Fitting data with Matlab

1. Generate a delimited text file (from LabVIEW, a text editor, Excel, or another spreadsheet application) with the x values (time) in the first column and the y values (temperature) in the second column. You can also include the uncertainties in the y values in the third column. The default tab delimiter in LabVIEW is a reasonable choice. A comma delimiter will also work.

    - You don't necessarily have to have the columns organized as indicated but you will have to modify how the file is loaded either in FitExperiment.m or the manual commands.

2. If you wish to use the script FitExperiment.m, you can download it from https://www.physics.byu.edu/courses/experimental/docs/physics240/FitExperiment.m. In Learning Suite, you can find it at `Content` ⇒ `Thermal Measurements Lab` ⇒ `Fitting Curves to Experiment Data`. It is most convenient if you put it in the same folder as your data file(s).
3. Open Matlab.
4. Use the browse button (...) in the upper right-hand corner to change to the folder where your data are stored.
5. If you downloaded FitExperiment.m:

    (a) You should be able just to type "FitExperiment" at the ">>" prompt to run the script. It will prompt you for the name of your data file after which it will load the data and start cftool.

    - If your data aren't organized with the x data in column 1, y in column 2, and dy (if present) in column 3, you will have to edit lines 10-24 and set "xcol,", "ycol,", or "errorcol" to the correct values. It is also possible to use a subset by selecting a range of x values using "xminimum" and "xmaximum."

    (b) In cftool, you will need to select the type of fit in the upper-right panel. It usually defaults to "Polynomial" but there are several choices including "Custom Equation" if there isn't one of the predefined fits that is appropriate for your equation. A fit will be attempted as soon as you select an equation.
    If the fit isn't correct or if you see the word "NaN" (not a number) in any of the entries in the "Results" box you may need to help find a starting point or place limits on the parameters. This can be done by selecting the `Fit Options...` box. Try entering different values in the `StartPoint` column until the fit looks right (this is trial-and-error but you can get some guidance from the graph if you look carefully at it – for example what is the value at long times for an exponential or at t=0).

(c) Once you are satisfied with the fit you should save the results by clicking on `Fit` on the top menu bar and select `Save to Workspace...`. You should keep the default MATLAB object names ("fittedmodel," "goodness," and "output") so the script knows where to find the results.

(d) Exit from the Curve Fitting Tool by clicking `File` and `Close Curve Fitting`. After you exit from the Curve Fitting Tool you need to press any key to continue with the script. Be sure you have selected the "Command Window" if MATLAB doesn't appear to be responding.

(e) The script will now generate a plot of your data and the fit as well as printing the results of the fit on the screen. MATLAB will give the results in terms of a 95% confidence level. That means that you will take the difference in the given range and divide by 4 to get the standard deviation if that is desired.

The plot that is generated will be 3.375 inches wide with all the text in an 8 point font. This is what is expected for publication in most journals with the plot the width of one column on a double-column page. There are entries right below the call to "cftool" that you can uncomment to create a plot that is 5.063 inches wide with the text in a 12 point font. This gives a plot that looks nicer in your notebook when printed but will have the correct font size when it is reduced to 3.375 inches wide for inclusion in a formal paper.

(f) If you look at the commands below "cftool" in the script you will see several commands including "legend," "xlabel," "ylabel," and "title." You can either edit these commands before running the script or reissue the desired commands to change the appearance of your plot. Remember that you should always properly label the axes on your graphs including what is represented by the numbers as well as giving the units for those numbers. For instance, you would use "Temperature (K)" for an axis representing the temperature in Kelvins.

If you are going to put the figure in a document you (usually) should NOT put a title on the graph. The information you would normally put in a title should be included in the (usually required) caption on the figure. A title would be appropriate for a graph that you are going to put in your lab notebook.

(g) You can print the figure by clicking on `File` and `Print ...` in the corner of the *figure* window. If you wish to save the figure in a file for inclusion in a document you should select `File` and `Save As...`, which will bring up a window to select a file name and folder for the file. Note that you can select several different file formats from the `Files of Type` box along the bottom of the window. Common formats include eps, pdf, jpg, tif, and png. You should choose a format compatible with the text processing software you will use for your documents.

(h) If you would like a nice copy of your final parameters you can print the screen (`File` and `Print...` from the menu in the upper left corner of the main MATLAB window – you should have the “Command Window” selected *before* you select `File`). FitExperiment issues the “clc” command when it runs so you will have a fairly clean screen to print. If you want the values in a file you will need to use copy and paste to put them in your favorite text editor.

