**ChE 310 Problem Set 8 Due Wed 3/27/19**

Collect all m-files in a single .zip file and upload the .zip file to the course webpage by midnight on Wednesday, March 27, 2019. Please note any collaborations in comments. Each student must upload their own unique copy of the work.

**8\_1 Regression (use a method of your choice)**

1. Problem 15.18 from the textbook. Be sure to report the correct units on your calculated coefficients A1 and A2.

***Solution:***

%PS8\_1A.m

%Chapra 15.18

clear

%data

P = [.985; 1.108; 1.363; 1.631]; %atm

V = [25000; 22200; 18000; 15000]; %mL

%constants

T = 303; %K

R = 82.05; %mL atm / K

%Let's group "known" dependent values into one vector y

y = P.\*V/(R\*T)-1;

%Use generalized linear least squares

g1 = @(x) 1./x; %pairs with A1

g2 = @(x) 1./x.^2; %pairs with A2

Z = [g1(V) g2(V)];

a = (Z'\*Z)\Z'\*y;

fprintf('The value of A1 is %4.4g mL.\n',a(1))

fprintf('The value of A2 is %4.4g mL^2.\n',a(2))

***Output:***

>> PS8\_1A

The value of A1 is -231.7 mL.

The value of A2 is -1.05e+05 mL^2.

1. Problem 15.23 from the textbook. Report cs, kmax, and the requested growth rate, with correct units.

%PS8\_1B

%Chapra 15.23

clear

%Nonlinear regression requested; we'll use fminsearch

%Known data

c = [0.5; 0.8; 1.5; 2.5; 4];

k = [1.1; 2.4; 5.3; 7.6; 8.9];

%a(1) = kmax; a(2) = cs

kstar = @(a) a(1)\*c.^2./(a(2)+c.^2);

S\_R = @(a) sum((k-kstar(a)).^2);

a\_bestfit = fminsearch(S\_R,[9 2]);

fprintf('The value of kmax is %4.4g /day.\n',a\_bestfit(1))

fprintf('The value of cs is %4.4g mg^2/L^2.\n',a\_bestfit(2))

k\_2 = a\_bestfit(1)\*2.^2./(a\_bestfit(2)+2.^2);

fprintf('The value of k at c=2 is %4.4g /day.\n',k\_2)

***Output:***

>> PS8\_1B

The value of kmax is 10.04 /day.

The value of cs is 2.018 mg^2/L^2.

The value of k at c=2 is 6.671 /day.

**8\_2 Linear vs. nonlinear regression**

Solve Chapra problem 14.31. Determine B and D using both a linear regression technique (after linearizing the model) and using a nonlinear regression technique on the original model equation. Plot the corresponding best-fit functions along with the original data. Ensure your plot is labeled (axes, legend); the plot is the only necessary output. Keep in mind that Andrade’s equation takes temperatures in K, while the table provides temperatures in .

***Solution:***

%PS8\_2.m

%Chapra 14.31

clear ; clf

%input known data

mu = [1.787; 1.519; 1.307; 1.002; 0.7975; 0.6529];

T\_c = [0 ; 5; 10; 20; 30; 40];

T = T\_c + 273.15;

%First, use general least squares.

%Linearize: ln(mu) = ln(D) + B\*(1/T)

%Our output data are

y = log(mu);

%basis functions: g0 = 1 ; g1 = 1/T ;

Z = [ones(size(T)) 1./T];

a = (Z'\*Z)\(Z'\*y) ;

%Convert a(1) and a(2) to the original parameters D and B

D\_lin = exp(a(1));

B\_lin = a(2);

fprintf('The value of D using linearization is %4.4g mN\*s/m^2.\n',D\_lin)

fprintf('The value of B using linearization is %4.4g K.\n',B\_lin)

%Now, let's use fminsearch

%Need to input B, D as one function to fminsearch. a = [B D]

mu\_star = @(a) a(2)\*exp(a(1)./T);

