

M331 Hmwk # 4 Solutions (3.1, 3.2, 3.3, 3.5, 3.6, 3.7, 3.8, 3.9)

Average: 11/15

Median: 11.5/15

Notes: Be sure to answer all questions relating to the MATLAB assignments. A summary paragraph of the question will greatly enhance your chances of receiving full credit. Make sure all answers are easily identified.

$$\frac{3.1}{(2 \text{ pts})} \quad E = \sum (y_i - c_0 x_i^{-3/2} - c_1 x_i^{5/2})^2$$

$$\frac{\partial E}{\partial c_0} = \sum 2(y_i - c_0 x_i^{-3/2} - c_1 x_i^{5/2})(-x_i^{-3/2}) = 0$$

$$\Rightarrow \sum y_i x_i^{-3/2} = c_0 \sum x_i^{-3} + c_1 \sum x_i \quad (1)$$

$$\frac{\partial E}{\partial c_1} = \sum 2(y_i - c_0 x_i^{-3/2} - c_1 x_i^{5/2})(-x_i^{5/2}) = 0$$

$$\Rightarrow \sum y_i x_i^{5/2} = c_0 \sum x_i + c_1 \sum x_i^5 \quad (2)$$

In matrix form:

$$\begin{bmatrix} \sum x_i^{-3} & \sum x_i \\ \sum x_i & \sum x_i^5 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} \sum x_i^{-3/2} y_i \\ \sum x_i^{5/2} y_i \end{bmatrix}$$

$$\text{Solving: (1)} \quad c_0 = \frac{\sum y_i x_i^{-3/2} - c_1 \sum x_i}{\sum x_i^{-3}} \quad (3)$$

Substituting:

$$\begin{aligned} (3) \text{ into } (2) \quad c_1 \sum x_i^5 &= \sum y_i x_i^{5/2} - c_0 \sum x_i \\ &= \sum y_i x_i^{5/2} - \left(\frac{\sum y_i x_i^{-3/2} - c_1 \sum x_i}{\sum x_i^{-3}} \right) \sum x_i \end{aligned}$$

$$= \sum y_i x_i^{5/2} - \frac{\sum y_i x_i^{-3/2} \sum x_i}{\sum x_i^{-3}} + \frac{c_1 (\sum x_i)^2}{\sum x_i^{-3}}$$

$$c_1 \left(\sum x_i^5 - \frac{(\sum x_i)^2}{\sum x_i^{-3}} \right) = \frac{\sum y_i x_i^{5/2} \sum x_i^{-3} - \sum y_i x_i^{-3/2} \sum x_i}{\sum x_i^{-3}}$$

$$c_1 = \frac{\sum y_i x_i^{5/2} \sum x_i^{-3} - \sum y_i x_i^{-3/2} \sum x_i}{\sum x_i^5 \sum x_i^{-3} - (\sum x_i)^2} \quad (4)$$

Sub in again gives:
(4) into (3)

$$c_0 = \frac{\sum y_i x_i^{-3/2} \sum x_i^5 - \sum x_i \sum y_i x_i^{5/2}}{\sum x_i^5 \sum x_i^{-3} - (\sum x_i)^2}$$

Problem 3.2: (2 pts)

Matlab code:

```
w = [3.5 4 6 25 103 117 200 252 300 437 1340 2000 2700 5000 22500 30000 33000  
50000 70000 100000 415000 450000 500000 3000000];  
r = [787 660 588 670 347 300 420 352 300 269 251 205 187 120 100 85 81 70 72 70 45  
38 40 48];
```

```
% use least squares on  $r=aw^n$   
% need to do following transformations so that  $y=mx+b$   
% here  $m=n$  and  $b=\log(a)$  will be our results so need to transform at end to get  $a$  &  $n$ .
```

```
x = log(w)';  
y = log(r)';
```

