ECE 8540

LAB 2 - NONLINEAR REGRESSION

September 7, 2021

Tim Driscoll
Clemson University
Department of Electrical and Computer Engineering
tdrisco@clemson.edu

1 Introduction

This lab report concerns the problem of nonlinear model fitting. This will allow for more accurate model predictions for complex data sets that can not be represented by linear model fitting. In order to preform this model fitting the root finding method was implemented in MATLAB. The program was designed to run and fit a model to three different data sets including, log-data-A, log-data-B, and log-data-C. To solve for the model's unknown the error function was minimized by determining where its derivative equaled zero. The error function and its derivative can be observed by the following:

$$E = \sum (\text{data} - \text{model}) \tag{1}$$

Derivative:

$$\frac{\partial E}{\partial \ \text{unknown(s)}} = 0 \tag{2}$$

By starting with an initial guess the code developed in MATLAB implemented an iterative method to determine where the unknown converged and the derivative approached zero.

2 Methods

2.1 Mathematical Algorithm

Given that all three data sets can be represented in the form of $y = ln(ax)$ where a is a nonlinear unknown the error function can be modeled by:

$$E = \sum_{i=1}^{N} (y_i - ln(ax_i))^2 \tag{3}$$

The partial derivative is then taken with respect to a:

$$\begin{aligned} \frac{\partial E}{\partial a} &= \sum_{i=1}^{N} 2(y_i - ln(ax_i))(\frac{-1}{ax_i})(x_i) && (4) \\ &= \sum_{i=1}^{N} 2(y_i - ln(ax_i))(\frac{-1}{a}) && (5) \end{aligned}$$

In order to minimize the error function we must set equation 5 equal to zero:

$$\sum_{i=1}^{N} 2(y_i - ln(ax_i))(\frac{-1}{a}) = 0 \tag{6}$$

Equation 6 can be simplified to the following form.

$$\sum_{i=1}^{N} (y_i - ln(ax_i))(\frac{1}{a}) = 0 \tag{7}$$

Next, to solve the problem using the iterative root method we need to define the following equation:

$$f(a) = \sum_{i=1}^{N}(y_i - ln(ax_i))(\frac{1}{a}) \tag{8}$$

$$= \sum_{i=1}^{N}(\frac{y_i}{a} - \frac{ln(ax_i)}{a}) \tag{9}$$

The derivative of the function $f(a)$ is

$$f'(a) = \sum_{i=1}^{N}(\frac{-y_i}{a^2} - \frac{1 - ln(ax_i)}{a^2}) \tag{10}$$

Finally, an initial guess is made for a. Then through iteration a is updated given equation 11 until the desired convergence is met.

$$a_{n+1} = a_n - \frac{f(a_n)}{f'(a_n)} \tag{11}$$

2.2 Code Algorithm

The mathematical algorithm that was explained in section 2.1 above was implemented in code using the 2020a version of MATLAB on a Windows operating system. The program functioned by first reading in all three of the provided data sets, A, B, and C respectively. The user is then prompted to select which data set they would like to model and to enter an initial guess for the unknown. The selected data set is then plotted and the nonlinear regression is performed. This is performed by first calculating both equations 9 and 10. Then given the current value of a (a_n) a new estimate of a (a_{n+1}) is calculated using equation 11. The difference between (a_n) and (a_{n+1}) is checked to determine if the values are within 0.0000001 of each other. If the two values are within this threshold they meet the condition set for convergence and the final value of a will be set to (a_{n+1}). If this condition is otherwise not met the process will repeat again. The algorithm allows for a maximum of 50000 iterations. After the value of a is determined the final model is graphed on the original plot represented by $y = ln(ax)$.

3 Results

By coding the a program in MATLAB the iterative process of the root finding method could be quickly performed to produce accurate values of a for the model $y = ln(ax)$. As discussed in section 2 above an initial guess for a was needed to start the calculations. For each model a number of different initial values for a were used to observe the effects on the convergence. The three tables below represent initial guesses that were used, the number of iterations needed for convergence, and the final value the algorithm converged to.

