Shashi Shivaraju
XID:C88650674

REPORT
ECE 8540 Lab #2 – Nonlinear Regression

Objective: To fit a given nonlinear model y = ln(a*x) to given set of data.

Implementation:

Let the model be represented by the equation y = ln(a*x).This model is nonlinear in nature and the unknown in the model is 'a'.

The root finding method is used to determine the value of the 'a'.The error function of the given model is determined and by finding its minimum (i.e where the derivative of error function is zero),the value of 'a' is found. Starting with an initial guess, an iterative technique is used to move towards the minima.

The given nonlinear model is represented by:

$$y = ln(ax)$$

The error function of the model is given by:

$$E = \sum_{i=1}^{N} (y_i - ln(ax_i))^2$$

We take the partial derivative of the error functiom with respect to the unknown 'a' :

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{N} 2(y_i - ln(ax_i)) * (\frac{-1}{a})$$

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{N} \frac{-2(y_i - ln(ax_i))}{a}$$

Shashi Shivaraju
XID:C88650674

We wish to minimize this error, which is where this partial derivative is equal to zero:

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{N} \frac{-2(y_i - ln(ax_i))}{a} = 0$$

Since -2 is constant with respect to 'a',it can be simplified as:

$$\frac{\partial E}{\partial a} = \sum_{i=1}^{N} \frac{y_i - ln(ax_i)}{a} = 0$$

To solve this problem iteratively using the root finding approach, let

$$f(a) = \sum_{i=1}^{N} \frac{y_i - ln(ax_i)}{a} = 0$$

The derivative of f(a) is given by:

$$f'(a) = \sum_{i=1}^{N} (\frac{a(\frac{-1}{a}) - (y_i - ln(ax_i))}{a^2})$$

$$f'(a) = \sum_{i=1}^{N} \frac{-1 - y_i + ln(ax_i)}{a^2}$$

Shashi Shivaraju
XID:C88650674

We take an initial guess of 'a' value and through successive iterations of the below equation we converge to the solution:

$$a_{n+1} = a_n - \frac{f(a_n)}{f'(a_n)}$$

Using the above mentioned procedure,the value of 'a' is found to fit the model y=ln(ax) to the provided data points in files log-data-A, log-data-B and log-data-C.

[code snippet]:

```
an = initial_guess[count]; /*initial guess near to true value*/
for(i = 0;i < MAX_ITERATIONS;i++) /*iterate until zero crossing is determined*/
{
        fan = 0;
        fpan1 = 0;

        for(j = 0;j < DATA_LEN;j++)
        {
          fan = fan + (y_data[j] – log(an*x_data[j]))/an; /*derivative of error function*/
          fpan1 = fpan1 +(log(an*x_data[j])-y_data[j]-1)/(an*an); /*derivative of f(a)*/
        }

        an1 = an - (fan/fpan1);

        printf("iteration = %d an = %lf an1=%lf\n",i,an,an1);
        if(fabs(an1 -an) < 0.0000001) /*check for error threshold*/
        {
                break; /*value found*/
        }

        an = an1;
}
```