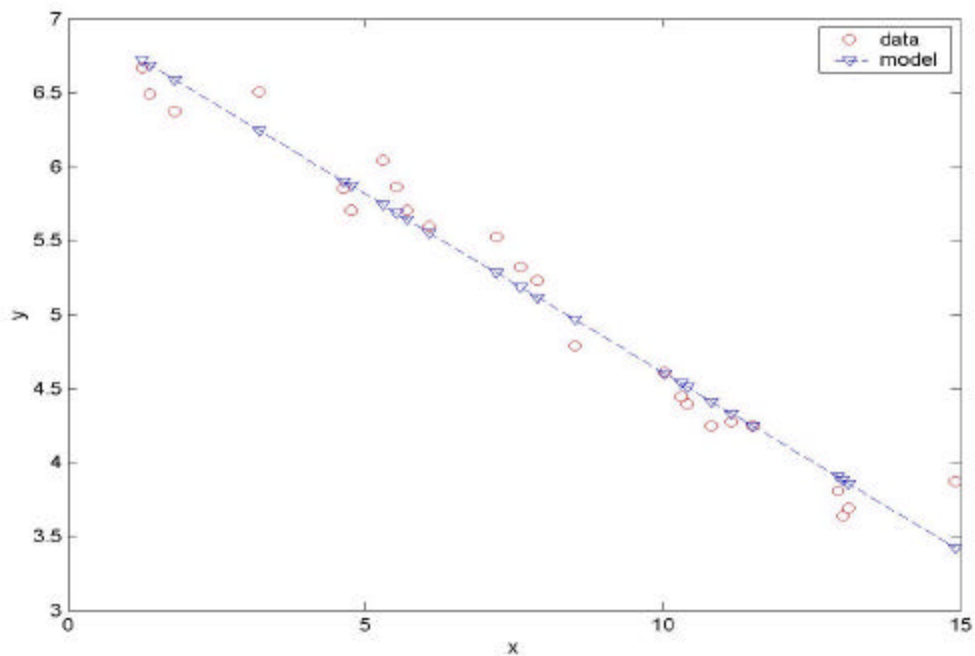
```
[m,b] = ls_normal (x,y); % to run this I didn't need to change ls_normal at all.
```

```
n=m  
a=exp(b)
```

Answers:

$a=1118.1$

$n=-0.241$ (note this is smaller than the dimensional analysis $-1/3$ answer)



Problem 3.3 (2 pts)

Matlab driver:

```
w = [3.5 4 6 25 103 117 200 252 300 437 1340 2000 2700 5000 22500 30000 33000  
50000 70000 100000 415000 450000 500000 3000000];  
r = [787 660 588 670 347 300 420 352 300 269 251 205 187 120 100 85 81 70 72 70 45  
38 40 48];
```

```
x = w.^(-1/3)'  
y = r'
```

```
[m,b] = ls_interp_3_3(x,y)
```

where ls_interp_3_3:

%Input:

% x is a column vector of domain (input) variables

% y is a column vector of range (output) variables

%Output:

% m is the slope of the line

% b is the intercept of the line

```
function [m,b] = ls_interp_3_3(x,y)
```

```
%Compute the matrix X
```

```
P = size(x,1)
```

```
cs = ones(P,1)
```

```
X1 = [cs x]
```

```
X2 = [cs x x.^2]
```

```
vec1 = X1\y
```

```
vec2 = X2\y
```

```
b = vec1(1) % coefficients for the linear least squares fit
```

```
m = vec1(2)
```

```
c0 = vec2(1) % coefficients for the quadratic least squares fit
```

```
c1 = vec2(2)
```

```
c2 = vec2(3)
```

```
E1=y - b - m.*x; % linear error
```

```
error1=sum(E1.^2)
```