Table 1 shows the outcome of four different initial guesses used for data set A. As you move down the column you can observe a decrease in the number of iterations. All 4 of these

Table 1: Convergence of a for data set A.

Initial Guess (a)	Number of Iterations	Final Value (a)
1	9	6.711359
10	8	6.711359
6	5	6.711359
6.5	4	6.711359

Table 2: Convergence of a for data set B.

Initial Guess (a)	Number of Iterations	Final Value (a)
1	11	18.996116
30	9	18.996116
18	4	18.996116
18.9	3	18.996116

initial guesses produced the same final value for a. Then looking at figure 1 the raw data from set A can be observed by the red circles. The model can be viewed by the continuous black line which represents the equation $y = ln(6.711359x)$. The domain for this model spanned from the minimum to maximum x value found in data set A.

Table 2 shows the outcome of four different initial guesses used for data set B. As you move down the column you can observe a decrease in the number of iterations. This data set allowed for a wider range of initial guesses to be used then set A and C. All 4 of these initial guesses produced the same final value for a. Then looking at figure 2 the raw data from set B can be observed by the red circles. The model can be viewed by the continuous black line which represents the equation $y = ln(18.996116x)$. The domain for this model spanned from the minimum to maximum x value found in data set B.

Table 3 shows the outcome of four different initial guesses used for data set C. As you move down the column you can observe a slight decrease in the number of iterations. This data set was restricted to the tightest set of initial guesses. All 4 of these initial guesses produced the same final value for a. Then looking at figure 3 the raw data from set C can be observed by the red circles. The model can be viewed by the continuous black line which represents the equation $y = ln(0.289998x)$. The domain for this model spanned from the minimum to maximum x value found in data set C.

Table 3: Convergence of a for data set C.

Initial Guess (a)	Number of Iterations	Final Value (a)
0.1	7	0.289998
0.4	6	0.289998
0.2	6	0.289998
0.25	5	0.289998

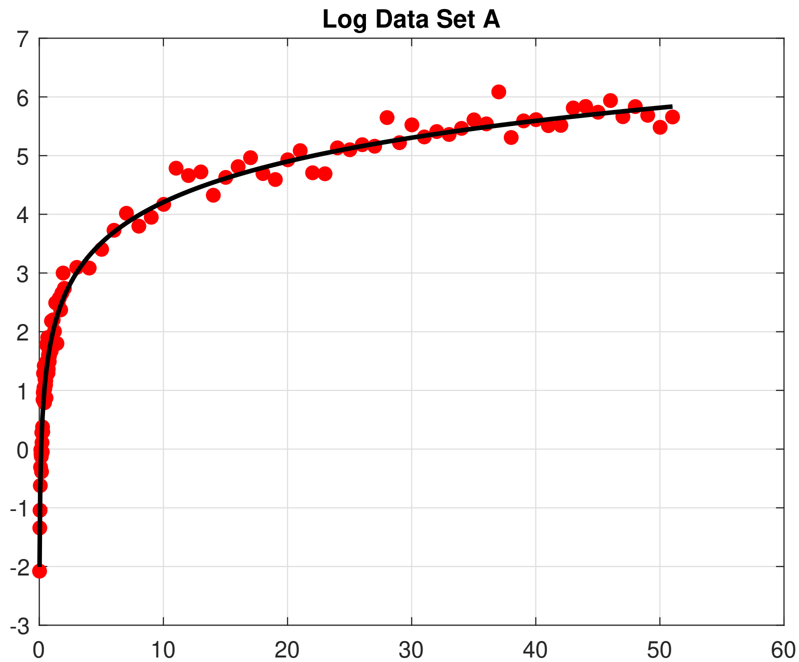

Figure 1: Fitting a logarithmic function to data set A.

