.5	3.1	4.0
y:	1.1	1.3	1.6	2.0	3.4	4.2

Fit a second degree polynomial.

Solution: Let $y = a_3x^2 + a_2x + a_1$ be a second degree polynomial.

The normal equations for this second-degree polynomial are

$$na_1 + a_2\Sigma x + a_3\Sigma x^2 = \Sigma y$$
$$a_1\Sigma x + a_2\Sigma x^2 + a_3\Sigma x^3 = \Sigma xy$$
$$a_1\Sigma x^2 + a_2\Sigma x^3 + a_3\Sigma x^4 = \Sigma x^2 y$$

Computing the values of Σx, Σx^2, Σx^3, Σx^4, Σy, Σxy, and $\Sigma x^2 y$ and substituting in the normal equations, we get

$$6a_1 + 14.1a_2 + 39.11a_3 = 13.6$$
$$14.1a_1 + 39.11a_2 + 121.791a_3 = 38.59$$
$$39.11a_1 + 121a_2 + 409.477a_3 = 122.799$$

Solving this system of non-homogeneous equations in three unknowns by Gauss elimination method using pivoting, we get

$a_1 = 0.678 \quad a_2 = 0.137 \quad a_3 = 0.195$ (rounded off to three decimal places)

Thus the equation of the second-degree polynomial is

$$y = 0.195x^2 + 0.137x + 0.678$$

Algorithm 7.2: To Fit a polynomial of degree *m*

Two one-dimensional arrays named x and y, each of size n, are used to store the data points at which the function is tabulated. Third one-dimensional array named a of size $(m+1)$ is used to store the computed values of regression coefficients.

```
Begin
    read: m, n
    for i = 1 to n by 1 do
        read: x_i, y_i
    endfor
    for i = 1 to (m+1) by 1 do
        for j = 1 to (m+1) by 1 do
            set c_ij = 0
              for k = 1 to n by 1 do
                  set c_ij = c_ij + (x_k)^(i+j-2)
            endfor
        endfor
        set c_i(m+2) = 0
        for k = 1 to n by 1 do
              set c_i(m+2) = c_i(m+2) + (x_k)^(i-1) x y_k
        endfor
endfor
```

```
for k = 1 to m by 1 do
    set max = | c_ii |
    set p = k
    for q = (k+1) to (m+1) by 1 do
        if (| c_qk | > max ) then
            set max = | c_qk |
            set p = q
        endif
    endfor
    if ( p ≠ k ) then
        for q = k to (m+1) by 1 do
            set temp = c_kq
            set c_kq = c_pq
            set c_pq = temp
        endfor
    endif
    for i = (k+1) to (m+1) by 1 do
        set temp = c_ik/c_kk
        for j = k to (m+2) by 1 do
            set c_ij = c_ij - temp × c_kj
        endfor
    endfor
endfor
set a_m+1 = c_(m+1)(m+2)/c_(m-1)(m+1)
for i = m to 1 by -1 do
    set sum = 0
    for j = (i+1) to (m+1) by 1 do
        set sum = sum + c_ij × a_j
    endfor
    set a_i = (c_i(m+2) - sum)/c_ii
endfor
write: "Regression coefficients are"
for i = 1 to (m+1) by 1 do
    write: a_i
endfor
End.
```

7.6 FITTING A NON-LINEAR FUNCTION

In many real life situations, the experimental data follow some other geometric forms which can be deduced from the inspection of the scatter diagram or other physical considerations, such as exponential curves, trigonometric curves, parabola, ellipse, etc.

The normal practice of fitting these functions is to transform them to linear form, which may not always be possible. In this section, we will consider some familiar functions that can be transformed to linear form.

7.6.1 Fitting a Geometric Curve

The geometric curve is described by the equation

$$y = ax^b \quad \ldots (7.21)$$

where b is a real number.

