**Curve Fitting Exercise in MATLAB**

# Lab exercise :

1. Write a function that takes the data (in whatever form you want) as an input, fits the data with a straight line using polyfit and polyval, and returns the parameters and the model values at the datapoints. Plot both the model and the data on the same plot. **Include this plot and the parameters in your lab write up.**

Coding :

x = [0 0.25 0.5 1 2 3 4 5 6 8 10];

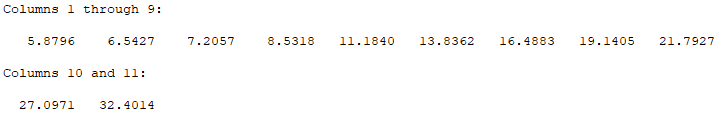
     y = [2.5 3.6 5.3 9.5 14.0 16.5 18.8 21.5 23.2 26.8 28.4];

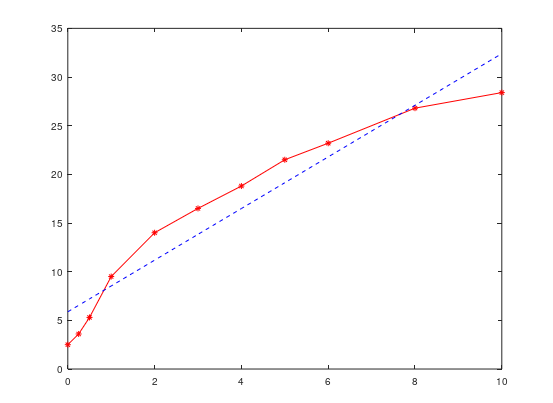
     p = polyfit(x, y, 1);

     z = polyval(p, x);

     plot(x, y, 'r-\*', x, z, 'b--');

     z

>> z = 



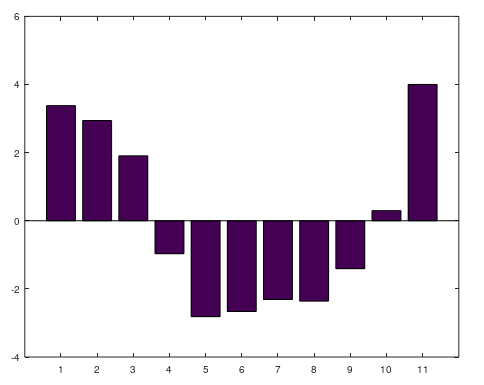
1. Modify the function created in question 1 to compute and return the residuals. Plot the residuals of the model as a bar graph. **Include this plot in your lab write up. (**Hint: Check out the command “bar”)

Coding :

p = polyfit(x, y, 1);

z = polyval(p, x);

bar(z - y);



1. Compute the sum of the squares of all of the residuals. **Include the computed value in your lab write up.**

Coding :

x = [0 0.25 0.5 1 2 3 4 5 6 8 10];

     y = [2.5 3.6 5.3 9.5 14.0 16.5 18.8 21.5 23.2 26.8 28.4];

     p = polyfit(x, y, 1);

     z = polyval(p, x);

     sumerror = sum(z - y)

>> sumerror = 2.3093e-14

1. Modify the main program and/or the function to fit the data using a **2nd** and **3rd** order polynomial as well. **Include the plots, residuals, and sum of squares of residuals in your write-up**. Note: if you already know how to use flow control, you can create an iteration loop to fit the given data with a 1st , 2nd and a 3rd order polynomial, plot the raw data and the model for each of them, and store the residuals for each of the models.

Coding :

for i=1:3

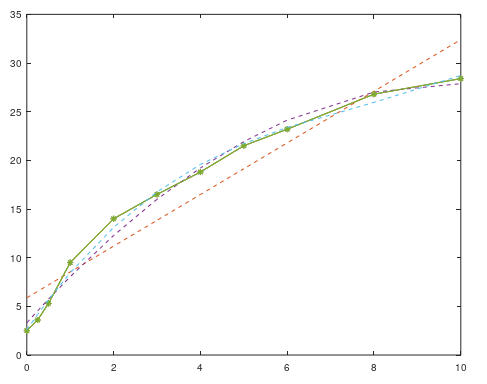
        p = polyfit(x, y, i);

        z = polyval(p, x);

        plot(x, y, '-\*', x, z, '--');

        hold on;

    end



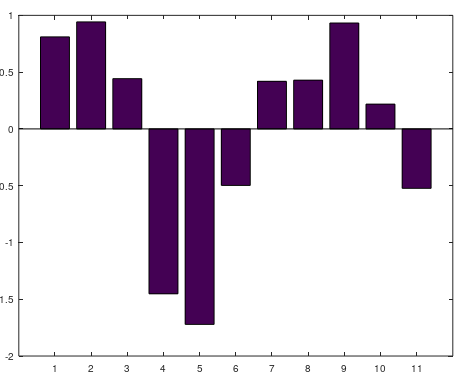
**For i = 2**

p = polyfit(x, y, 2);

         z = polyval(p, x);

         bar(z - y);

         sumerror = sum(z - y)



