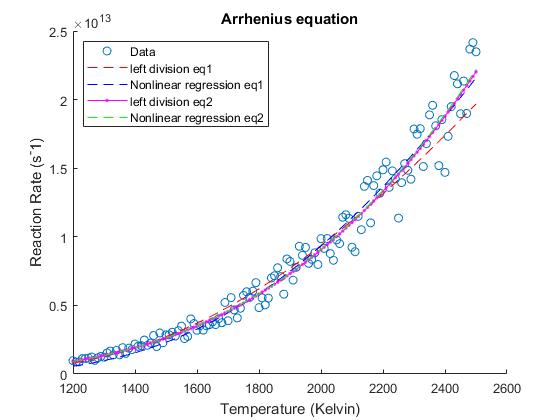
Project 2

Hung Le(John) & Anh Nguyen(Ryan)

**Result**



**Discussion:**

We were asked to import a .dat data file to a variable in our .mat file and use those numbers to perform curve fit to the given equations. Out of the three curve fitting methods that we went through, we first tried to use the General Linear Least Squares with Left Division because it does not require any initial guesses and it can perform with the least restrictions (can be applied on multivariable functions unlike polyfit. Predictably, this method did well on both equations and give reasonable results. However, it was not the most accurate method out of the three, so we tried Nonlinear Regression, which is believed to give a better result.

Nonlinear Regression is a method that requires initial guesses for each model. Therefore, we decided to use the results that we found when we used GLLS for both equations for the initial guesses for both fminsearch. Consequently, the curve fit turned out to be very accurate as predicted. We used the coefficient of determination (r^2) as our way to measure fit quality. As predicted, the r^2 number got closer to 1 when we use the Nonlinear Regression for both cases.

As a result, we decided to use both methods for this curve fit problem. GLLS with Left Division to look for initial guesses for fminsearch. The most challenging aspect of this project is the process of looking for initial guesses for fminsearch when we look for optimized accuracy. Because if initial guesses are poorly chosen, our results might not be totally converged after the maximum iteration. The interesting aspect of this project is that we could learn more about Arrhenius equation. We also learned to import raw data points into MATLAB and find the optimized fitting method. Before, we used programs like Logger Pro and Excels which does not fully provide good curve fits, now we use the actual mathematical way to fit the curve to this data, which is very interesting in our opinion.

Even though the results for equation 2 was not fully converge, the numbers are good enough because the initial guesses are very close to roots of Left Divisions. Furthermore, r^2 is really close to 1 so the results are very accurate and reliable.

**Import File Code :**

function OH2rxn1 = importfile(filename, startRow, endRow)

%IMPORTFILE Import numeric data from a text file as a matrix.

% OH2RXN1 = IMPORTFILE(FILENAME) Reads data from text file FILENAME for

% the default selection.

%

% OH2RXN1 = IMPORTFILE(FILENAME, STARTROW, ENDROW) Reads data from rows

% STARTROW through ENDROW of text file FILENAME.

%

% Example:

% OH2rxn1 = importfile('O\_H2\_rxn.dat', 1, 131);

%

% See also TEXTSCAN.

% Auto-generated by MATLAB on 2018/02/10 12:42:59

%% Initialize variables.

delimiter = ' ';

if nargin<=2

startRow = 1;

endRow = inf;

end

%% Format for each line of text:

% column1: double (%f)

% column2: double (%f)

% For more information, see the TEXTSCAN documentation.

formatSpec = '%f%f%[^\n\r]';

%% Open the text file.

fileID = fopen(filename,'r');

%% Read columns of data according to the format.

% This call is based on the structure of the file used to generate this

% code. If an error occurs for a different file, try regenerating the code

% from the Import Tool.

dataArray = textscan(fileID, formatSpec, endRow(1)-startRow(1)+1, 'Delimiter', delimiter, 'MultipleDelimsAsOne', true, 'TextType', 'string', 'HeaderLines', startRow(1)-1, 'ReturnOnError', false, 'EndOfLine', '\r\n');

for block=2:length(startRow)

frewind(fileID);

dataArrayBlock = textscan(fileID, formatSpec, endRow(block)-startRow(block)+1, 'Delimiter', delimiter, 'MultipleDelimsAsOne', true, 'TextType', 'string', 'HeaderLines', startRow(block)-1, 'ReturnOnError', false, 'EndOfLine', '\r\n');

for col=1:length(dataArray)

dataArray{col} = [dataArray{col};dataArrayBlock{col}];

end

end

%% Close the text file.

fclose(fileID);

%% Post processing for unimportable data.