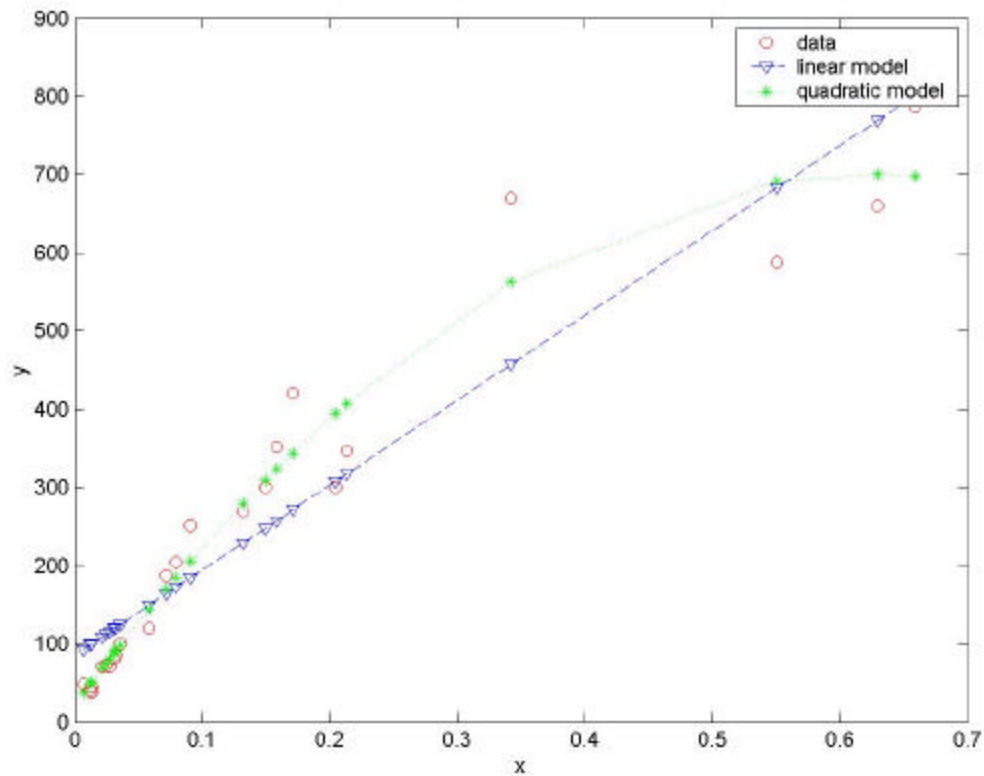
```
E2=y - c0 - c1.*x - c2.*(x.^2); % quadratic error
```

```
error2=sum(E2.^2)
```

```

%plot results
plot(x,y,'ro')
hold
plot(x,b+m.*x,'--v')
plot(x,c0+c1.*x+c2.*x.^2,'g:*')
legend('data','linear model','quadratic model')
xlabel('x')
ylabel('y')
hold off;

```



for quadratic fit: $c_0=22.71$, $c_1=2175.9$, $c_2=-1746.2$ and error = 55087
for linear fit: error = 130850.

So you can see that the error for the linear fit is more than twice the quadratic fit error.

3.5)
(1 pt)

$$y_1 = c_0 + c_1 x_1 + c_2 x_1^2$$

$$y_2 = c_0 + c_1 x_2 + c_2 x_2^2$$

⋮

$$y_p = c_0 + c_1 x_p + c_2 x_p^2$$

In matrix:

$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_p & x_p^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}$$

$$X c = Y$$

multiply both sides by X^T : $X^T X c = X^T Y$

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_p \\ x_1^2 & x_2^2 & \dots & x_p^2 \end{bmatrix} \begin{bmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_p & x_p^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_p \\ x_1^2 & x_2^2 & \dots & x_p^2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \sum_{i=1}^p 1 & \sum_{i=1}^p x_i & \sum_{i=1}^p x_i^2 \\ \sum_{i=1}^p x_i & \sum_{i=1}^p x_i^2 & \sum_{i=1}^p x_i^3 \\ \sum_{i=1}^p x_i^2 & \sum_{i=1}^p x_i^3 & \sum_{i=1}^p x_i^4 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^p y_i \\ \sum_{i=1}^p x_i y_i \\ \sum_{i=1}^p x_i^2 y_i \end{bmatrix}$$

which is
eqn 3.3

3.6)
(2 pts) Can derive either using partial derivatives of error function $E = \sum (y_i - c_0 - c_1 x_i - c_2 x_i^2 - c_3 x_i^3)^2$

$$\frac{\partial E}{\partial c_0} = \frac{\partial E}{\partial c_1} = \frac{\partial E}{\partial c_2} = \frac{\partial E}{\partial c_3} = 0$$

or using interpolation and matrix multiplication

$$X^T X c = X^T y : \begin{bmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_p \\ x_1^2 & \dots & x_p^2 \\ x_1^3 & \dots & x_p^3 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & \dots & 1 \\ x_1 & \dots & x_p \\ x_1^2 & \dots & x_p^2 \\ x_1^3 & \dots & x_p^3 \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix}$$

either way results:
$$\begin{bmatrix} p & \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 \\ \sum x_i^3 & \sum x_i^4 & \sum x_i^5 & \sum x_i^6 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \\ \sum x_i^3 y_i \end{bmatrix}$$

So for degree nine polynomial:

$$\begin{bmatrix} p & \sum x_i & \sum x_i^2 & \dots & \sum x_i^9 \\ \sum x_i & \dots & \dots & \dots & \sum x_i^{10} \\ \vdots & \dots & \dots & \dots & \vdots \\ \sum x_i^9 & \sum x_i^{10} & \dots & \dots & \sum x_i^{18} \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_9 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \vdots \\ \sum x_i^9 y_i \end{bmatrix}$$