Taking logarithm on both sides, we get

$$\log y = \log a + b \log x$$

Substituting

$$z = \log y,\ c = \log a, \text{ and } t = \log x$$

in the above equation, we get

$$z = bt + c$$

which is in linear form. The same steps as discussed in the case of straight line fitting are applied here, and we get the following normal equations for the above geometric curve

$$nc + b\Sigma t = \Sigma z$$

$$c\Sigma t + b\Sigma t^2 = \Sigma tz$$

$$\Rightarrow \quad nc + b\Sigma \log x = \Sigma \log y$$

$$c\Sigma \log x + b\Sigma (\log x)^2 = \Sigma \log x \log y$$

Solving these equations, we get

$$\left.\begin{aligned} b &= \frac{n\Sigma \log x \log y - \Sigma \log x \Sigma \log y}{n\Sigma(\log x)^2 - (\Sigma \log x)^2} \\ c &= \frac{\Sigma \log y \Sigma(\log x)^2 - \Sigma \log x \Sigma \log x \log y}{n\Sigma(\log x)^2 - (\Sigma \log x)^2} \end{aligned}\right\} \quad \ldots (7.22)$$

Once the values of b and c are computed, the value of a is computed from the equation

$$a = e^c \quad \ldots (7.23)$$

7.6.2 Fitting an Exponential Curve

The exponential curves are described by the equations of type

$$y = ae^{bx} \quad \text{and} \quad y = ae^{-bx} \quad \ldots (7.24)$$

We take the case of the exponential curve of form

$$y = ae^{bx}$$

Taking logarithm on both sides, we get

$$\log y = \log a + bx$$

$c = \log a$

$e^c = a$

Substituting

$$z = \log y, c = \log a$$

in the above equation, we get

$$z = bx + c$$

which is in linear form. The same steps as discussed in the case of straight line fitting are applied here, and we get the following normal equations for the above exponential curve

$$nc + b\Sigma x = \Sigma z$$
$$c\Sigma x + b\Sigma x^2 = \Sigma xz$$

$$\Rightarrow \quad nc + b\Sigma x = \Sigma \log y$$
$$c\Sigma x + b\Sigma x^2 = \Sigma x \log y$$

Solving these equations, we get

$$\left.\begin{aligned} b &= \frac{n\Sigma x \log y - \Sigma x \Sigma \log y}{n\Sigma x^2 - (\Sigma x)^2} \\ c &= \frac{\Sigma \log y \Sigma x^2 - \Sigma x \Sigma x \log y}{n\Sigma x^2 - (\Sigma x)^2} \end{aligned}\right\} \quad \ldots (7.25)$$

Once the values of b and c are computed, the value of a is computed from the equation

$$a = e^c$$

Example 7.3: Given a table of values for the function as

x :	600	500	400	350
y :	2	10	26	61

It is known that a relation of type

$$y = ae^{bx}$$

exist. Find the best possible values for a and b.

Solution: The normal equations for curve of type $y = ae^{bx}$ are

$$nc + b\Sigma x = \Sigma \log y$$

$$c\Sigma x + b\Sigma x^2 = \Sigma x \log y$$

where $c = \log a$.

Compute the values of Σx, Σx^2, $\Sigma \log y$, and $\Sigma x \log y$ as shown in the table on next page.

i	x_i	y_i	$\log y_i$	x_i^2	$x_i \log y_i$
1	600	2	0.30103	360000	180.618
2	500	10	1.0	250000	500.0
3	400	26	1.414973	160000	565.9892
4	350	61	1.78533	122500	624.8655
$n = 4$	$\Sigma x = 1850$	$\Sigma y = 99$	$\Sigma \log y = 4.501333$	$\Sigma x^2 = 892500$	$\Sigma x \log y = 1871.473$

Substituting these values in the above normal equations, we get

$$4c + 1850b = 4.501333$$

$$1850c + 892500b = 1871.473$$

Solving these normal equations, we get

$$c = 3.764167$$

$$b = -0.0057056$$

Since

$$c = \log a$$

therefore

$$a = \text{antilog } c = \text{antilog } (3.764167) = 43.12777$$

Hence the relation between x and y is

$$y = 43.12777e^{-0.0057056}$$