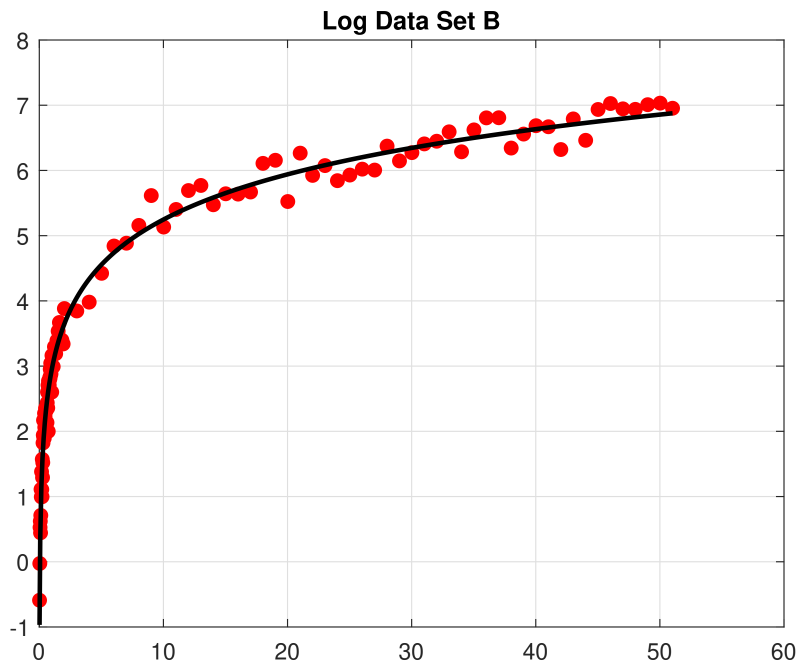

Figure 2: Fitting a logarithmic function to data set B.

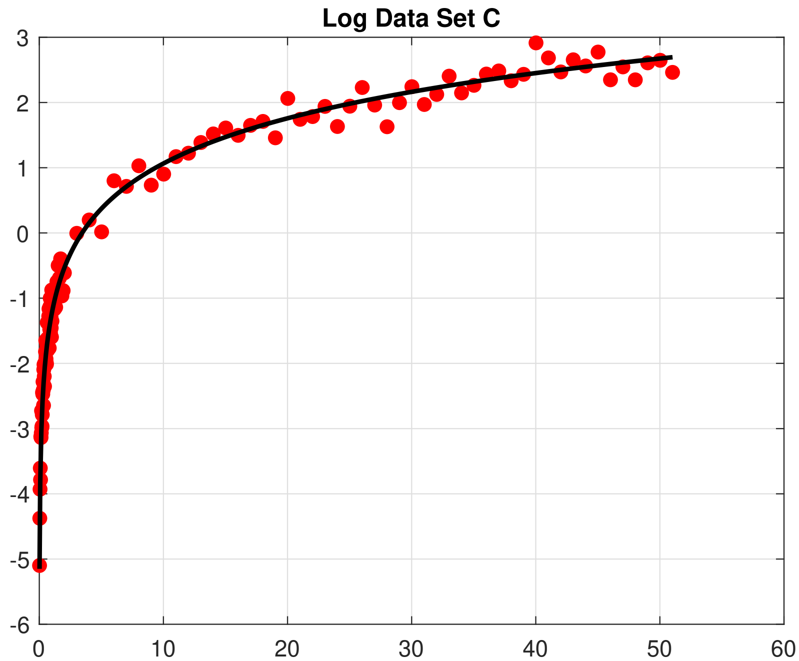

Figure 3: Fitting a logarithmic function to data set C.

4 Conclusion

The idea of using of root finding to determine the unknowns for nonlinear model fitting is a very powerful tool. It allows models to be accurately fitted to complex data sets that can't utilize linear models. For all three of the different data sets, A, B, and C the methods described in section 2 were able to accurately fit a nonlinear model. The residual for each point was reduced to a minimum given the noise within each data set. Although the models accurately fit all three data sets the method was limited by the initial guess. In many cases if the initial guess was not close enough to the final value the method would fail. This limitation added an element of experimentation to the algorithm to determine which starting values would produce correct results. This limitation proved particularly challenging in data set C where the rage of working initial guesses was very tight as shown in table 3. In the future it would be beneficial to take some time to get a deeper understanding of the raw data so a more accurate initial guess can be made.