Shashi Shivaraju
XID:C88650674

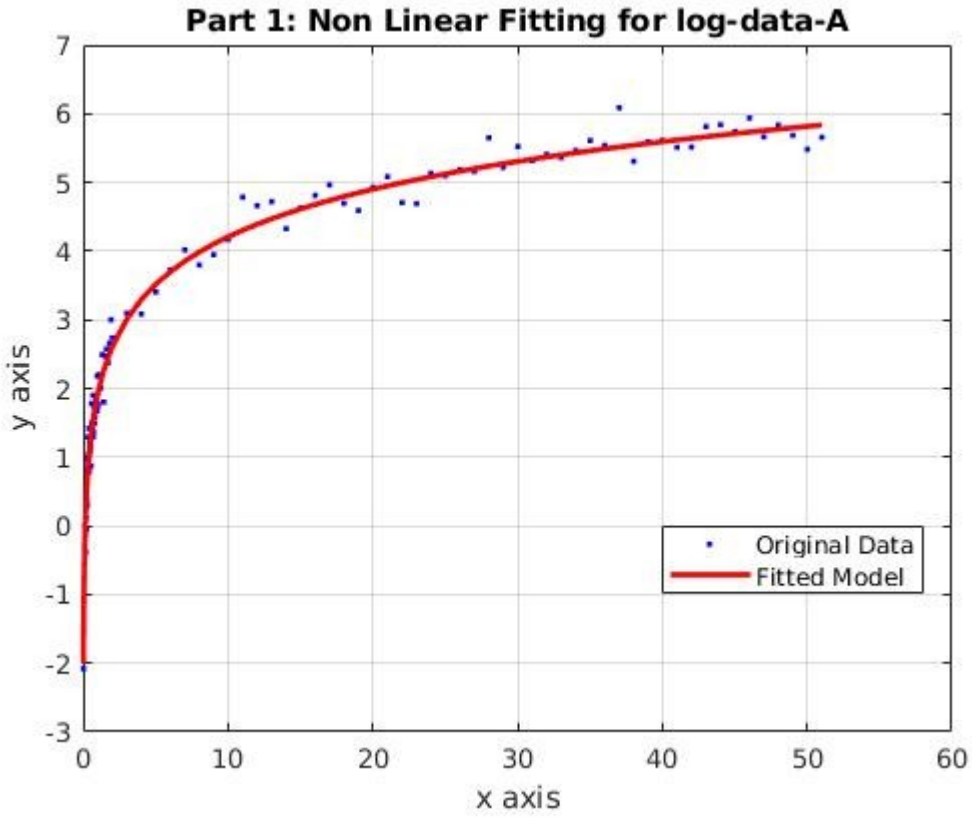
[Results]:

1) Data File : log-data-A.txt

Initial value of 'a' = 6.00
Final value of 'a' = 6.711359
Total iterations = 5

iteration	a_n	a_{n+1}
0	6.00	6.604520
1	6.604520	6.708830
2	6.708830	6.711357
3	6.711357	6.711359
4	6.711359	6.711359

[Plot]:



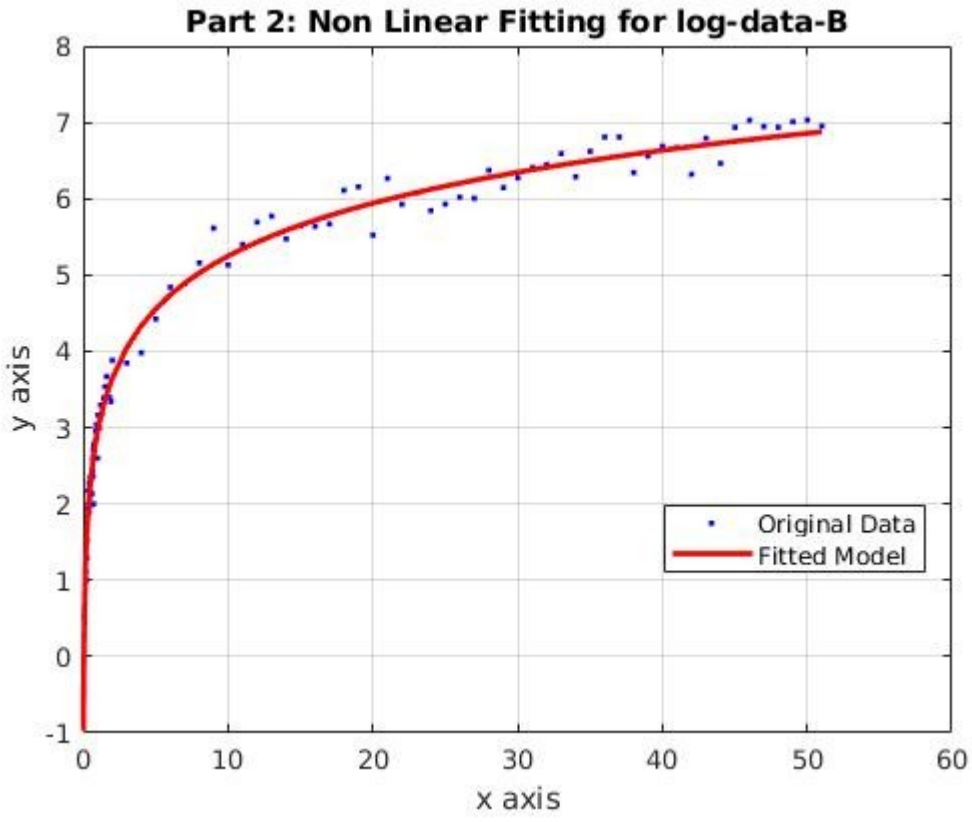
Shashi Shivaraju
XID:C88650674

2) Data File : log-data-B.txt

Initial value of 'a' = 15.00
Final value of 'a' = 18.996116
Total iterations = 5

iteration	a_n	a_{n+1}
0	15.00	17.865887
1	17.865887	18.898466
2	18.898466	18.995365
3	18.995365	18.996116
4	18.996116	18.996116

