**Function Discovery – Two-parameter Models**

Last week’s lab investigated methods for determining if a data set follows a linear function (y= mx+b). The primary focus was on graphing model residuals and visualizing whether the residuals were randomly distributed (data was linear) or whether they followed a pattern (data was not linear). It was noted there are five important steps in fitting an equation to data:

1. **Model**: clearly identify an equation with adjustable parameters to be fit.
For the linear case the model is y = mx + b, where m and b are the parameters to be fit.
2. **Setup:** Prepare data for fitting
For the linear case this was simply plotting the data
3. **Fit**: Use software (in our case MATLAB’s Basic Fitting Interface) to fit the parameters.
For the linear case this was choosing the linear option and adding the equation to the graph.
4. **Predict**: Use the fitted equation to find predicted y values for each x value (the “fits”)
For the linear case the Basic Fitting Interface did this for us automatically.
5. **Plot:** Check and present the fit by plotting the data as points and the fitted values as a line
For the linear case this is handled automatically be the fitting interface
It is also good to plot the residuals (the fitted values minus the actual value for each x)
For the linear case, selecting the appropriate check boxes completes this for us.

**Step 1. Model: Finding the model among three candidates**

At the end of the lab a data set was graphed on a linear plot, a semilog plot, and a log-log plot to visualize which graphical fit produced the best linear line and subsequently was the ‘correct’ fit for the data set. These plots are used to test for linear, exponential (semilog), or power (log-log) models. However, they cannot be used to fit these models in MATLAB.

Table 1 shows these three models and their correspondence to the three plots. This table lists the model name and equation. It then shows the linearized form. Notice it includes two versions of the exponential model one fit with a base of 10 and one with a base of e. The two models with result in different B values but the predicted y values will be the same regardless of which is fit. Note throughout this document lower case letters are used for parameters and variables from the original equation and capital letters are used when they are from the transformed model.

**Table 1:** Basic Two-Parameter Models – Finding the likely candidate

|  |  |
| --- | --- |
| **Models** | **Plots to test with** |
| **Name** | **Equation** | **Linearized form** | **Plot Name (& axis spacing)** | **MATLAB commands** |
| linear |  |  | Linear Plot (both axes spaced linearly) | plot(x, y, ’p’) |
| Exponential |  | log10(y) = log10(b) + mx Y = B + Mx  | Semilog Plot(y axis spaced logarithmically) | semilogy(x, y, ’o’)  |
|  | ln(y) = ln(b) + mx  Y = B + Mx |
| power |  | log10(y) = log10(b) + mlog10x Y = B + M X  | Log-Log Plot(both axes spaced logarithmically) |  loglog(x, y, ’s’) |

If the data follows a linear pattern on one of the “Plots to test with” in Table 1, the expected model is the one in the same row with that plot. The linearized equations match the plots. For example in the linearized equation for the exponential model a log is taken of the y value but the x value is not changed. This corresponds to a semilog plot where the y-axis is logarithmically spaced and the x-axis is linear.

It is possible that none of the plots are linear and these models will not work for the specific data set.

**Step 2: Setup** for the Exponential and Power Law Models

Unfortunately, the fitting interface will not change its fit because the plot has been changed to a semilog or log-log plot. Some of you noticed this if you tried to fit the data for the assignment due today. Some extra work is needed in step 2: Setup, for these last two models. In order to fit these models the experimental data must be transformed to match the linearized models. Table 2 summarizes the required transformations for each model. Notice the transformation matches the variable transformed in the Linearized form of the equation.

**Table 2:** Data Transformations required for the nonlinear two-parameter models

|  |  |
| --- | --- |
| **Models** | **Data Transformation Required** |
| **Type** | **Equation** | **Linearized form** | **Description**  | **MATLAB** |
| Exponential |  | log10(y) = log10(b) + mx Y = B + Mx  | Take a logarithm of the y data | >> log10(y)  |
|  | ln(y) = ln(b) + mx  Y = B + Mx | >> log(y)  |
| power |  | log10(y) = log10(b) + mlog10x Y = B + M X  | Take a logarithm of both the x and y data.  | >> log10(y) &>> log10(x) |

MATLAB Logarithmic Functions

log(x) = the natural log of x (i.e., ln(x) in typical mathematics nomenclature)

log10(x) = the base 10 log of x (i.e., log(x) in typical mathematics nomenclature)

Next plot the transformed variables using the plot command.

* For the exponential model plot (base e) plot log(y) vs. x. In MATLAB use >> plot(x, log(y), ‘p’)
* For the power law model plot (base 10) log10(y) vs. log10(x). In MATLAB use >> plot(log(x), log(y), ‘o’)

**Never use the semilogy or loglog plot functions with the Basic Fitting Interface – it does not work!**

**Step 3: Fit** the chosen transformed model

Once these plots are setup, open the Basic Fitting Interface and fit a linear model to the transformed data. Show the equation and notice the slope and intercept. These are the parameters for the linearized model.

**Step 4: Predict** the y values

While the fitting interface yields a plot, this graph, its predicted parameters and fitted equation are all based on the transformed variables. However, fitted values for the untransformed equation are usually needed. The goal in this step is to find the fitted parameters and equation for the untransformed model and then use that equation to predict y values.

