TO LOG OR NOT TO LOG

PETER W. WHITE
TARLETON STATE UNIVERSITY
DEPARTMENT OF MATHEMATICS
BOX T-0470, STEPHENVILLE, TX 76402
WHITE@TARLETON.EDU

1. The Problem

I gave the following question on a take home exam after covering "Least-Squares Regression":

(1) For the data set {(0.0, 0.0), (0.5, 0.105), (1.0, 0.215), (1.5, 0.44), (2.0, 0.91), (2.5, 1.85), (3.0, 3.85), (3.5, 8.0), (4.0, 16.5), (4.5, 34.0)} find the least-squares regression fit for the following models.
   (a) $f(x) = \alpha x^2 + \beta x$.
   (b) $g(x) = \alpha x^{\beta}$.
   (c) $h(x) = \alpha e^{\beta x}$.

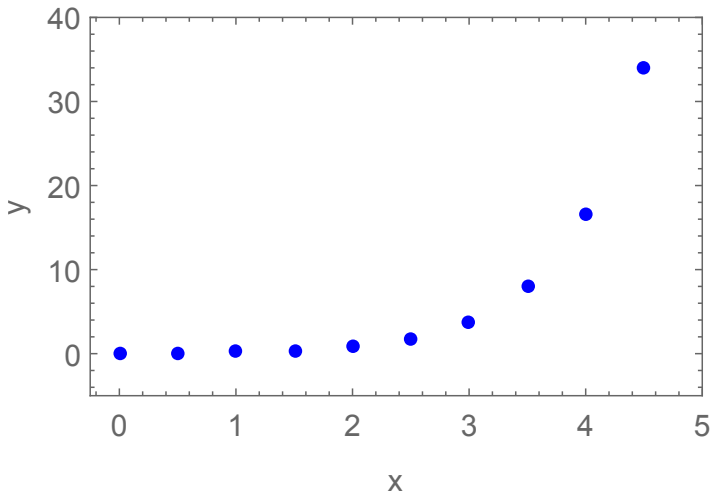

Figure 1. Exam problem data set.

In the class we had discussed transforming data using Log-Log and Semi-Log transformations to turn the second and third models into linear models. However, notice the first data point makes using logarithms problematic (several of the students sure noticed). So, one question is: "What do we do with that first data point when using logarithms to linearize the problem?"

More importantly, solving the least-squares problem on transformed data does not truly answer the question that was posed. Parameter values that minimize the square error for the linearized data, do not transform into parameter values that minimize the square error in the original problem. So, the more important question is "Do we use logarithms or not?"

Before we take a look at some of the issues that this exam question brought up, it may be helpful to discuss the background of these methods. The methods of (linear) least-squares regression are covered in all levels of undergraduate mathematics including College Algebra [1, 5], pre-calculus [3], matrix and linear algebra [6], and of course, numerical analysis [2]. Non-linear least-squares regression, however, is a more advanced topic that often requires numerical techniques to approximate solutions to systems of non-linear equations.

The basic idea behind least-squares regression is to find parameter values that minimize a square error function given by

$$E(\alpha, \beta) = \sum_{j=1}^{n} (y_j - \phi(x_j))^2, \tag{1}$$

where we are using the model functions from the exam question ($\phi \in \{f, g, h\}$) that each have the two parameters $\{\alpha, \beta\}$, and the data is given by $S = \{(x_j, y_j)\}$ for $j = 1, 2, 3, ..., n$ (with $n = 10$ for the exam question). To minimize this error function, we can apply the techniques in [7] by solving the system of equations resulting from

$$\frac{\partial E}{\partial \alpha} = 0, \ \frac{\partial E}{\partial \beta} = 0. \tag{2}$$

For the case when $\phi = f$ (the quadratic model) or any polynomial model, (2) leads to a linear system of equations that can be solved exactly. However, when $\phi = g$ (the power function model) or when $\phi = h$ (the exponential model), the parameter β appears in a non-linear manner in (2). In these last two cases, numerical methods are often needed to approximate solutions to this system of equations.