[Plot]:



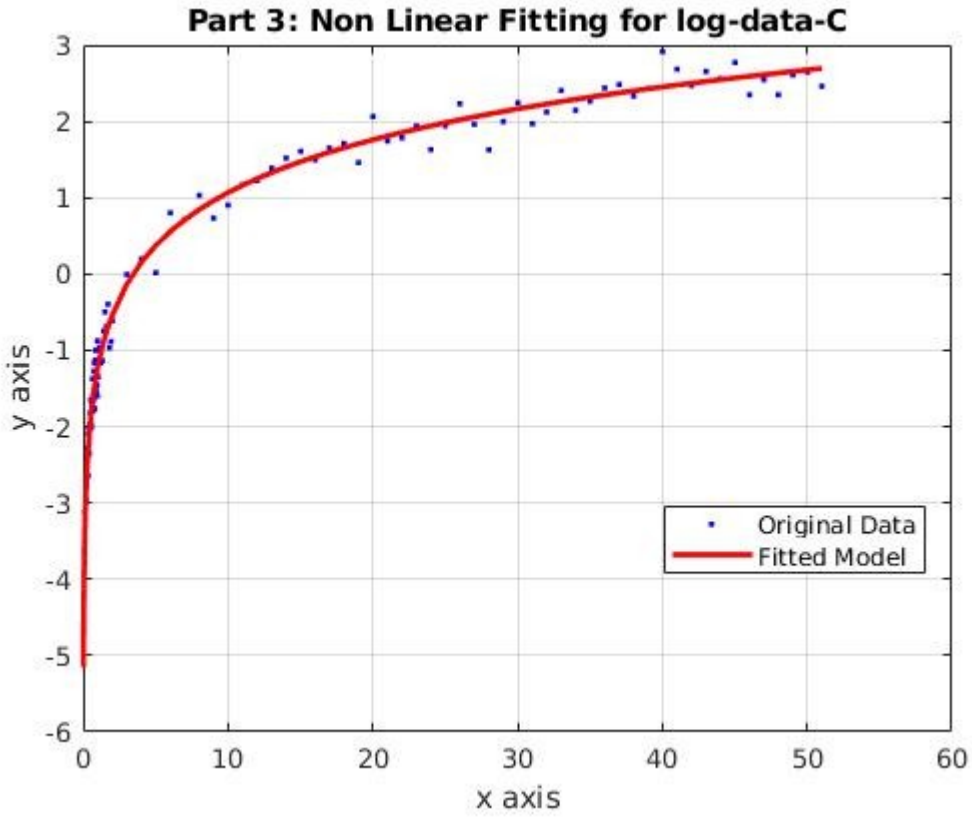
Shashi Shivaraju
XID:C88650674

3) Data File : log-data-C.txt

Initial value of 'a' = 0.2
Final value of 'a' = 0.289998
Total iterations = 6

iteration	a_n	a_{n+1}
0	0.2	0.254180
1	0.254180	0.283786
2	0.283786	0.289801
3	0.289801	0.289998
4	0.289998	0.289998
5	0.289998	0.288998

