

ECE 8540 Analysis of Tracking System

Lab 2 report
Non-Linear Fitting

Harshal B. Varpe
September 12, 2022

1 Introduction

Lab 2 focuses on calculating a nonlinear regression fit without using any of the high-level functions that matlab or any other software package provides. The students are provided with 3 different datasets: A, B, and C. Each file has two columns x and y. We are to fit a function of the form - Let us start with the derivation process of the non-linear regression. For non-linear model fitting, we used the idea of root finding. We write an error function and find its minimum. The minimum could be found where the derivative of this function is zero. We are given that all the data sets follow the form of , where is a non-linear unknown. We can write the error function as follows -

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

The partial derivative of the error function with respect to the unknown a :

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

We can simplify the equation 2 to the following form, since is a constant with respect to a .

$$\frac{\partial E}{\partial a} = \sum_{i=1}^N (y_i - \ln(ax_i)) \left(\frac{1}{a}\right) \quad (3)$$

We wish to minimize the error, therefore we equate equation 3 to 0.

$$\frac{\partial E}{\partial a} = \sum_{i=1}^N (y_i - \ln(ax_i)) \left(\frac{1}{a}\right) = 0 \quad (4)$$

We wish to use the root finding approach, which is an iterative approach. In order to implement this iterative approach, we have a following equation:

$$a_{n+1} = a_n - \frac{f(a_n)}{f'(a_n)} \quad (5)$$

Let us assume -

$$f(a) = \sum_{i=1}^N (y_i - \ln(ax_i)) \left(\frac{1}{a}\right) \quad (6)$$

If we take a partial derivative of equation 6, we get :

$$f'(a) = \sum_{i=1}^N \left(\frac{-1}{a^2}\right) (y_i - \ln(ax_i) + 1) \quad (7)$$

Now we substitute the equation 6 and the equation 7 in equation 5 and solve for a . We make an initial guess for a and then iterate until the desired convergence is observed.

2 Code and Result

The following image shows how the equations shown in section 1 are implemented in MATLAB. The inner loop with q as counter calculates the values for equations 6 and 7. Based on these values, we calculate equation 5. Thus, we get a new guess. The outer loop with counter p initiates new calculations for the unknown with an updated guess. To avoid the infinite loop, in the case of divergence, we use a condition with variable p_max . If the difference between (a_n) and (a_{n+1}) is greater than p_max , then we break out of the for loop. The value of p_max is arbitrary. It is selected to suit the needs of the script. If the difference between (a_n) and (a_{n+1}) falls below 10^{-5} , we stop the calculations as sufficient degree of accuracy is achieved. At the end we print out how many iteration it took to converge to the solution and the final value of the unknown.

```
% Actual calculation of the Loop
x = data.Var1;
y = data.Var2;
p_max = 30000;
for p=1:1:p_max
    f_an = 0;
    f_pan = 0;
    for q=1:1:length(x)
        f_an = f_an + ((y(q)-log(a0*x(q)))*(1/a0));
        f_pan = f_pan + ((-1/(a0^2)))*(y(q) - log(a0*x(q))+ 1);
    end
    a1 = a0 -(f_an/f_pan);
    fprintf("a0 is %0.5f  a1 is %0.5f \n",a0,a1)
    diff = abs(a0-a1);
    if diff < 0.000001
        fprintf("Convergence reached!!")
        break;
    elseif diff > p_max
        fprintf("Wrong guess! Possible Divergence!")
        break;
    end
    a0 = a1;
end

fprintf("No. of iterations is : %d. ",p)
fprintf("The final value of unknown a is %0.5f ",a0)
```

Figure 1: A snippet of the MATLAB script

We had three different datasets. For each of the dataset, five different initial guesses were considered. The purpose of selecting different initial guesses was to study the affect on convergence. Below are the three tables for each of the dataset containing initial guess, number of iteration for convergence ,and a convergence value.

In the table 1, we can see the initial values chosen for the five iterations. As expected, as our initial guess gets closer to the answer or the approximate value of the unknown a , number of iteration required reduce. Although not shown here, if the initial guess differs from the final values of a significantly, then convergence may not be reached.

Initial Guess (a)	Number of Iterations	Final value of a
1	9	6.71136
2	8	6.71136
5	5	6.71136
7	4	6.71136
10	8	6.71136

Table 1: Initial guesses for data set A

In the table 2, we can see just like the Data set A, initial guess was 1. Since the initial guess was significantly lower than the final answer, it took 11 iterations to reach the final answer.

Initial Guess (a)	Number of Iterations	Final value of a
1	11	18.99612
9	7	18.99612
15	5	18.99612
20	4	18.99612
30	9	18.99612

Table 2: Initial guesses for data set B

In the table 3, we observe similar pattern. The closer our initial guess is to the final answer, the faster we converge to the solution. Initial guess of 1 did not work for this data set. As compared to the data set A and data set B, the initial guess had to be lowered to 0.2. It was observed that for the Initial guess value of 0.5, the algorithm failed to converge.

Initial Guess (a)	Number of Iterations	Final value of a
0.2	5	0.29000
0.1	7	0.29000
0.4	6	0.29000
0.45	8	0.29000
0.35	5	0.29000

Table 3: Initial guesses for data set C

Below are the plots for each of the data sets.