S\_R = @(a) sum((mu\_star(a)-mu).^2) ;

a\_best = fminsearch(S\_R,[2000 1e-5]) ;

B\_nonlin = a\_best(1);

D\_nonlin = a\_best(2);

fprintf('The value of D using nonlinear methods is %4.4g mN\*s/m^2.\n',D\_nonlin)

fprintf('The value of B using nonlinear methods is %4.4g K.\n',B\_nonlin)

plot(T,mu,'ko')

hold on

fplot(@(T) D\_lin\*exp(B\_lin./T),'r-')

fplot(@(T) D\_nonlin\*exp(B\_nonlin./T),'b-')

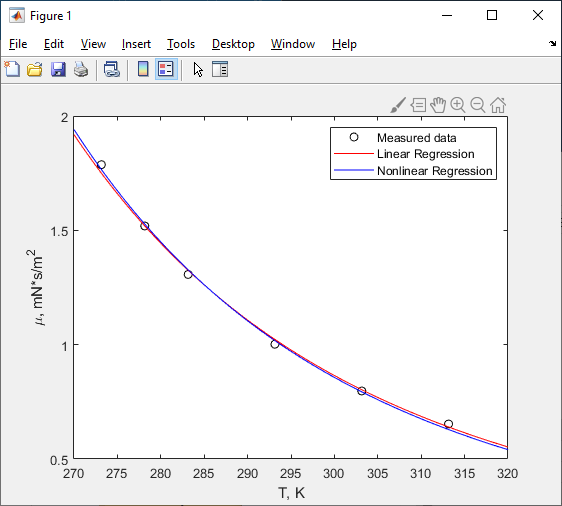
xlim([270 320])

xlabel('T, K')

ylabel('\mu, mN\*s/m^2')

legend('Measured data','Linear Regression','Nonlinear Regression')

***Output***



**8\_3**

The kinetic rate of a chemical equation is dependent on temperature (). The Arrhenius equation is used to show this dependence and when properly fitted also shows the activation energy () of the reaction:

where is the rate constant, is the pre-exponential factor, and is the universal gas constant.

Eleven experiments were conducted to determine the activation energy of the following reaction:

The data from these experiments are contained in the ‘kinetics’ zipped folder. Each experiment is its own \*.csv file with the first column = temperature (K) and the second column = reaction rate or k(T) in cm3/(molecule\*s) (Note: units matter!). Plot the raw data with the model fits (one subplot for each of the 11 experiments). Determine the activation energy (kJ/mole) for each experiment and report the average over all 11 experiments along with the associated 95% confidence interval; assume a normal distribution applies to the activation energy data.

For this problem you can linearize the equation or solve in the non-linear form. Which is better? Note: you can automate reading in the file name with the help of the num2str and strcat commands for the file name.

***Solution***

% PS8\_3.m

% Coded by Nigel F. Reuel

% This function solves question 5 on the problem set

%

% METHOD 1: Linearizing (how I solved it in 2016)

% Create an outer loop to step through the 11 data files

Ea\_vec = zeros(11,1);

for i = 1:11

Data = csvread(['R',num2str(i),'.csv']);

T = Data(:,1);

k = Data(:,2);

lnk = log(k);

InvT = 1./T;

R = 8.3144598; %J/K/mol

RkJ = R/1000; %kJ/K/mol

% Linearizing and rearranging the arrhenius equation we can get this

% Y = mx\*b form

% ln(k) = -Ea/R\*(1/T) + ln(A)

x = -InvT./RkJ;

p = polyfit(x,lnk,1); % NOTE: another way to do linear fit, returns the parameters directly

Ea\_vec(i,1) = p(1); % Units should be kJ/mol

end

% Print out the 11 values and the statistics

disp('For linearization followed by fit')

Ea\_vec

Avg = mean(Ea\_vec);

Conf = std(Ea\_vec)\*2;

fprintf('The average activation energy is %4.2fkJ/mol with +/- %4.2fkJ/mol for 95 percent confidence interval\n',Avg,Conf)

% Method 2 - Linearizing fits error of ln residuals. Not as

% accurate a fit as using the non-linear form. Let's do this!

