###### University of Dhaka

#### Department of Electrical and Electronic Engineering

EEE-3102: Numerical Technique Laboratory

Experiment-04: Determination of model parameters by fitting nonlinear data with a specific function.

|  |  |  |
| --- | --- | --- |
| Section | : |  |
| Registration No.. | : |  |
| Name | : |  |
| Date | : |  |

**Theory for curve fitting:**

Curve fitting is a technique of finding an algebraic relationship that “best” (in least squares sense) fits a given set of data. Unfortunately, there is no magical function (in MATLAB or otherwise) that can give the relationship if we simply supply the data. We have to have an idea of what kind of relationship might exist between the input data and the output data. However, if we do not have the firm idea but have data that we trust, MATLAB can help us in exploring the best possible fit. A detailed description of curve fitting techniques has been discussed in the previous laboratory experiment.

Purpose:

In the previous experiment, given data sets (xi, yi), i = 1, 2, ……n has been fitted with a polynomial function of the form yi = a0+a1x+ a2x2+……+ anxn to find the model parameters a0, a1, a2,……. an using ‘polyfit’ built-in Matlab function. In can easily be observed that this fitting gives reasonable accuracy (i.e., MSE) when the data only follows such polynomial function. But this is not the case in many situations. In such situations, data are needed to be fitted with a specific function to obtain a reasonable accuracy. This type of curve fitting can easily be done using ‘lsqcurvefit’ built-in Matlab function which is discussed in this experiment.

**Fitting data with specific function:**

Sometimes, we may want to fit the experimental data with a given model (may be different from polynomial model). In such case we can use ‘lsqcurvefit’ matlab function. The function can actually be used for any curve fitting. First, we want to fit a data set having polynomial function using ‘polyfit’ and then we will try to fit of same data set with ‘lsqcurvefit’ Matlab function for the comparison.

**Example-1:** For this, try the following code:

clc;close all;

x=0:0.01:5; y=2\*x.^2+3\*x+4; % x and y data

db=30; yo=awgn(y,db,'measured'); % noisy data

%%%% polyfit %%%%%

ap=polyfit(x,yo,2) %% model parameters for n=2

yp=polyval(ap,x); %% fitted data

MSEp=mean((yo-yp).^2) %% MSE

figure;

plot(x,yo,'b',x,yp,'r')

xlabel('value of x');ylabel('value of y')

legend('observed','polyfitted');grid on

**Q.1:** (i) What are the model parameters?

(ii) What is the MSE?

**Example-2**: Now try the following code which is simply followed by the code in **example-1**.

clc;close all;

x=0:0.01:5; y=2\*x.^2+3\*x+4; % x and y data

db=30; yo=awgn(y,db,'measured'); % noisy data

%%%% polyfit %%%%%

ap=polyfit(x,yo,2) %% model parameters for n=2

yp=polyval(ap,x); %% fitted data

MSEp=mean((yo-yp).^2) %% MSE

%%% lsqcurvefit %%%%%%%

F = @(a,x)(a(1)\*x.^2+a(2)\*x+a(3)); %% 2nd order polynomial

ai=[1 1 1]; %% model parameters, needs to initialize/guess this

[al] = lsqcurvefit(F,ai,x,yo) %% model parameters

yl=F(al,x); %Fitted data

MSEl=mean((yo-yn).^2) %% MSE

figure;

plot(x,yo,'b',x,yp,'r',x,yl,'g')

xlabel('value of x'); ylabel('value of y')

legend('observed','polyfitted','lsqfitted');grid on

**Q.2:** (i) What are ‘ap’ and ‘al’ in the above code? What are their values?

(ii) What are ‘MSEp’ and ‘MSEl’ in the above code? What are their values?

(iii) Comment on the results obtained in (i) and (ii).

From **example-2**, it is observed that curve fitting using ‘polyfit’ and ‘lsqcurvefit’ matlab functions give the same fitting performance. But notice that in the example, the data sets used follow a polynomial function. If the data sets do not follow a polynomial function, the result obtained using ‘polyfit’ may not give accurate fitting. But the fitting with ‘lsqcurvefit’ may give reasonable performance.

**Example-3**: Now try the following code which is similar to that in **example-2**, but the function is different (e.g., not polynomial)

clc; close all;

x=0:0.001:3; y=4.^x+1; % x and y values

dB=30; yo=awgn(y,dB,'measured');

%%%% polyfit %%%%%%

ap=polyfit(x,yo,2)

yp=polyval(ap,x); MSEp=mean((yo-yp).^2)

%%%% lsqcurvefit %%%

F = @(a,x)(a(1).^x+a(2)); %% 2nd order polynomial

ai=[1 1]; %% model parameters, needs to initialize/guess this

[al] = lsqcurvefit(F,ai,x,yo) %% model parameters

yl=F(al,x); %Fitted data

MSEl=mean((yo-yl).^2) %% MSE

figure;

plot(x,yo,'k',x,yp,'b',x,yl,'r')

xlabel('value of x'); ylabel('value of y')

legend('observed','polyfitted','lsqfitted');grid on

**Q.3:** (i) What are ‘ap’ and ‘al’ in the above code? What are their values?

(ii) What are ‘MSEp’ and ‘MSEl’ in the above code? What are their values?

(iii) Comment on the results obtained in (i) and (ii).

**Exercise-1:** A Gaussian distribution function is given by. Let a = 10, b = 50 and c = 0.1 for with a step of 0.1. Add noise to y and produce yo such that SNR = 30 dB.

1. Draw a graph to show the ‘yo’ and polyfitted data ‘yp’ versus ‘x’ in a plot for n = 2. What is the MSE?
2. Draw a graph to show the ‘yo’ and polyfitted data ‘yp’ versus ‘x’ in a plot for n = 5. What is the MSE?
3. Draw a graph to show the ‘yo’ and polyfitted data ‘yp’ versus ‘x’ in a plot for n = 10. What is the MSE?
4. Comment on the graphs and MSEs obtained in (i), (ii) and (iii).
5. Fit the data sets using ‘polyfit’ function with different order of n = 1, 2 ….10 and plot a graph to show the variation of MSE with order ‘n’. What is the value of MSE for n=10?
6. Now fit the data sets using ‘lsqcurvefit’ Matlab function and draw a graph to show the ‘yo’ and lsqcurvefitted data ‘yl’ versus ‘x’ in a plot. What is the MSE now?
7. Make an overall comment of curve fitting techniques using ‘polyfit’ and ‘lsqcurvefit’ Matlab functions

**The End**

**References**

[1] S.K. Mitra, Digital Signal Processing, 3rd Edition, McGraw-Hill Education (Asia), 2009.

[2] J.G. Proakis and D.G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, 4th Edition, Pearson International Edition, 2007.

[3] McClellan, Schafer and Yoder, *DSP FIRST: A Multimedia Approach*. Prentice Hall, Upper Saddle River, New Jersey, 1998 Prentice Hall.

[4] *Using Matlab*, The Math Works Inc.