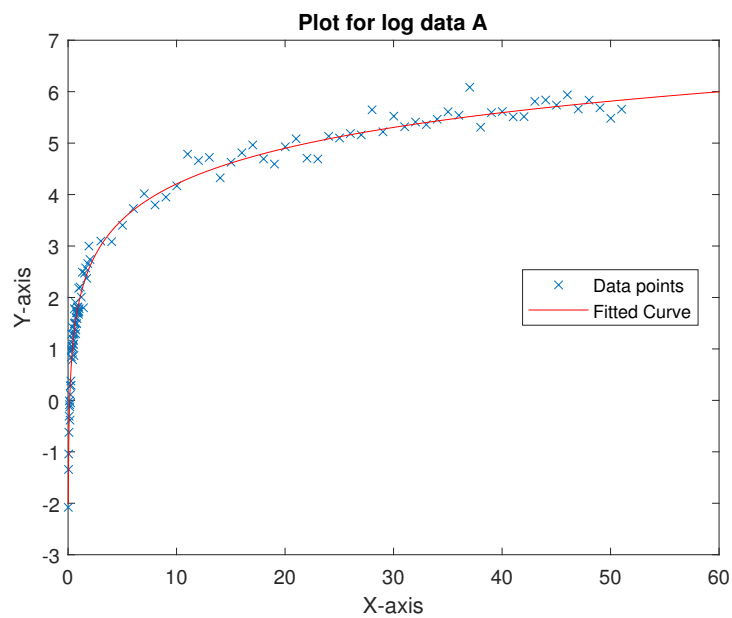


Figure 2: Data points and fitted curve for Dataset A

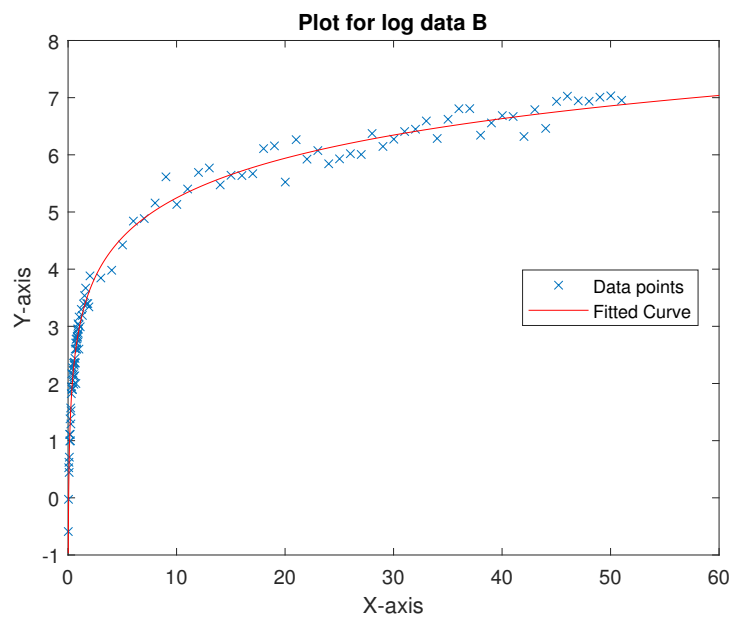


Figure 3: Data points and fitted curve for Dataset B

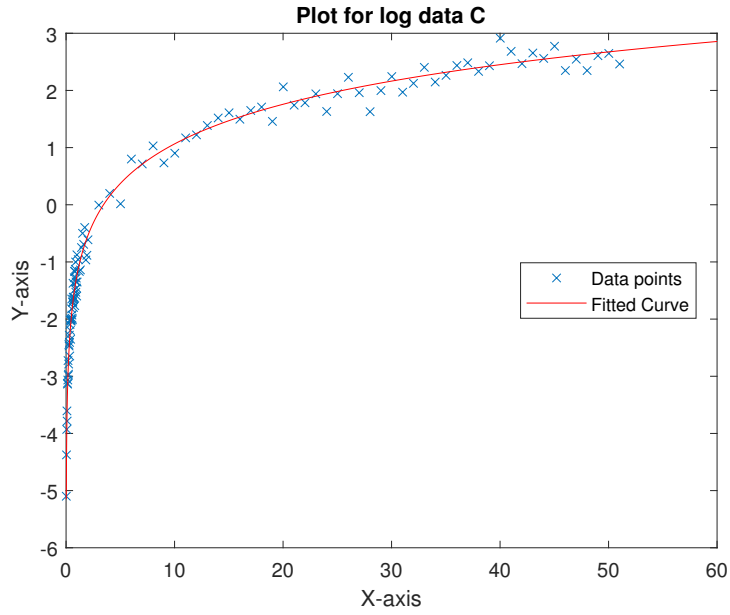


Figure 4: Data points and fitted curve for Dataset C

In summary, we can say that root finding techniques can be used for the purpose of non-linear curve fitting. One of the demerits of this technique is that the results are limited to initial guess. If the initial guess is far off from the solution, the convergence may not be reached leading to failure. However, within the certain limits for an initial guess, the technique is quite robust and consistent.