Ea\_vec2 = zeros(11,1);

for i = 1:11

Data = csvread(['R',num2str(i),'.csv']);

T = Data(:,1);

k = Data(:,2);

R = 8.3144598; %J/K/mol

RkJ = R/1000; %kJ/K/mol

% The non-linear model

% k = A\*exp(-Ea/R\*(1/T))

% build in exp1 function is A\*exp(b\*x), so b = -Ea/R and x = 1/T

x = 1./T;

ExpFit = fit(x,k,'exp1');

MyCoeffs = coeffvalues(ExpFit);

Ea\_vec2(i,1) = MyCoeffs(2)\*-1\*RkJ; % Units should be kJ/mol

subplot(3,4,i)

plot(T,k,'o')

hold on

funcfit = @(T) MyCoeffs(1)\*exp(MyCoeffs(2)./T);

fplot(funcfit,[T(1) T(end)])

xlabel('Temp')

ylabel('k')

legend('Measured Points','Model Fit','Location','northwest')

hold off

end

disp('For non-linear fit')

Ea\_vec2

Avg = mean(Ea\_vec2);

Conf = std(Ea\_vec2)\*2;

fprintf('The average activation energy is %4.2fkJ/mol with +/- %4.2fkJ/mol for 95 percent confidence interval\n',Avg,Conf)

disp('Note: the non-linear fit is a better route and in this case also gives tighter confidence interval.')

***Output*:**

>> PS8\_3

For linearization followed by fit

Ea\_vec =

197.0224

264.3459

146.3205

158.7910

251.1172

171.3110

209.5324

251.1172

127.1643

213.7842

100.5027

The average activation energy is 190.09kJ/mol with +/- 108.04kJ/mol for 95 percent confidence interval