If you wish to enter the commands yourself, you enter the following. Text preceded by the symbols “>>” (the MATLAB prompt) indicate commands you should enter at the command prompt (don’t enter the “>>” – just the text that follows it). Any text not preceded by “>>” is an explanation and should not be entered in MATLAB. The sequence:

(a) >> data=load(‘filename’);
Replace “filename” with the name of your file. It must be enclosed in single quotes. Be sure to put the semicolon (;) at the end of the line, so it won’t print all your data on the screen.

(b) >> xvalues=data(:,1);
This gets the first column in the variable xvalues. If the desired values are not in the first column, replace the “1” with whatever column number those values are in.

For this command, as well as the following two commands, it is possible to extract just a subset of your data by using the command
>> xvalues=data(minind:maxind,1);
where minind and maxind are the minimum and maximum indices into the array of the data you wish to use in the fit. The same values of minind and maxind have to be used on all three of these commands.

(c) >> yvalues=data(:,2);
This gets the second column in the variable yvalues.

(d) If you have uncertainties in the third column give the command
>> weights=1.0./(data(:,3).∧2);
This properly defines the weights used in the fitting process from the uncertainties.

(e) >> cftool(xvalue,yvalues,[],weights);
This starts the interactive curve fitting tool. Replace “weights” with “[]” if you don’t have the uncertainties for your y values.

(f) In cftool, you will need to select the type of fit in the upper-right panel. It usually defaults to “Polynomial” but there are several choices including “Custom Equation” if there isn’t one of the predefined fits that is appropriate for your equation. A fit will be attempted as soon as you select an equation.

If the fit isn’t correct or if you see the word “NaN” (not a number) in any of the entries in the “Results” box you may need to help find a starting point or place limits on the parameters. This can be done by selecting the

`Fit Options...` box. Try entering different values in the `StartPoint` column until the fit looks right (this is sort of trial-and-error but you can get some guidance from the graph if you look carefully at it – for example what is the value at long times for an exponential or at t=0).

(g) Once you are satisfied with the fit you should save the results by clicking on `Fit` on the top menu bar and select `Save to Workspace...`. You should keep the default MATLAB object names ("fittedmodel," "goodness," and "output") so the following instructions will work correctly.

(h) Exit from the Curve Fitting Tool by clicking `File` and `Close Curve Fitting`. After you exit from the Curve Fitting Tool, you need to press any key to continue with the script. Be sure you have selected the "Command Window" if MATLAB doesn't appear to be responding.

(i) >> plot(fittedmodel,xvalues,yvalues);
This will generate a plot with your data values overlaid on the model fit.

(j) >> legend('description of data','description of fit','Location','NorthEast');
This will put up a legend for the plot. You should replace the two "descriptions" with appropriate descriptions of the items being plotted. The location is given by a compass point: "North" for the top of the graph, "NorthEast" for the upper-right corner, "SouthWest" for the lower-left corner, etc.

(k) >> xlabel('Label for x axis (units)');
Properly labels the x-axis. The axes of your graph should always be properly labeled with the item represented by the numbers on that axis as well as the units for those numbers.

(l) >> ylabel('Label for y axis (units)');
Properly labels the y axis.

(m) >> title('Plot title');
Provides a title across the top of the graph. You should (usually) NOT title a graph if you are going to put it in a document. The information normally put in a title will be put in the (usually required) caption for the figure.

(n) >> fittedmodel
Don't put a semicolon on this line. It will print on the screen the results of the fit giving the best fit parameters and the 95% confidence range in those parameters. That means you will take the difference in the given range and divide by 4 to get the standard deviation if that is desired.

(o) >> goodness
Again, leave the semicolon off. This will give parameters that indicate the goodness of the fit.

(p) You can print the figure by clicking on `File` and `Print ...` in the corner of the *figure* window. If you wish to save the figure in a file for inclusion in a document you should select `File` and `Save As...`, which will bring up a window to select a file name and folder for the file. Note that you can select several different file formats from the `Files of Type` box along the bottom of the window. Common formats include eps, pdf, jpeg, tiff, and png. You should choose a format compatible with the text processing software you will use for your documents.

At some point, you will need to adjust the dimensions and font size on your figures. The standard is generally that the text on the figure must be in at least 8-point font when printed to be legible. This means that you need to adjust the font size and figure dimensions so that you get the desired results when you put the figure in a document.

The APS recommends that the font size be 8-point when the figure is reduced to a width of either $3\frac{3}{8}$ inches or 8.5 cm, which is the width of a single column figure on a double column printed page.

(q) If you would like a nice copy of your final parameters you can just print the screen (`File` and `Print...` from the menu in the upper left corner of the main MATLAB window – you should have the "Command Window" selected before you select `File`). If you type the command "clc" before entering the last two commands (fittedmodel and goodness) you will have a fairly clean screen to print. If you want the values in a file you will need to use copy and paste to put them in your favorite text editor.

[Modified: January 16, 2019]