% No unimportable data rules were applied during the import, so no post

% processing code is included. To generate code which works for

% unimportable data, select unimportable cells in a file and regenerate the

% script.

%% Create output variable

OH2rxn1 = table(dataArray{1:end-1}, 'VariableNames', {'e03','e11'});

**Project Code :**

clc

close

clear

%% Data and equation

%Data

Data\_proj2= importfile('O\_H2\_rxn.dat');

Data\_proj2=table2array(Data\_proj2);

Ta=Data\_proj2(:,1);

k =Data\_proj2(:,2);

R=8.314; % J/(mol K)

%equation

k\_model1=@(T,A,E) A\*exp(-E./(R.\*T));

k\_model2=@(T,A,b,E) A\*T.^b.\*exp(-E./(R.\*T));

%% First Equation

%% GLLS with left division

y\_lin=log(k);

z0=ones(length(k),1);

z1=log(Ta);%This is for second equation

z2=1./(Ta);

Z1=[z0,z2];

c1=Z1\y\_lin;

A\_GLLS1=exp(c1(1));

E\_GLLS1=-R\*c1(2);

% fit quality for first equation

Sr1 = sum((k-k\_model1(Ta,A\_GLLS1,E\_GLLS1)).^2);

St1 = sum((k-mean(k)).^2);

r\_squared\_GLLS1 = 1-Sr1/St1;

Syx\_GLLS1=sqrt(Sr1/(length(Ta)-2));

%% Nonlinear regression

Sr\_fun1 = @(d1,xi,yi) sum((yi-k\_model1(xi,d1(1),d1(2))).^2);

d1 = fminsearch(Sr\_fun1,[A\_GLLS1,E\_GLLS1],[],Ta,k);

A\_NR1 = d1(1);

E\_NR1 = d1(2);

%fit quality

Sr1 = sum((k-k\_model1(Ta,A\_NR1,E\_NR1)).^2);

r\_squared\_NR1 = 1-Sr1/St1;

Syx\_NR1=sqrt(Sr1/(length(Ta)-2));

%% Second equation

%% GLLS with left division

Z2=[z0,z1,z2];

c2=Z2\y\_lin;

A\_GLLS2=exp(c2(1));

b\_GLLS2=c2(2);

E\_GLLS2=-R\*c2(3);

% fit quality for second equation

Sr2 = sum((k-k\_model2(Ta,A\_GLLS2,b\_GLLS2,E\_GLLS2)).^2);

St2 = sum((k-mean(k)).^2);

r\_squared\_GLLS2 = 1-Sr2/St2;

Syx\_GLLS2=sqrt(Sr2/(length(Ta)-3));

%% Nonlinear regression

Sr\_fun2 = @(d2,xi,yi) sum((yi-k\_model2(xi,d2(1),d2(2),d2(3))).^2);

d2 = fminsearch(Sr\_fun2,[A\_GLLS2,b\_GLLS2,E\_GLLS2],[],Ta,k);

A\_NR2=d2(1);

b\_NR2=d2(2);

E\_NR2= d2(3);

%fit quality

Sr2 = sum((k-k\_model2(Ta,A\_NR2,b\_NR2,E\_NR2)).^2);

r\_squared\_NR2 = 1-Sr2/St2;

Syx\_NR2=sqrt(Sr2/(length(Ta)-3));

%% plot data

figure()

hold on

plot(Ta,k,'o');

plot(Ta,k\_model1(Ta,A\_GLLS1,E\_GLLS1),'r--');

plot(Ta,k\_model1(Ta,A\_NR1,E\_NR1),'b--');

plot(Ta,k\_model2(Ta,A\_GLLS2,b\_GLLS2,E\_GLLS2),'m.-');

plot(Ta,k\_model2(Ta,A\_NR2,b\_NR2,E\_NR2),'g--');

title('Arrhenius equation');

xlabel('Temperature (Kelvin)');

ylabel('Reaction Rate (s^-1)');

legend('Data','left division eq1','Nonlinear regression eq1','left division eq2','Nonlinear regression eq2');