For non-linear fit

Ea\_vec2 =

176.2948

195.9645

124.2051

133.2853

227.7555

152.6589

203.9686

227.7555

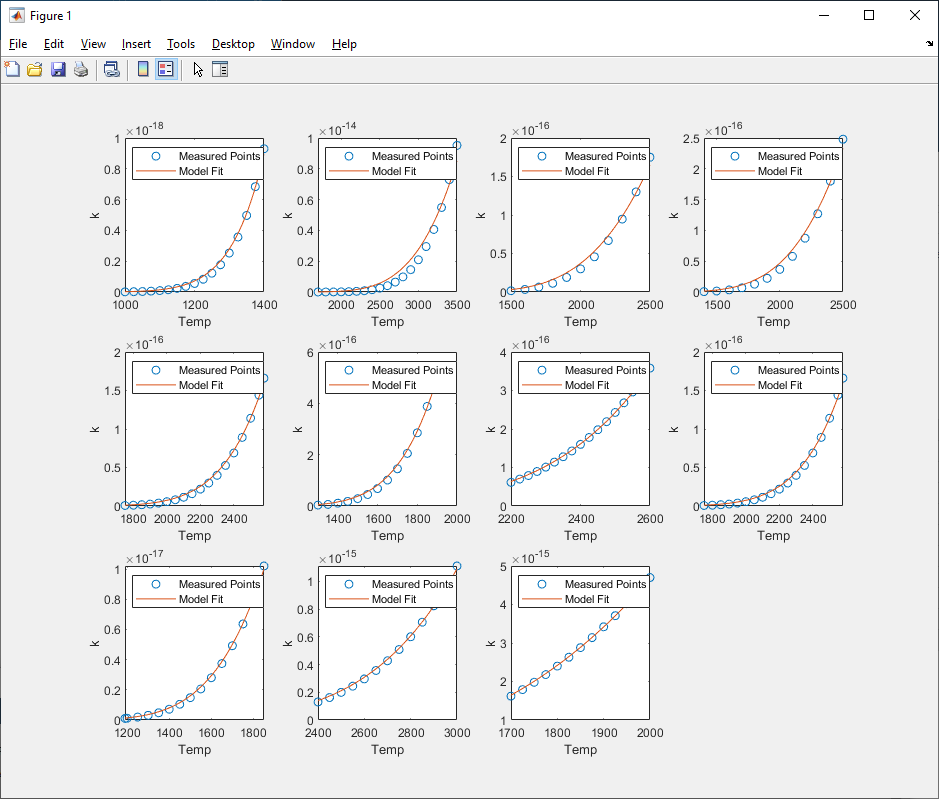
120.7660

204.8221

98.2187

The average activation energy is 169.61kJ/mol with +/- 91.81kJ/mol for 95 percent confidence interval

Note: the non-linear fit is a better route and in this case also gives tighter confidence interval.



**8\_4 Multivariable regression**

***Background***. As part of Dr. Roling’s postdoctoral research, he used linear regression to parameterize the energies of metal atoms in nanoparticles.[[1]](#footnote-1) These models fit 20 parameters corresponding to 20 independent variables in a 140-point alloy data set; this was done for each of 21 bimetallic combinations. Here, we’ll focus on a smaller subset of data.

In this model, each bond formed by a metal atom is considered to have its own unique energy. The total energy of nanoparticle is then determined by the sum of all its bond energies . For example, a metal dimer (two isolated bonded atoms) is composed of two atoms, each with a single bond; the energy of a metal dimer is therefore calculated as . A triangular arrangement of three atoms (each bound to two other atoms) has an energy calculated as .

***Problem Statement****.* We seek a model to calculate the energy of a small cluster of metal atoms as:

where are parameters and are independent variables denoting the number of each type of bond in a configuration. The following data were collected using density functional theory calculations for Pt clusters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Geometry Type |  |  |  | (kJ/mol) |
| dimer | 2 | 0 | 0 | -255.3 |
| 3-atom triangle | 3 | 3 | 0 | -603.2 |
| 3-atom line | 3 | 1 | 0 | -475.0 |
| 5-atom line | 5 | 3 | 0 | -865.6 |
| 4-atom tetrahedron | 4 | 4 | 4 | -969.7 |
| 4-atm trigonal planar | 4 | 1 | 1 | -655.2 |
| 4-atom square | 4 | 4 | 0 | -858.7 |
| 4-atom diamond | 4 | 4 | 2 | -911.5 |
| 6-atom rectangle | 6 | 6 | 2 | -1481.9 |
| 6-atom pair of triangles | 6 | 6 | 2 | -1333.4 |

Use linear regression to find the best-fit values of , , and ; also report the corresponding R2 value.

***Solution***

%PS8\_4.m

clear;

%We'll solve this using general linear least squares:

%E = g1\*alpha1 + g2\*alpha2 + g3\*alpha3

%g1(n1,n2,n3) = n1; g2(n1,n2,n3) = n2; g3(n1,n2,n3) = n3

Z = [2 0 0;

3 3 0;

3 1 0;

5 3 0;

4 4 4;

4 1 1;

4 4 0;

4 4 2;

6 6 2;

6 6 2];

%y contains known energies

y = [-255.3 -603.2 -475 -865.6 -969.7 -655.2 -858.7 -911.5 -1481.9 -1333.4]';

a = (Z'\*Z)\Z'\*y;

alpha1 = a(1)

alpha2 = a(2)

alpha3 = a(3)

SR = sum((Z\*a-y).^2); ST = sum((y-mean(y)).^2);

r2 = 1-SR/ST

***Output***

>> PS8\_4

alpha1 =

-127.3095

alpha2 =

-89.0633

alpha3 =

-34.6003

r2 =

0.9838

1. L. T. Roling, T. Choksi, F. Abild-Pedersen, “A coordination-based model for transition metal alloy nanoparticles”, *Nanoscale* 11, 4438-4452 (2019), <https://dx.doi.org/10.1039/C9NR00959K> [↑](#footnote-ref-1)