Chapter 6 HW

Table of Contents

Lucas Gobaco

ENGI-111-01

Mahnaz Firouzi

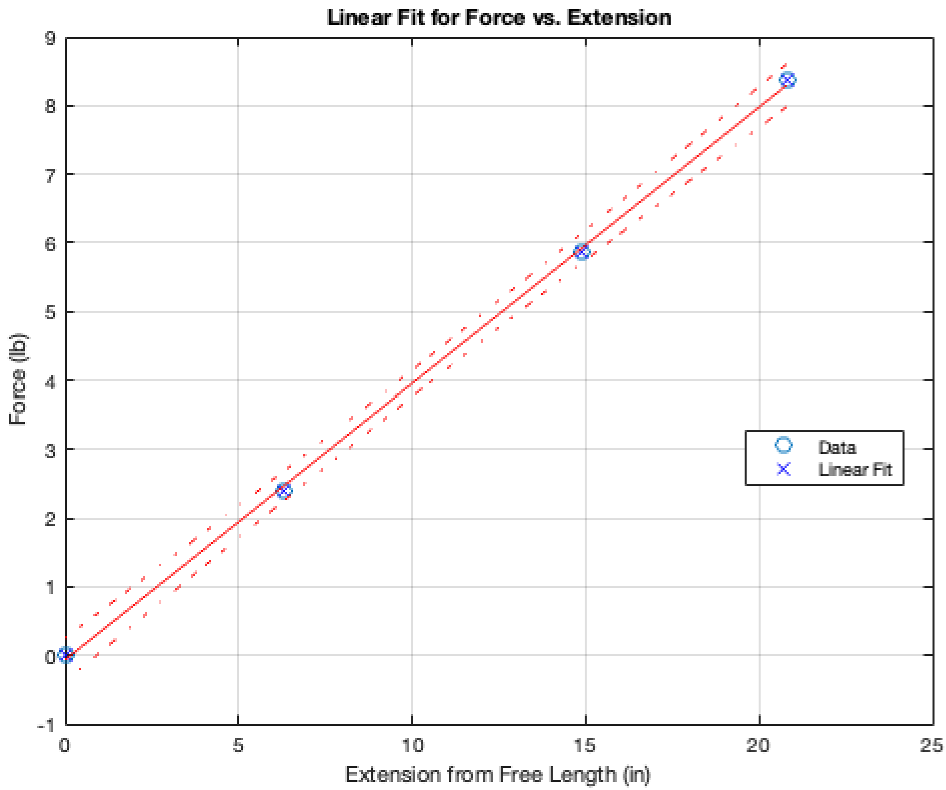
9 April 2024

Problem 1

```
force = [0; 2.40; 5.87; 8.36]; % Force (lb)
spring_length = [12; 18.3; 26.9; 32.8]; % Spring length (in)

% Calculate extension from free length
extension = spring_length - 12; % Extension from free length (in)

% Fit linear model
linear_model = fitlm(extension, force);

% Display the linear model equation
disp('Linear model equation:')
disp(linear_model)

% Plot the data and the linear fit
plot(extension, force, 'o', 'MarkerSize', 8);
hold on;
plot(linear_model);
xlabel('Extension from Free Length (in)');
ylabel('Force (lb)');
title('Linear Fit for Force vs. Extension');
legend('Data', 'Linear Fit', 'Location', 'best');
grid on;
hold off;
```

```
Linear model equation:

Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate        SE        tStat       pValue
                   ________    ________    ________    ________
```

```
    (Intercept)     -0.064052     0.073591     -0.87038       0.47586
    x1                0.40205    0.0055856       71.981    0.00019295


Number of observations: 4, Error degrees of freedom: 2
Root Mean Squared Error: 0.0889
R-squared: 1,  Adjusted R-Squared: 0.999
F-statistic vs. constant model: 5.18e+03, p-value = 0.000193
```



