BRAC UNIVERSITY
CSE 330: Numerical Methods (LAB)

Lab 6: Curve Fitting, Regression & Solving Linear Algebraic Equations.

Curve Fitting

Data is often given for discrete values along a continuum. However we may require estimates at points between the discrete values. For this 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. This is 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 the pattern of the points taken as a group. One approach of this process is known as ***least squares regression.***

Second, when 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***. Interpolation has been discussed in Lab 5.

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 criteria 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.

Linear Regression:

The simplest example of least squares regression is fitting a straight line to a set of paired observations: (x_1, y_1), (x_2, y_2),……(x_n, y_n). The mathematical expression for straight line is

$$y_m=a_0 + a_1x$$

Where a_0 and a_1 are co-efficients representing the intercept and slope respectively. And y_m is the model value [value according to our 'fitted' line]. If y_0 is the actual observed value and e is error or residual between the model and observation then

$$e = y_0 - y_m = y_0 - a_0 - a_1 x$$

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

$$Sr = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(y_{i.observed} - y_{i.\text{mod}el}\right)^2 = \sum_{i=1}^{n} \left(y_i - a_0 - a_1 x_i\right)^2 \quad \ldots\ldots\ldots\ldots..(1)$$

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

$$\frac{\partial \mathrm{Sr}}{\partial \mathrm{a}_0} = -2\sum \left(\mathrm{y}_i - \mathrm{a}_0 - \mathrm{a}_1 \mathrm{x}_i\right)$$

$$\frac{\partial \mathrm{Sr}}{\partial \mathrm{a}_1} = -2\sum \left(\mathrm{y}_i - \mathrm{a}_0 - \mathrm{a}_1 \mathrm{x}_i\right)\mathrm{x}_i$$

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

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

$$0 = \sum \mathrm{y}_i \mathrm{x}_i - \sum a_0 x_i - \sum a_1 x_i^2$$

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

$$n a_0 + \left(\sum x_i\right) a_1 = \sum y_i$$

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

from where,

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

$$a_0 = \bar{y} - a_1 \bar{x} \qquad \ldots\ldots\ldots\ \ldots\ldots\ldots\ldots.\ (2)$$

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

Example 1: Fit a straight line to the x and y values for the following table

X	1	2	3	4	5	6	7
Y	0.5	2.5	2	4	3.5	6	5.5

Ans: a_0 =0.07143 and a_1=0.83929

Solution:

```
clc;
clear all;
close all;
n=input('Enter Number of data points : ')
for i=1:n                              % taking the data inputs
    fprintf('Enter Data set =%d\n',i)
    x(i)=input(' x value:  ') ;
    y(i)=input(' y value:  ');
end

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

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

plot(x,y,'o');
hold on;
ym=a0+a1.*x;
plot(x,ym)
legend('Actual data points', 'Fitted straight line')
```

Figure 1: Code for linear Regression

Direct Use of MATLAB`s Curve Fitting features

It is possible to fit a straight line or any other polynomial to a set of data points and obtain the corresponding co-efficients automatically using MATLABS "*Basic Fitting*" features. To do this first we need to plot the data points.

To plot the given (x, y) data sets write in M file or Command Window, **plot (x, y,'o').** This will pop up the figure window

Then from the figure window go to **tools→ basic fitting**

As shown in figure 2, select the desired fitting type also select the show equation box.

