1 Hands-on–Interpolation and Curve Fitting with MATLAB III

1. **Fitting a cubic to the data by using MATLAB**. For the given data points;

x	Y
0.000	1.500
0.142	1.495
0.285	1.040
0.428	0.821
0.571	1.003
0.714	0.821
0.857	0.442
1.000	0.552

   - Evaluate the cubic on the x data and plot

```
>> x=[0 0.142 0.285 0.428 0.571 0.714 0.857 1]'
>> Y=[1.5 1.495 1.04 0.821 1.003 0.821 0.442 0.552]'
>> p = polyfit(x,Y,3)
p =   -0.3476    0.9902   -1.6946    1.5518
>> f = polyval(p,x)
>> plot(x,Y,'o',x,f,'-')
```

2. **Fitting a non-linear curve to the data with least-square method**.

   - Use the data in the previous item.
   - We will fit $y(x) = \alpha e^{\beta x}$.
   - Repeat each of the steps given in the following solution by hand.
     Solution:

     (a) First, we should compute a new table with $z(x) = ln y(x)$

```
>> Y=[1.5 1.495 1.04 0.821 1.003 0.821 0.442 0.552]'
>> z=log(Y)
```

Then our new data points;

x	z
0.000	0.4055
0.142	0.4021
0.285	0.0392
0.428	-0.1972
0.571	0.0030
0.714	-0.1972
0.857	-0.8164
1.000	-0.5942

(b) Construct the normal equations (with $A = ln\alpha$ and $C = \beta$)

$$y(x) = \alpha e^{\beta x}$$
$$lny(x) = ln\alpha + \beta x$$
$$z = A + Cx$$
$$S = \sum_{i=1}^{N} (z_i - Cx_i - A)^2$$
$$\frac{\partial S}{\partial C} = 0 = \sum_{i=1}^{N} 2(z_i - Cx_i - A)(-x_i)$$
$$\frac{\partial S}{\partial A} = 0 = \sum_{i=1}^{N} 2(z_i - Cx_i - A)(-1)$$

(c) Dividing each of these equations by -2 and expanding the summation, we get the so-called *normal equations*

$$\begin{aligned} C\sum x_i^2 + A\sum x_i &= \sum x_i z_i \\ C\sum x_i + AN &= \sum z_i \end{aligned}$$

(d) Solve these normal equations to find A and C

```
>> x=[0 0.142 0.285 0.428 0.571 0.714 0.857 1]'
>> Y=[1.5 1.495 1.04 0.821 1.003 0.821 0.442 0.552]'
>> z=log(Y)
>> sum(x'*x)
>> sum(x')
>> sum(x'*z)
>> sum(z')
>> A=[ 2.8549 3.997; 3.997 8]
>> B=[-1.4491 -0.9553]'
>> X=A\B
X =
   -1.1328
    0.4466
```

(e) So; we obtained $C = -1.1328$ and $A = 0.4466$, we should convert back to the original variables. Convert back to the original variables we have

```
>> exp(0.4466)
ans =    1.5630
```

$$z = 0.4466 - 1.1328x, \Rightarrow y = 1.563 * e^{-1.1328x}$$

(f) Plot Y vs x and y vs x then compare them. For plotting (see Fig. 1);

```
>> y=1.5630*exp(-1.1328*x)
>> plot(x,Y,'o',x,y,'-')
```

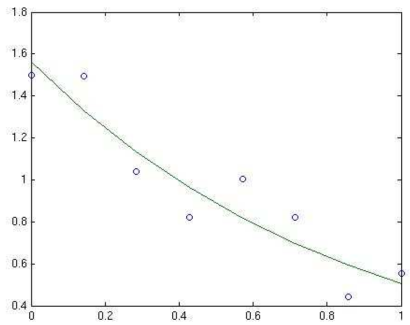

Figure 1: plot(x,Y,'o',x,y,'-').

- Compare this least-square polynomial results with the built-in MATLAB functions results in the previous item (item 1), see Fig.2.

```
>> plot(x,Y,'o',x,f,'-',x,y,'+')
```

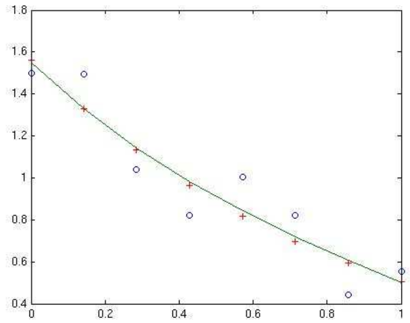

Figure 2: plot(x,Y,'o',x,f,'-',x,y,'+').