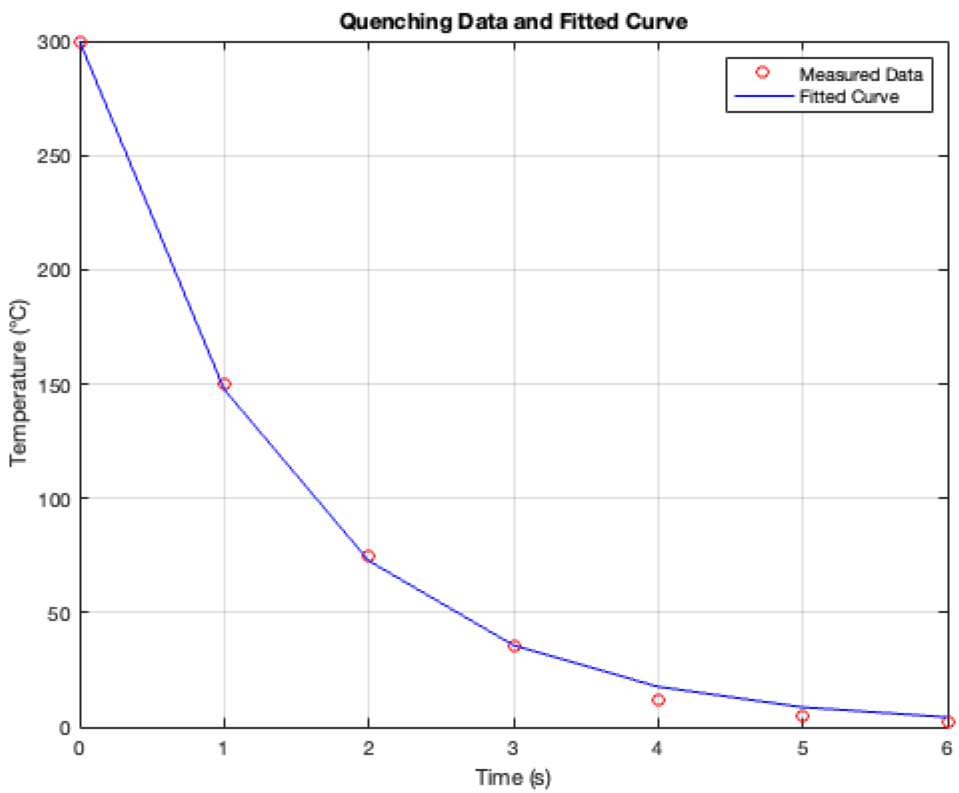
Problem 5

```
time = [0 1 2 3 4 5 6];
temp = [300 150 75 35 12 5 2];

% Initial temperature and bath temperature
T0 = temp(1);
Tb = 0;

% Define the cooling function
cooling_function = @(k, time) Tb + (T0 - Tb) * exp(-k * time);

% Use the curve fitting tool
k_fit = lsqcurvefit(cooling_function, 0.5, time, temp); % 0.5 is an initial
guess for k
```

```
% Generate fitted values
temp_fit = cooling_function(k_fit, time);

% Plot the data and fitted curve
figure;
plot(time, temp, 'ro', 'DisplayName', 'Measured Data'); % Original data as
red circles
hold on;
plot(time, temp_fit, 'b-', 'DisplayName', 'Fitted Curve'); % Fitted data as
blue line
xlabel('Time (s)');
ylabel('Temperature (°C)');
legend;
title('Quenching Data and Fitted Curve');
grid on;
```

Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative to
its initial value is less than the value of the function tolerance.