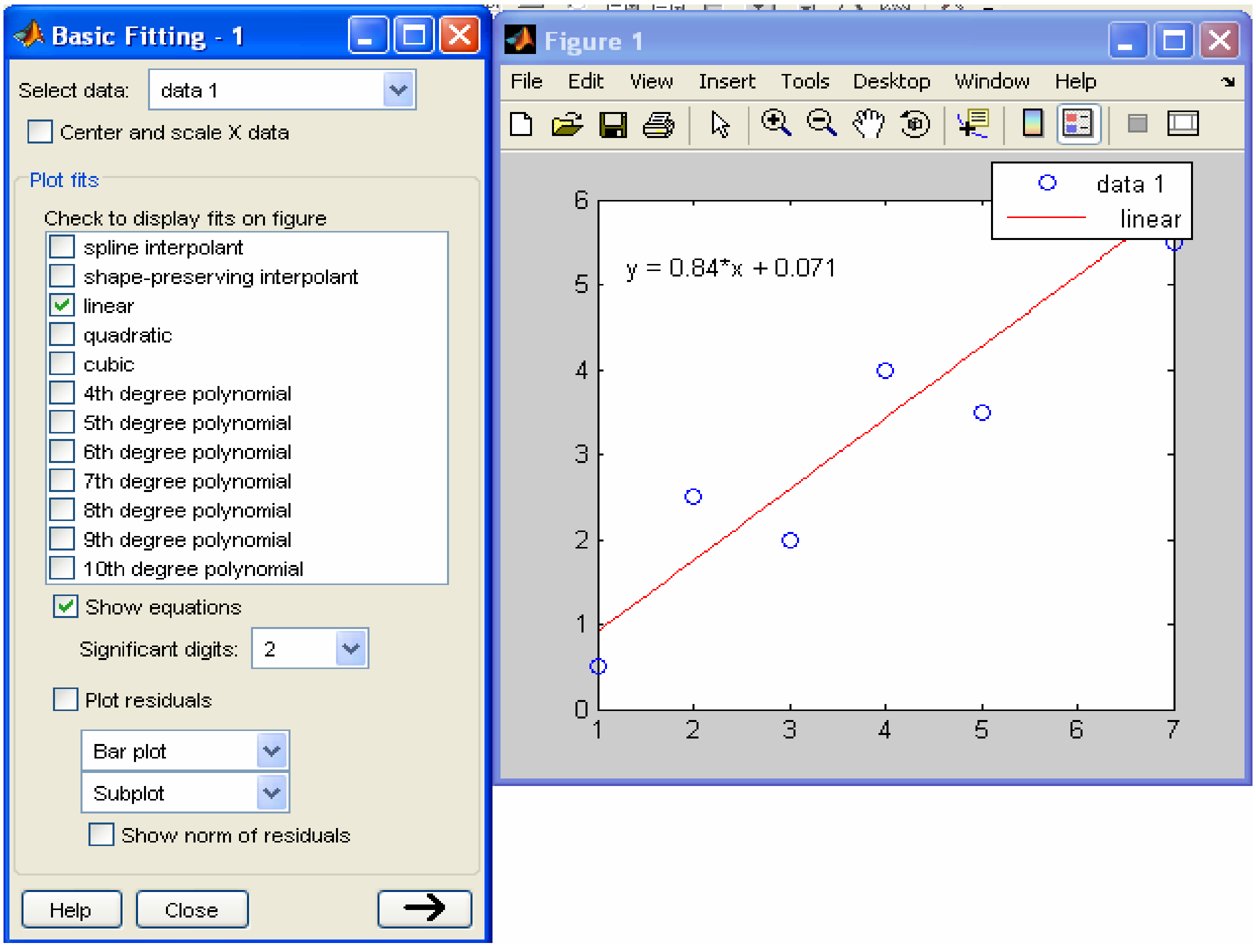


Figure 2: Use of Basic Fitting Options to fit any type of function to the data points

Linearization of Nonlinear Relationship

Linear regression is a powerful technique for fitting a best line to data. However it is dependent on the fact that the relationship between the dependent and independent variables should be linear. This is always not the case. In those cases we need to use polynomial regression. In some cases transformation can be used to express the data in a form that is compatible with linear regression.

One example is the exponential model

$$y = a_1 \exp(b_1 x) \quad \ldots\ldots\ldots\ldots\ldots\ldots..(3)$$

where a_1 and b_1 are constants.

Another example of non linear model is the power equation

$y = a_2 x^{b_2}$ ……………………(3)

where a_2 and b_2 are constants.

Nonlinear regression techniques are available to fit these equations to experimental data directly. However a simpler alternative is to use mathematical manipulations to transform the equations to linear forms. Then simple linear regression can be used to fit the equation to data. For example equation (2) can be linearized by taking its normal logarithm to yield

$\ln y = \ln a_1 + b_1 x$ …………..…..(4)

Thus a plot of lny vs. x will yield a straight line with slope of b_1 and an intercept of $\ln a_1$
Similarly equation (3) can be linearized by taking its logarithm to give

$\ln y = \ln a_2 + b_2 \ln x$ ………..….(5)

Thus a plot of lny vs. ln x will yield a straight line with slope of b_2 and an intercept of $\ln a_2$

Lab Task 1:

Fit the equation $y = a_2 x^{b_2}$ to the following data:

X	1	2	3	4	5
Y	0.5	1.7	3.4	5.7	8.4

Ans: $a_2 = 0.5$, $b_2 = 1.75$

Hints: First, find *ln y* and *ln x* for all points. Then using these converted points fit a straight line by linear regression [see code in figure (1)] and then find slope b_2 and intercept $\ln a_2$,and finally find a_2 using exp ($\ln a_2$) function.

Caution: In MATLAB ln(x) is represented as log (x) and log (x) is represented as log10(x)

Solving Linear Algebraic Equation with MATALB

A 3 x 3 set of linear equation can be expressed as,

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

Which can be expressed as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

or, [A][x]=[b]

$$[\mathrm{A}][\mathrm{x}] = [\mathrm{b}]$$

$$\Rightarrow [\mathrm{A}]^{-1}[A][x] = [A]^{-1}[b]$$

$$\Rightarrow [x] = [A]^{-1}[b] ....(6)$$

Therefore the equation has been solved for [x].

