Fitting data with *Mathematica*

1. Generate a delimited text file (from LabVIEW, a text editor, Excel, or a similar spreadsheet application) with the x values (time) in the first column and the y values (temperature) in the second column. If you have uncertainties in the y values, they can be placed in the third column. The default tab delimiter in LabVIEW is the best choice. If you save files from Excel be sure to select the output as a ".txt" file. This will write a tab-delimited file.

   The statement regarding delimiters is true for the `Import` and `ReadList` commands. Your mileage will vary with other file input commands. You can make the `Import` command work for comma-delimited (.csv) files if you enter "CSV" as the second argument.

2. Download the *Mathematica* notebook NLSQfit240.nb (also available on Learning Suite in `Content` ⇒ `Thermal Measurements Lab` ⇒ `Fitting Curves to Experiment Data`). The notebook must be in the same folder as the data file generated in step 1.

   Mathematica provides a very capable nonlinear fitting routine (NonLinearModelFit). There are simpler routines available (Fit, FindFit, LinearModelFit, etc.) but NonLinearModelFit has some extra features in flexibility and capabilities. For instance, it will accept weights to allow for including the effects of uncertainties in the data. And it will provide considerable information on the quality of the fit and the uncertainties in the final fitting parameters.

   Due to the complexity of NonLinearModelFit, the notebook is provided so that you don't have to deal with getting everything right for it while still benefiting from the added capabilities.

3. Open *Mathematica*.

4. Click on `File` and `Open` and open NLSQfit240.nb – *be sure it is the one located in the folder with the data file.*

5. Enter the name of your data file between the double quotes after the variable "datafilename" in the `Input the experimental data` section.

   In this section, there are also three variables, `xcol`, `ycol`, and `errorcol`, that tell the notebook how to interpret the file. These tell the script which column in your data file correspond to the x values, the y values, and the uncertainty or errors in the y values. If you do not wish to include the uncertainties set `errorcol` to zero (0).

6. In the `Define the theoretical model curve and plot the initial guess` section you will need to define the model equation in the variable "theory." The independent variable is always referred to in this equation as x (assumed to be the first column in your data file). You have to use *Mathematica* syntax for

this equation. For instance an exponential curve $a\,e^{b\,x}$ would be represented as "a*Exp[b*x]" in this equation.

Note: The fitting routine used in NLSQfit240.nb requires that there be more data points available than the number of parameters in the equation to calculate the uncertainty in the fitting parameters. This means that if you want a linear fit $(a + b\,x)$ with uncertainties, you will need at least three data points. It will still give you a fit to the points but some brown text will explain that you didn't have enough points to determine the uncertainty in the fitting parameters. *However*, if you only give the notebook two data points to fit, it will assume you want a linear fit and will provide that fit in the form $fit = a + b\,x$ and will include uncertainties in the parameters if you have provided uncertainties in the y values.

7. Define the initial guess for your parameters in the "params" variable. There must be an entry for each parameter in the equation defined in "theory." With a nonlinear fit, it is usually wise to make as good a guess as you can so the fit will converge nicely.

8. There are also variables

   `logx` to determine if the x axis should be logarithmic ("True") or linear ("False"). [Safe value for Physics 240 is "False"]

   `logy` to determine if the y axis should be logarithmic ("True") or linear ("False"). [Safe value for Physics 240 is "False"]

   `showerror` to specify if error bars are to be shown if uncertainties in the y values are provided.

   `Col3AreErrors` determines how the values provided in column 3, if present, are interpreted. If this variable is "True" it is assumed that they are errors or uncertainties in the y values (σ_y), they are converted to a fitting weight ($weight = 1/(\sigma_y^2)$), and the fitting routine is told to treat these as variances. If it is "False" they are assumed to be actual fitting weights. A large weight for a given point will put more emphasis on that particular point than a point with a small weight. [Safe value for Physics 240 is "True"].

   If you have set `errorcol` equal to zero in the initial load, you need to set this to "False."

   `XLabel` is the text used to label the x axis on the plot.

   Note that an appropriate label will include both what is being represented by the values on the axis and the units for the values when appropriate.

   `YLabel` is the text used to label the y axis on the plot.

9. In the section `Perform Non-Linear Least Squares fit and plot it together with the data`, there are two variables:

   `ShowEquation` which governs whether the equation for the fit is shown on the top of the plot as a title. This is useful for plots to be included in your lab notebook. However, if you are creating a plot for publication it is usually best to leave this off (set to "False") and give the equation for the fit in the figure caption or the body of the text.

`FontPoints` specifies the font size to use on the plot labels. The font size doesn't change as you scale the size of a plot in Mathematica. The font size is important when you want to include a plot in a published document. For instance, most journals will use a finished plot width of $3\frac{3}{8}$ inches which is the width of a single column on a double-column page. All text on a plot (or any other figure) must be at least an 8-point font at the final plot size to be legible. If you use a 12-point font on your plot and want a $3\frac{3}{8}$ inch final width, you should make your plot *on the screen* $3\frac{3}{8} * 12/8$ inches so that the 12-point text will scale to 8 points when you reduce the plot to the final width.

10. Click on `Evaluation` and `Evaluate Notebook` to give it a try. Answer "yes" if Mathematica asks if it should run the initialization cells.
11. If your initial guess isn't adequate you can update it and run just the `Define the theoretical model...` section by clicking somewhere in that section and pressing either Shift-Enter or the Enter key on the keypad. Check the graph to see if your guess is any better and keep fiddling with the initial values until the solid line is vaguely similar to your data.
12. If you have changed your initial values after first running the notebook, you will need to execute the `Perform Non-Linear Least Squares fit and plot it together with the data` section by pressing either Shift-Enter or the Enter key on the keypad.

    Keep iterating these last two steps until you get a reasonable fit.
13. When the figure is the way you want (including changing the size to something more reasonable) you can right-click on the graph and select either `Print Graphic...` or `Save Graphic As....` Printing the graphic will bring up a dialog to select the printer, etc. `Save Graphic` will bring up a dialog box to allow you to specify the name of the output file and the folder in which it should be placed. If you click on the `Files of Type` box near the bottom you can select the format for your file including jpeg, bmp, tiff, eps, png, pdf, or others. You should pick a format appropriate for your text processing software if it will be included in a document.

    Be sure to properly size the plot before printing or saving.
14. Above the graph is the information on the fitting parameters and uncertainties in those parameters ("Standard Error" is the standard deviation). There is a section for 63% confidence ($\pm\sigma$) and 95% confidence ($\pm 2\,\sigma$) intervals. To the right, there is a square bracket enclosing those two sections. If you right-click on the bracket, you will have a drop-down box with many choices including `Print Selection...` and `Save Selection As...` just like those for the plot described above. This will allow you to have a nice copy for your notebook.

If you were performing a linear fit to two points, you would just get a section with three lines of text giving the parameters and standard error for the two parameters in the fit.

[Modified: January 16, 2019]