Problem 9

a.

```
% degree 1
A = 0:1:9;
B = [130 115 110 90 89 89 95 100 110 125];
p = polyfit(A,B,1);

x1=linspace(0,20,1000);
y1=polyval(p,x1);

figure
plot(x1,y1)
hold on

% degree 2
p=polyfit(A,B,2);

x1=linspace(0,20,1000);
y1=polyval(p,x1);

plot(x1,y1)
hold on
scatter(A,B)

% degree 3
hold on
p=polyfit(A,B,3);
x1=linspace(0,20,1000);
y1=polyval(p,x1);

plot(x1,y1)
hold on
scatter(A,B)

% degree 4
hold on
p=polyfit(A,B,4);
x1=linspace(0,20,1000);
y1=polyval(p,x1);

plot(x1,y1)
hold on
scatter(A,B)
legend('1st degree ','2nd degree','3rd degree','4th degree ')

% b.
hold on
p=polyfit(A,B,3);
x1=linspace(0,20,10000);
y1=polyval(p,x1);
```

```
disp('The amount of additive to minimize Time is : ')
for i=1:length(x1)
    if(y1(i)==min(y1))
        disp(x1(i))
    end
end

legend('hide')
clf
```

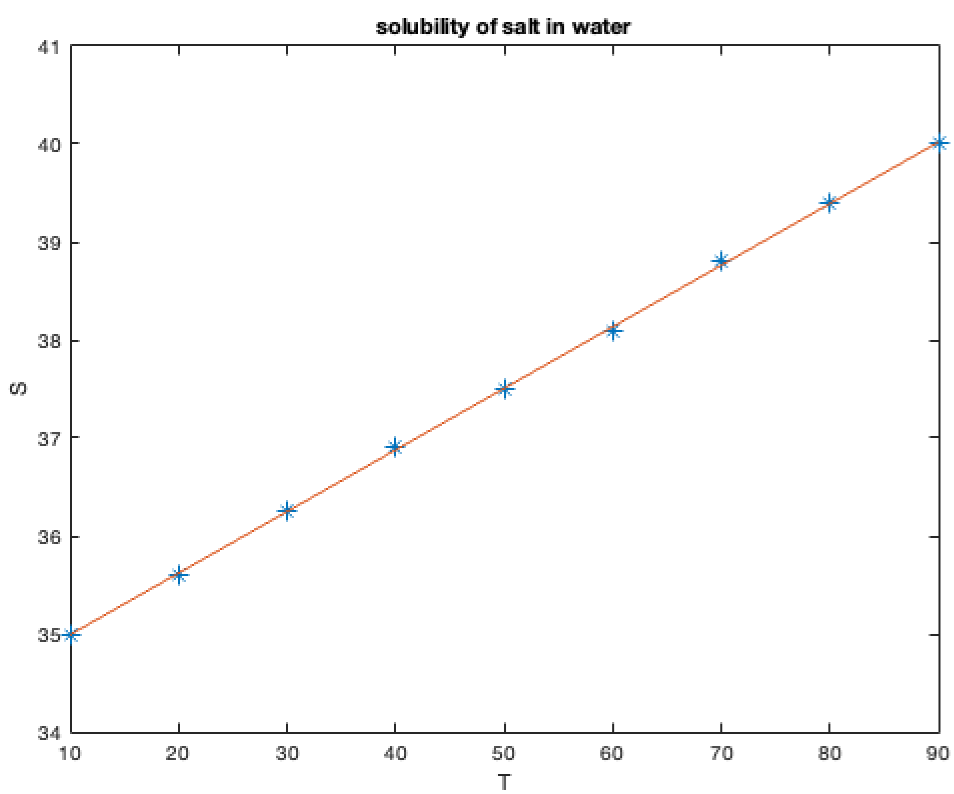
```
The amount of additive to minimize Time is :
    4.7245
```

Problem 15

```
%write the temp
T=10:10:90;

%enter the data for solubility
S=[35,35.6,36.25,36.9,37.5,38.1,38.8,39.4,40];

%use polyfit command
c = polyfit(T,S,1)
Se = c(1)*25 + c(2)
```

```
%give increment in temp
Tp =10:.1:90;
Splot = c(1)*Tp+c(2);

%plot the curve
plot(T,S,'*',Tp,Splot)

%give x axis label
xlabel('T')

%give y axis label
ylabel('S')

%give title to plot
title('solubility of salt in water')
```

```
c =

    0.0628   34.3639


Se =

   35.9347
```



Published with MATLAB® R2023b