It should be noted that matrix A must be square. If the system has more equations (rows) than unknowns (columns) then the system is called overdetermined. Conversely a system with fewer equations (rows) than unknowns (columns) is called underdetermined. These two types of systems can not be solved by this method.

Solving by MATLAB

MATLAB uses two direct ways to solve systems with linear algebraic equation.

The first and <u>most efficient</u> way is to employ backslash or "left-division" operator

X=A\ b ;

The second and <u>less efficient [do not use this method]</u> method is to use matrix inversion:

X=inv (A)*b;

Example 2: Solve the following system of equations

$$5x_1 - 10x_2 + x_3 = 45$$
$$-2x_1 + 9x_2 + 6x_3 = -1$$
$$x_1 + x_2 = -1$$

Solution:

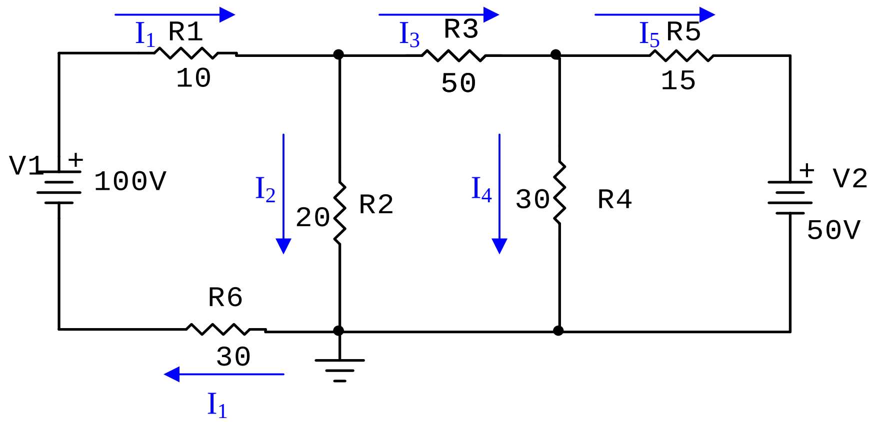
File Edit Text Cell Tools Debug Desktop Window Help

Stack: Base

```
clc;
clear all;
A=[5 -10 1; -2 9  6; 1 1 0];  % Matrix representing the co efficients
b=[45 -1 -1]';   % column vector reprenting the constants (USE TRANSPOSE operator)
x=A\b            % using left division operator to find solution
```

Figure 3: Solving linear Algebraic equations by left division operator

Lab Task 2: For the following circuit find the current passing through each resistor [Currents I_1 to I_5].

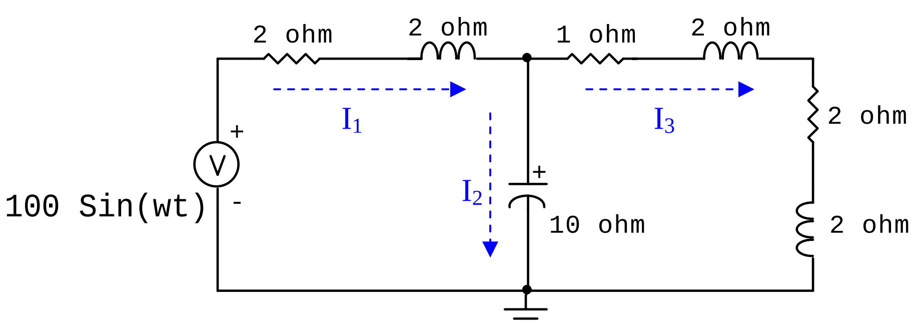


Hints: There are 5 unknowns here, so 5 equations will be required. To obtain 5 equations write KVL in the three loops and write two node equations using KCL. So you will obtain a 5 X 5 matrix. Then do as in figure 3.

$$\text{Ans: } \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix} = \begin{bmatrix} 1.667 \\ 1.667 \\ 0 \\ 1.111 \\ -1.111 \end{bmatrix}$$

Home Work:

[1]. For the following circuit find I_1, I_2 and I_3. using Matrix left division method.



Hints: Represent the source as V=100-0i, impedance Z_1=2+2i , Z_2= -10i and so on. Write three equations and obtain the 3 x 3 matrix. Here the matrix elements are all imaginary.

Ans: I_1=8.3691 - 5.1502i , I_2 = 0.6438 + 7.2961i, I_3=7.7253 -12.4464i

[2]. For the problem in lab task 1, page 5, fit the equation $y = a_1 \exp(b_1 x)$ to the data points and obtain the constants a_1 and b_1

For both the problems submit the printed M files including the results in the command window. For problem [1] also show the derivation of equations that you used in coding.