[Plot]:



```c
/*****************************************************************************
*  FILE NAME    : non_linearfitting.c
*
*  DESCRIPTION  : Program to calculate a nonlinear regression fit by
*                 implementing a root finding method.
*
*  PLATFORM     : Linux
*
*  DATE                 NAME                    REASON
*  8th Sep,2018         Shashi Shivaraju        ECE_8540_lab_02
*                       [C88650674]
*****************************************************************************/
/*Header file inclusions*/
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/*MACRO decalrations*/
#define DATA_LEN    110 /*total number of (xi,yi) points in data file*/
#define DATA_FILE_NUM   3 /*Total number of data file*/

#define MAX_ITERATIONS  500000

/*Main function of the program*/
int main()
{
    FILE *fp = NULL;        /*File pointer to open and read data from file*/
    int ret = 0;            /*Variable to check return value*/
    int count = 0,i = 0,j = 0;          /*variable for loop*/
    double an = 0,an1 = 0,fan = 0,fpan1 = 0; /*variable for calculations*/
    double x_data[DATA_LEN] = {0};  /*Array to store the x coordinates*/
    double y_data[DATA_LEN] = {0};   /*Array to store the y coordinates*/
    double initial_guess[DATA_FILE_NUM] = {6,15,0.2}; /*Initial guess value of 'a'*/
    /*file name of the data file*/
    char* filenames[DATA_FILE_NUM] = {"log-data-A.txt","log-data-B.txt","log-data-C.txt"};

    /*Calculate the unknown 'a' in the model y = ln(ax) for the three data files*/
    for(count = 0;count < DATA_FILE_NUM;count++)
    {
        fp = fopen(filenames[count],"r"); /*open the data for reading*/
        if(!fp)
        {
            printf("fopen failed for %s",filenames[count]);
            break;
        }

        printf("Current file : %s\n",filenames[count]);

        /*read the data from the file*/
        for(i = 0;i < DATA_LEN;i++)
        {
            fscanf(fp,"%lf %lf",&x_data[i],&y_data[i]);
            //printf("%lf %lf\n",x_data[i],y_data[i]);
        }

        an = initial_guess[count];/*initial guess near to true value*/
        for(i = 0;i < MAX_ITERATIONS;i++) /*iterate until zero crossing is determined*/
```

```
        {
            fan = 0;
            fpan1 = 0;

            for(j = 0;j < DATA_LEN;j++)
            {
               fan = fan + (y_data[j] - log(an*x_data[j]))/an;
               fpan1 = fpan1 +(log(an*x_data[j])-y_data[j]-1)/(an*an);
            }

            an1 = an - (fan/fpan1);

            printf("iteration = %d an = %lf an1=%lf\n",i,an,an1);
            if(fabs(an1 -an) < 0.0000001)
            {
                break; /*value found*/
            }

            an = an1;
        }

        /*close the file*/
        if(fp)
        {
            fclose(fp);
        }
        fp = NULL;
    }

    return 0;
}
```

[Matlab Code for Plots]:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FILE NAME     : non_linearfit.m
%
% DESCRIPTION   : To plot the graphs to fit a
%                 function of the form y = ln(ax) to given set of data.
%
% PLATFORM      : Matlab
%
% DATE          NAME                REFERENCE     REASON
% 11-Sep-2018   Shashi Shivaraju      Initial code  ECE 8540 lab2
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear; %clear all the varaibles
clc; %clear the screeen

%Read the data from file and store it in matrices
A = dlmread('log-data-A.txt');
B = dlmread('log-data-B.txt');
C = dlmread('log-data-C.txt');

%plot the data points and the fitted line for log-data-A.txt
YA = A(:,2);
XA = A(:,1);
a1 = 6.711359; %value of a found using root finding method
figure(1)
plot(XA, YA,'b.'); %plot original data points
grid on;
hold on;
y1 = log(a1.*XA);  %plot the fitted line
plot(XA,y1,'r','LineWidth',2);
legend('Original Data','Fitted Model')
hold off;
title('Part 1: Non Linear Fitting for log-data-A');
xlabel('x axis');
ylabel('y axis');


%plot the data points and the fitted line for log-data-B.txt
YB = B(:,2);
XB = B(:,1);
a2 = 18.996116; %value of a found using root finding method
figure(2)
plot(XB, YB,'b.'); %plot original data points
grid on;
hold on;
y2 = log(a2.*XB);  %plot the fitted line
plot(XB,y2,'r','LineWidth',2);
legend('Original Data','Fitted Model')
hold off;
title('Part 2: Non Linear Fitting for log-data-B');
xlabel('x axis');
ylabel('y axis');
```

```
%plot the data points and the fitted line for log-data-C.txt
YC = C(:,2);
XC = C(:,1);
a3 = 0.289998; %value of a found using root finding method
figure(3)
plot(XC, YC,'b.'); %plot original data points
grid on;
hold on;
y3 = log(a3.*XC);  %plot the fitted line
plot(XC,y3,'r','LineWidth',2);
legend('Original Data','Fitted Model')
hold off;
title('Part 3: Non Linear Fitting for log-data-C');
xlabel('x axis');
ylabel('y axis');
```