Look at the linearized models in Table 2. Notice the ***M*** is unchanged by the transformation. So the m value for the original model is equal to the M found with the transformed model.
However, ***b*** values have been changed, where B = log(b) – remembering that capital letters represent values in the linearized model and lower case letters represent values in the original model. So to transform the B (intercept) value found above the base of the log used in the step 2 transformation is used. In other words:

* If the original transformation was a natural log then b = eB. In MATLAB this is: >> b = exp(B)
* If the original transformation was a base 10 log, then b = 10B. In MATLAB this is: >> b = 10^B

Now the y-fit ($\hat{y}) $vector can be calculated using the non-transformed fit equation (from the Equation Column in Table 2) and parameters plus the original x vector.

For example, if our data was determined to be log-log then our non-transformed fit equation would be $\hat{y}=bx^{m}$ where m and b were calculated above and the x vector would the original vector of x values.

 **Step 5: Plot** the data and the predicted equation.

Graph y versus x as points and y-fit versus x as a line on one graph using the plot command and with the appropriate labeling commands we have learned earlier (e.g., xlabel, ylabel, and legend).

**You have created a fitted line for a non-linear data set!**

**Example: Capacitor Discharge Data (the Example Reviewed in Video 12.1)**

Figure 2 shows an example using these steps to fit discharging capacitor data. This data is in the EMP3.mat file as the variables V and t. Plots of this data shows that the model is likely exponential.

Example of Fitting a Data Set

**Fitting Capacitor Discharge Data**

1. **Model:** The data is straight on a semilog plot implying an exponential model.

 Exponential model equations:

 V = b10mt or log(V) = log(b) + mt

1. **Setup:** >> LogV = log10(V)
 >> plot(t, LogV, ‘s’)
2. **Fit:** fit a linear model (using Basic Fitting) 🡺 LogV = 1.99 - 0.429 t
3. **Predict:** Untransform >> b = 10^(1.99) (b = 97.7 V), m = - 0.429

 Predict >> V\_fit = b\*10.^(m.\*t)

1. **Plot:** on linear plot

 >> plot(t, V, ‘s’, t, V\_fit) plus labeling

**Figure 2:** Fitting the capacitor discharge data to an exponential model (log base 10). The five basic steps are outlined with typical MATLAB statements. For a power law model Step 2. Setup would need to also include transformation of the x variable and Step 4. Predict would be need to use the power law equation (see Table 2).

**Practice Problems:** All data vectors are available in emp3.mat. Work through the following problems.

1. Wind Power Generation[[1]](#footnote-2) (POWER vs. WIND): Prepare the transformed plots (semilogy and loglog) of the wind power (MW) vs. wind speed (MPH) data from last lab Treat WIND as the independent variable and POWER as the dependent variable (Step 1). Only complete the first step on this data.
Are any of these two parameter models likely candidates? **Discuss with a neighbor.**
2. LEGO Flowmeter Calibration (Flow vs. Read): Go through the five steps in Figure 2 for the flowmeter calibration data. Using them, select an appropriate model (you turned this in today); fit the model and prepare a linearly scaled plot of the data and the fit with appropriate labels.

**When you have done 1 & 2 (or gotten quite stuck) watch Video 12.2**

1. Bearing Life[[2]](#footnote-3) (Life vs. Temp): The life of a bearing was measured at several different temperatures. Bearing life is listed in Thousands of hours and Temperature is in degrees F.
2. Use plots to determine a likely model for this data.
3. Prepare a script that go through the steps 2, 4 and 5 for this data (in Figures 1 & 2).
Step 3 will have to be done manually external to the script. However, the input function can be used in your script, to ask for the resulting m and B. The script will pause at an input command and allowing the user to use the Basic Fitting interface to fit the model and then enter the resulting parameters. The script should then complete steps 4 and 5.

Please **turn in** (1) this script and (2) a graph of the non-transformed data (points) plus the calculated fit vector (line). Both graphs should show the fitted equation.

**Grading Rubric for Lab 13**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Expectation**  |  |
| 1 | Presentation  | Overall Presentation: problem clearly labeled and presented |  |
| 2 | Intro comments: include problem ID, student name and purpose  |  |
| 3 | Body Comments: identify logical steps and variables |  |
| 4 | Script | 2. Linearize & Plot: Prepare plot of transformed data with MATLAB fit line |  |
| 5 | 3. Determine Fit: Determine transformed fit parameters.  |  |
| 6 | 4. Predict: Untransform the b value and calculate fitted values |  |
| 7 | Overall: Above steps are complete (all steps present) and consistent |  |
| 8 | Result  | 5. Plot: Life vs. Temp (linear axes) for both the experimental data and the fitted model (two series, one plot). Curves should agree, Appropriate size. |  |
| 9 | Plot Format: including axis labels with units, legend, data is plotted with points; fit plotted with lines, font size is close to text size (≥ 9 point) |  |
| 10 | Model: A correct final model is presented on graph or separately on paper (may be hand written).  |  |

1. Data from Montgomery and Runger, *Applied Statistics and Probability for Engineer*s, 3rd Edition, Wiley, (2003). [↑](#footnote-ref-2)
2. Data from Palm, W. J., *A Concise Introduction to MATLAB*, McGraw-Hill (2008). [↑](#footnote-ref-3)