2. The Common Mistake

In the power and exponential models there is a simple method to convert the problem into a linear model. Namely, use logarithms. Note the following relationships:

$$y = \alpha x^{\beta} \iff \ln(y) = \ln(\alpha) + \beta \ln(x) \tag{3}$$

and

$$y = \alpha e^{\beta x} \iff \ln(y) = \ln(\alpha) + \beta x. \tag{4}$$

The common mistake is to think that if the transformed data "looks" linear, then (3) or (4) gives an easy way to fit the original non-linear models. Fitting a straight line to data is a relatively painless exercise [4].

That is, if $S_p = \{(\ln(x_j), \ln(y_j))\}$ "looks" linear, then we could find the best fit straight line $Y = A + BX$ to the data S_p and then the best fit power function to the

data S would be when $\alpha = e^A$ and $\beta = B$ from (3). If $S_e = \{(x_j, \ln(y_j))\}$ "looks" linear, then we could find the best fit line $Y = A + BX$ to the data S_e and then the best fit exponential function to the data S would be when $\alpha = e^A$ and $\beta = B$ from (4). End of problem, turn it in.

This can be a big mistake! Sadly, a mistake that the majority of my class made even after being warned to the contrary. So why does the above argument fail and what should we do?

3. The Solution

A graph of the data appears in Figure 1. A visual inspection of the data might suggest that the data could be modeled by any of the three models posed in the exam problem.

First, look at the best fit quadratic model. In the background we are imposing the condition that the model goes through the origin. We do this to reduce the number of parameters in the model to two (the same as the other models). This is justified since the data includes the origin.

Using a computer algebra system (*Mathematica* [9]), we find that the optimal parameter values in this case give

$$f(x) = 2.65626x^2 - 5.70446x.$$

Figure 2 shows that this is not a terrible fit but not really good either. The square error for this model, $E = 102.428$, is fairly large.

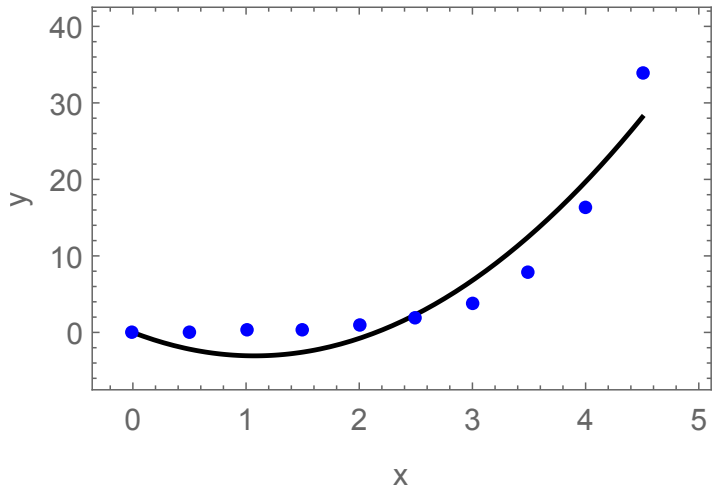

Figure 2. Best fit quadratic model.

For the power function model we first look at the linearized problem. To do this we need to note that the first data point (the origin) causes problems because $\ln(0)$ is undefined. For this model we can ignore the first data point because the power

function model automatically passes through the origin. Thus, no additional error is introduced in dropping the origin from the data set.

In Figure 3 the data points S_p are graphed along with the best fit straight line, $Y = 2.62683X - 1.28611$. This leads, through (3), to a power function model of

$$g(x) = 0.276344x^{2.62683} \tag{5}$$

and a square error of $E = 424.794$.

