Islamic University of Technology **(IUT)**
Organization of Islamic Cooperation **(OIC)**
Department of Electrical and Electronic Engineering **(EEE)**

Course No : Math 4522
Course Name : Numerical Methods Lab
Experiment No : 07
Experiment Name : Linear and Non- Linear Regression

Introduction:

Data is often given for discrete values along a continuum. However we may require estimates at points between the discrete values. Then we have to fit curves to such data to obtain intermediate estimates. In addition, we may require a simplified version of a complicated function. One way to do this is to compute values of the function at a number of discrete values along the range of interest. Then a simpler function may be derived to fit these values. Both of these applications are known as **curve fitting**.

There are two general approaches of curve fitting that are distinguished from each other on the basis of the amount of error associated with the data. First, where the data exhibits a significant degree of error, the strategy is to derive a single curve that represents the general trend of the data. Because any individual data may be incorrect, we make no effort to intersect every point. Rather, the curve is designed to follow the pattern of the points taken as a group. One approach of this nature is called ***least squares regression***.

Second, where the data is known to be very precise, the basic approach is to fit a curve that passes directly through each of the points. The estimation of values between well known discrete points from the fitted exact curve is called ***interpolation***.

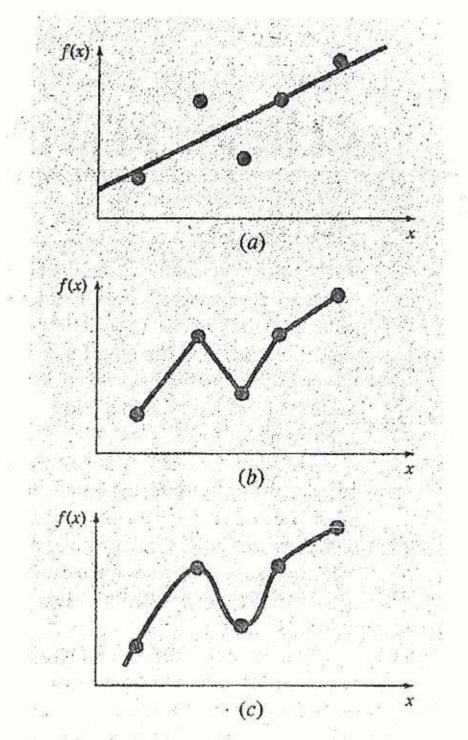


Figure 1: (a) Least squares linear regression (b) linear interpolation (c) curvilinear interpolation

Least squares Regression:

Where substantial error is associated with data, polynomial interpolation is inappropriate and may yield unsatisfactory results when used to predict intermediate values. A more appropriate strategy for such cases is to derive an approximating function that fits the shape or general trend of the data without necessarily matching the individual points. Now some criterion must be devised to establish a basis for the fit. One way to do this is to derive a curve that minimizes the discrepancy between the data points and the curve. A technique for accomplishing this objective is called least squares regression, where the goal is to minimize the sum of the square errors between the data points and the curve. Now depending on whether we want to fit a straight line or other higher order polynomial, regression may be linear or polynomial. They are described below.

Linear regression:

The simplest example of least squares regression is fitting a straight line to a set of paired observations: (x1, y1), (x2, y2), , , ,(xn, yn). The mathematical expression for straight line is

$y_m=a_0+a_1x$

Where a_0 and a_1 are coefficients representing the intercept and slope and y_m is the model value. If y_0 is the observed value and e is error or residual between the model and observation then

$e=y_0-y_m=y_0 - a_0 - a_1x$

Now we need some criteria such that the error e is minimum and also we can arrive at a unique solution (for this case a unique straight line). One such strategy is to minimize the sum of the square errors. So sum of square errors

$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_{i,observed} - y_{i,model})^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2 \quad \ldots\ldots\ldots\ldots 1$$

To determine the values of a_0 and a_1, equation (1) is differentiated with respect to each coefficient.

$$\frac{\partial S_r}{\partial a_0} = -2\sum (y_i - a_0 - a_1 x_i)$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum (y_i - a_0 - a_1 x_i)x_i$$