3.7)
(2 pts)
$$P_3(x) = \frac{(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)} y_1 + \frac{(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)} y_2$$

$$+ \frac{(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)} y_3 + \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_1)(x_4-x_2)(x_4-x_3)} y_4$$

3.8)
(2 pts)
$$S(x) = \begin{cases} S_1(x) & \text{if } x_1 \leq x < x_2 \\ S_2(x) & \text{if } x_2 \leq x < x_3 \\ S_3(x) & \text{if } x_3 \leq x \leq x_4 \end{cases} = \begin{cases} \frac{1}{2}x & \text{if } 0 \leq x < 2 \\ -3x+7 & \text{if } 2 \leq x < 3 \\ 2x-8 & \text{if } 3 \leq x \leq 5 \end{cases}$$

Problem 3.9: (2 pts)

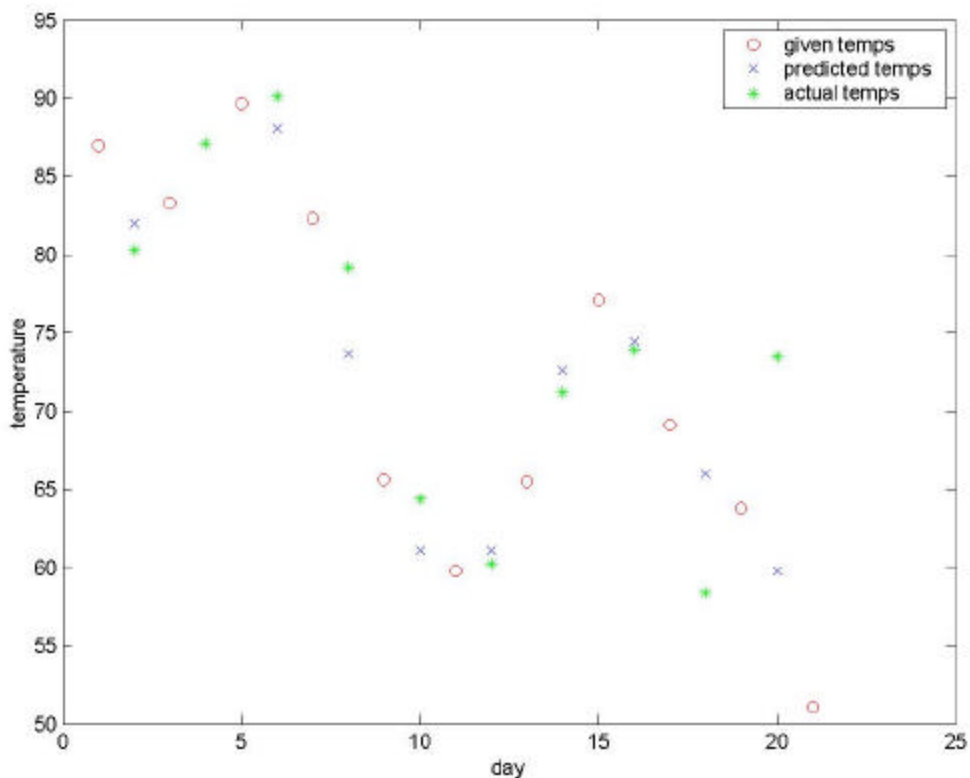
Matlab code:

```
%MODEL DATA (TRAINING DATA)
x = 1:2:21 %define the domain of the function - odd days
y = [87 83.3 89.7 82.3 65.6 59.8 65.5 77.1 69.1 63.8 51.1] %temperature at these days
y1 = [80.3 87.1 90.1 79.2 64.4 60.2 71.2 73.9 58.4 73.5] % actual temps on even days

%PREDICTED VALUES
xx = 2:2:20%domain values where we seek predictions - even days

%The spline function returns the predicted values yy at the domain values xx. The data
% x, y is used to build the model.
yy = spline(x,y,xx);

plot(x,y,'ro',xx,yy,'x',xx,y1,'g*')
legend('given temps','predicted temps','actual temps')
xlabel('day'); ylabel('temperature')
```



So you can see that some of the even days are predicted by the spline better than others. In particular the predictions at the end of the data set are not very good at all. It makes sense that the spline will not be able to pick up drastic temperature changes on the even days when the odd days show a trend that the even days do not follow.