>> sumerror = -2.2204e-15

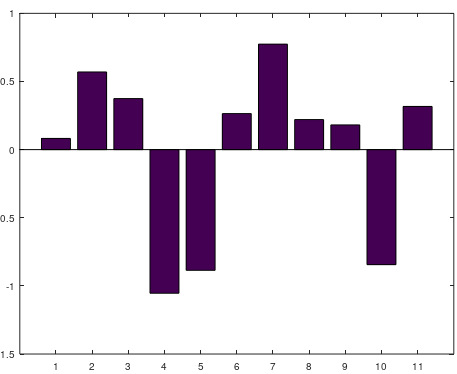
**For i = 3**

p = polyfit(x, y, 3);

         z = polyval(p, x);

         bar(z - y);

         sumerror = sum(z - y)



>> sumerror = 7.1498e-14

1. What is the order of the polynomial that gives the best fit in terms of the lowest sum of square of residuals?

Answer :

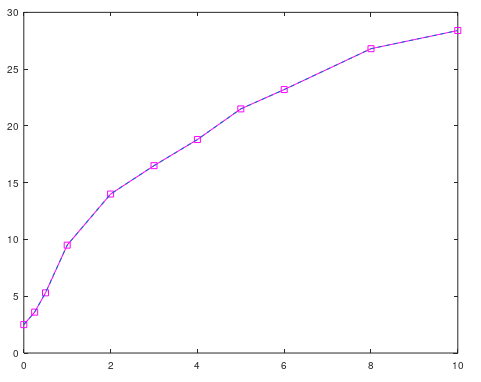
the order of the polynomial that gives the best fit in terms of the lowest sum of square of residuals is **n=10**

Coding

p = polyfit(x, y, 10);

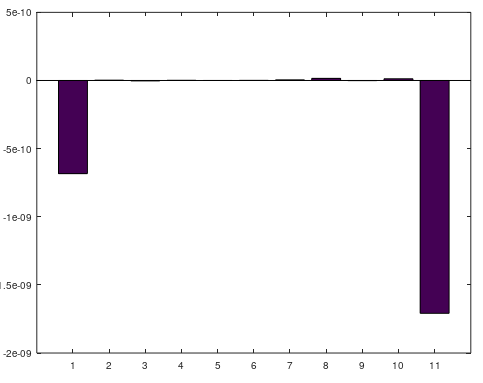
     z = polyval(p, x);

     plot(x, y,'m-s', x, z,'--')



bar(z - y);

     sumerror = sum(z - y)



>> sumerror = -0.0000000023593

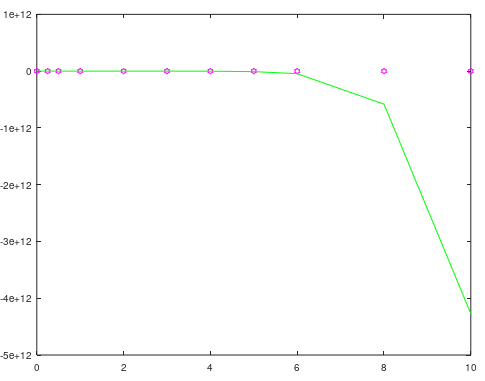
1. Using MATLAB, fit the same data to this new model, using the initial estimates . Because this equation is not a polynomial function, you cannot use polyfit, and will instead use a more generic minimization function called fminsearch (see hints at end). Plot the best model prediction and the data on the same plot. **Include this plot in your lab write up**. **What are the best-fit parameters?**

Coding :

x = [0 0.25 0.5 1 2 3 4 5 6 8 10];     p1 = arrayfun(@(t) fminsearch(@(a) 10.\*(1- exp(-a.\*t)), 2), x)

   z1 = polyval(p1, x);

plot(x, y, 'mh', x, z1, 'g')



Answer

The best-fit parameters is x = [0 0.25 0.5 1 2 3 4 5 6];