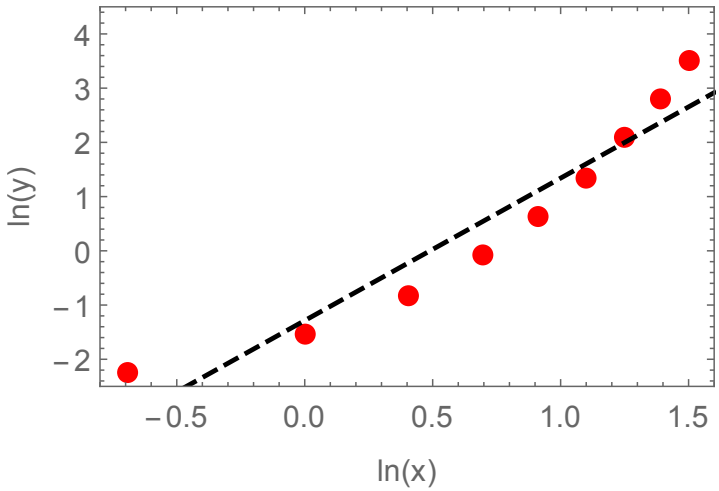

FIGURE 3. Best fit line and Log-Log data for power model.

We can see from Figure 4 that this method for generating the power function model produces a curve that looks pretty good for the left half of the data, but fails to capture the behavior for right half. Also note that the square error is four times larger than that of the quadratic model.

It is interesting to note that both the TI-84 Plus C Silver Edition calculator and the TI-NSpire CAS, running updated operating systems, give this answer as the Power Regression model. Now let's see just how far off these parameter values are from the true best fit power function model.

Using *Mathematica* [8] to find a numerical approximation to the parameter values that minimize the square error in (1), we get a power function model of

$$g(x) = 0.00612061x^{5.72753} \tag{6}$$

with a square error of $E = 1.84181$. Figure 4 shows the original data, the fit using the linearized model and the approximate best fit power function. It is clear that the approximate best fit power function captures the behavior of all of the data while using logarithms does not.

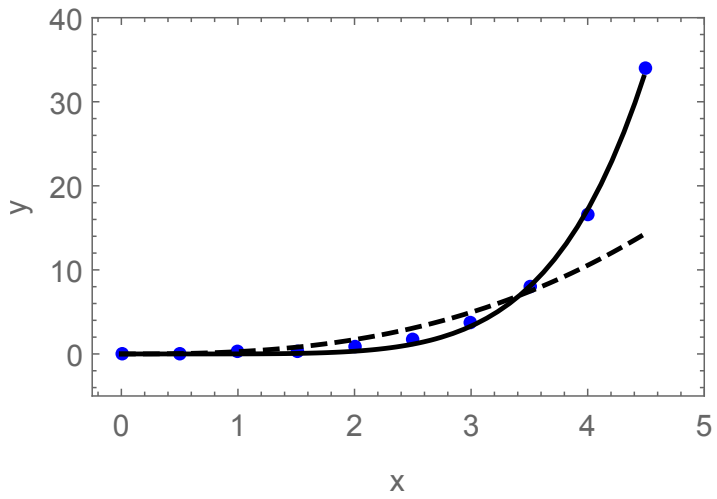

FIGURE 4. Best fit power function using Levenberg-Marquardt method (solid), and linearized method (dashed).

Please note that the technique that seems to be implemented in modern calculators, and is widely used in practice, generates a square error two orders of magnitude larger than the actual least-squares regression fit of a power function on this data!

Lastly, the exponential model. As in the power function model, we drop the first data point when looking at the linearized problem. Here there is no good justification for doing this, but it is a common practice to avoid the ln(0) disaster. Note that the exponential model will only pass through the origin if $\alpha = 0$, so dropping the first data point will add to the error.

In Figure 5 the data S_e and the best fit line, $Y = 1.44618X - 2.98586$, to this data are graphed. This seems to be a good fit for the data and leads to an exponential model, via (4), of

$$h(x) = 0.050496e^{1.44618x} \tag{7}$$

with a square error of $E = 0.0325555$. At first glance this seems to be a very good fit and is the exponential fit given by the calculators. However, once again, this is misleading.

Approximating the non-linear least-square error problem directly leads to an exponential model of

$$h(x) = 0.0501896e^{1.44856x} \tag{8}$$

with a square error of $E = 0.00420048$. While there is no visual difference between using the semi-log linearized method and the approximation to best fit exponential regression, the latter is an order of magnitude better in square error than in the linearized problem. Figure 6 shows this model.