Setting these derivatives equal to zero will result in a minimum Sr. If this is done, the equations can be expressed as

$$0 = \sum y_i - \sum a_0 - \sum a_1 x_i$$

$$0 = \sum y_i x_i - \sum a_0 x_i - \sum a_1 x_i^2$$

Now realizing that $\sum a_0 = na_0$ **,** we can express the above equations as a set of two simultaneous linear equations with two unknowns a_0 and a_1.

$$na_0 + (\sum x_i)a_1 = \sum y_i$$

$$(\sum x_i)a_0 + (\sum x_i^2)a_1^1 = \sum x_i y_i$$

from where

$$a_1 = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}$$

$$a_0 = \bar{y} - a_1 \bar{x}$$

Where $\bar{y}$ and $\bar{x}$ are the means of y and x respectively

Demonstration :

Fit a straight line to the x and y values of table 1

Table 1:

x	Y
1	0.5
2	2.5
3	2.0
4	4.0
5	3.5
6	6.0
7	5.5

Code :

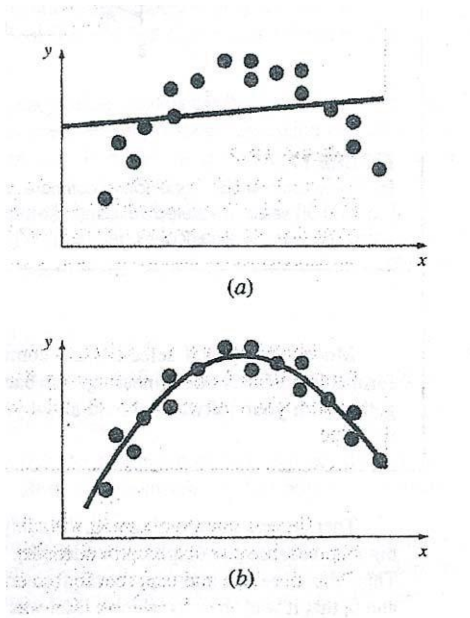
```
n=input('Number of sets of points : ');

%taking input
for i=1:n
x(i)=input('X(i):');
y(i)=input('Y(i):');
end

%calculating coefficients
sumx=0;
sumy=0;
sumxy=0;
sumxsq=0;
for i=1:n
sumx=sumx+x(i);
sumy=sumy+y(i);
sumxy=sumxy+x(i)*y(i);
sumxsq=sumxsq+x(i)^2;
end

format long ;

%calculating a1 and a0
a1=(n*sumxy-sumx*sumy)/(n*sumxsq-sumx^2)
a0=sumy/n-a1*sumx/n

%plotting observed data
plot(x,y,'o')
hold on;

%plotting fitted data
ym=a0+a1.*x;
plot(x,ym);
```

Verification :

Ans: a_0=0.0714, a_1=0.83928

Polynomial or Non –Linear Regression:

In some cases, we have some engineering data that cannot be properly represented by a straight line. We can fit a polynomial to these data using polynomial regression.



The least squares procedure can be readily extended to fit the data to a higher order polynomial. For example, we want to fit a second order polynomial

$y_m = a_0 + a_1 x + a_2 x^2$

For this case the sum of the squares of residuals is

$$S_r = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 2$$

Taking derivative of equation (2) with respect to unknown coefficients a_0, a_1 and a_2

$$\frac{\partial S_r}{\partial a_0} = -2\sum(y_i - a_0 - a_1 x_i - a_2 x_i^2)$$
$$\frac{\partial S_r}{\partial a_1} = -2\sum x_i(y_i - a_0 - a_1 x_i - a_2 x_i^2)$$
$$\frac{\partial S_r}{\partial a_2} = -2\sum x_i^2(y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

These equations can be set equal to zero and rearranged to develop the following set of normal equations:

$$na_0 + (\sum x_i)a_1 + (\sum x_i^2)a_2 = \sum y_i$$
$$(\sum x_i)a_0 + (\sum x_i^2)a_1 + (\sum x_i^3)a_2 = \sum x_i y_i$$
$$(\sum x_i^2)a_0 + (\sum x_i^3)a_1 + (\sum x_i^4)a_2 = \sum x_i^2 y_i$$

Now a_0, a_1 and a_2 can be calculated using matrix inversion.

Task:

Write a code to fit a second order polynomial to the data given in table 2

Table 2

x	y
0	2.1
1	7.7
2	13.6
3	27.2
4	40.9
5	61.1