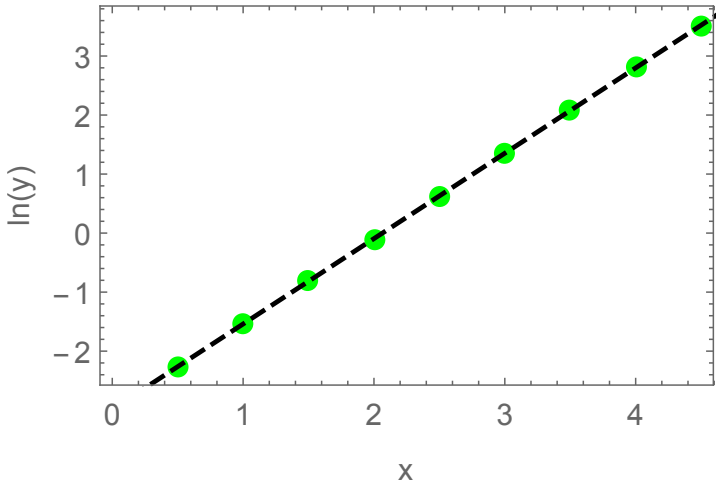

FIGURE 5. Best fit line and semi-Log data for exponential model.

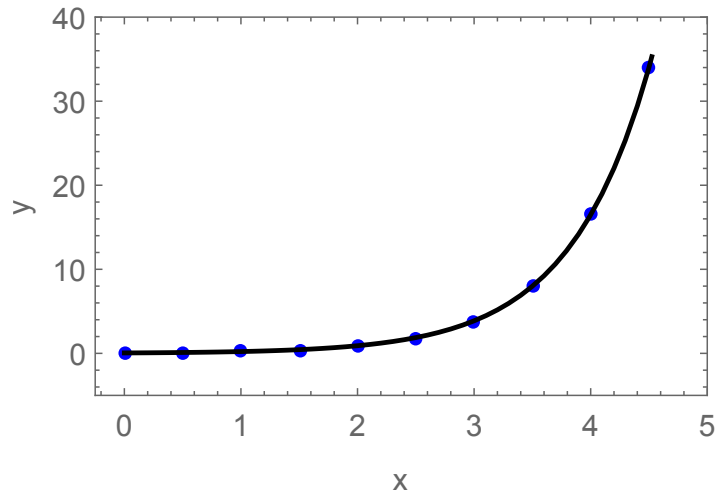

FIGURE 6. Best fit exponential function using approximate solution that minimizes the original error function.

4. To Log or not to Log?

When trying to fit the non-linear models, clearly stopping with the transformed parameter values leads to a fit that is not the best one possible and thus not the answer to the problem posed. The CAS software, *Mathematica* [8] uses an implementation of the Levenberg-Marquardt algorithm that is efficient and robust [10] at approximating the best square error fit of non-linear models. Thus using log's to linearize the models should not be used. This also avoids the ln(0) problem present in transforming the data.

References

[1] Carlos Almada and Laura Nunley, A College Algebra Approach to Least Squares, *J. of Math. Sci. and Math. Ed.* **5** 31–36.

[2] Richard L. Burden and J. Douglas Faires, *Numerical Analysis*, ninth edition, Brooks/Cole, 2011.

[3] Connally, Hughes-Hallett, Gleason, et. al., *Functions Modeling Change, A preparation for Calculus*, second edition, Wiley, 2004.

[4] Eric S. Key, A Painless Approach to Least Squares, *Coll. Math. J.* **36** (Jan. 2005) 65–67.

[5] Ron Larson, *College Algebra*, ninth edition, Brooks/Cole, 2014.

[6] David C. Lay, *Linear Algebra and its Applications*, third edition, Pearson/Addison Wesley, 2006.

[7] James Stewart, *Multivariable Calculus*, seventh edition, Brooks/Cole, 2012.

[8] Wolfram *Mathematica 10*, Documentation Center, `http://reference.wolfram.com/language/` **FindFit** command.

[9] Wolfram *Mathematica 10*, Documentation Center, `http://reference.wolfram.com/language/` **Fit** command.

[10] `http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm`