I. INTRODUCTION DATA MODELING: PARAMETER ESTIMATION THROUGH NON-LINEAR LEAST SQUARES CURVE FITTING.

In the first part of this tutorial lab you will follow along through many examples of using python to analyze experimental data through non-linear curve fitting, for the purposes of finding the values of unknown parameters of physics models you already assume to be valid. You should strive to become comfortable with both the statistical concepts as well as the python code. Ultimately you are encouraged to borrow (i.e., cut and paste) code snippets from this tutorial for use in future labs; however you must still understand what the code is doing. I have tried to write the code in a very transparent style (no clever tricks), but if you are new to computer programming, or to the python language particularly, I strongly urge you to view the online beginner python tutorials referenced in the syllabus.

Each numbered item below corresponds to a certain section of prewritten code in the accompanying DataLabPython.ipynb file, which you also need to download. Execute all of the code underneath each section heading of the corresponding number, as you reach that number in this document.

When you reach section III of this document, you should be at the end of the notebook. Create a new cell and start cutting and pasting code from the earlier sections, in order to generate code to complete your assigned curve fitting task.

1. Define the underlying law of nature. In an actual lab, this is taken care of for you ahead of time!! ← (Joke)
2. Generate samples for 30 discrete, evenly spaced, values of the abscissa over the range 0 to 4 seconds.
3. Plot the samples.
4. Generate a smooth plot of your underlying law of nature and overlay with the discrete samples.
5. Define a source of noise, simulating using an imperfect apparatus to make measurements of nature. Here, even with no input from the outside world, the instrument may nevertheless output a non-zero value of whatever it is measuring.
6. Generate a plot of 30 random noise samples, one for each abscissa value in step 2.
7. Calculate the mean and standard deviation of your 30 noise samples.
8. Plot a histogram of the noise samples.
9. Generate a plot of 30 samples of your specific function, now with the noise fluctuations added in. Overlay with the previous plots.
10. Define a model for a nonlinear curve fit to these data. Let us simply use the form of the law of nature that we know these data derive from (of course, in a real lab, we are TESTING laws of nature to determine, if possible, which is True.)
11. Perform a nonlinear least squares fit of your model function to your data. Report what the best fit values of your fit parameter, as well as its uncertainty.
12. Plot a smooth line of the best fit model, overlaid with the measurements. Why do not the two smooth curves agree?
13. Plot the fit residuals.
14. Calculate the mean and standard deviation of the fit residuals, and compare with the mean and standard deviation of the noise as calculated in step 7. Why does the mean not agree? The noise does not have zero mean. This is called a “systematic bias”. If you can know about the bias ahead of time (i.e., though independent control measurements) you can correct your data for the bias. Let us do that now by subtracting the estimated bias that is the mean value of task 7. Note that we are not subtracting what we “really” know to be the bias, (i.e., what we entered in step 4 when we generated our fake noise), because in a real experiment, we do not have access to the hidden “real” value of the bias; we only know the estimated bias, here coming from our control measurement of 1000 observations with no input. This is a crucial point to understand.

15. Find the mean value of the noise as a good estimate of the bias.

16. Subtract this estimated bias from the data, to get “bias-corrected” data.

17. Fit the corrected data, and view the results as before. Now the best fit model agrees with the underlying law of nature.

18. Plot the fit residuals from the new fit.

19. Calculate the standard deviation and mean of the new residuals. They have zero mean, and look randomly distributed. This suggests that the model is a good fit to the data. To make any stronger conclusion need error bars on the measurements, which we will get to in the second part of the tutorial.

20. Write a loop structure to perform 1000 such experiments, with new noise for each new set of observations, storing every fit parameter for later processing. Here we suppose we “zero-out” our instrument in the beginning, to eliminate any systematic bias in the measurements.

21. Plot a histogram of the list of 1000 parameter best fit values. Plot a smooth line of the best fit model, and overlay with the plot of of step (2). Do the two smooth curves agree? Why not? Note that the best fit model does not agree with the underlying “real” law of nature. Does this mean the data contradicts the theory?

22. Calculate the standard deviation of the list of 1000 parameter best fit values, and compare to the standard error from the one fit done in step 17. Why are they close? Why are they not identical?

At this point, we see that the standard error in the parameter estimation will be approximately equal to the standard deviation of a set of many independent estimates of that parameter, for many realized data sets. But we do not yet have a way to decide if a given set of data IS or IS NOT consistent with the hypothesis that the underlying law of nature is True. This is the subject we take up in the next section.

II. INTRODUCTION DATA MODELING: MODEL VALIDATION AND THE CHI-SQUARE GOODNESS OF FIT.

As we discussed in lecture, there does exist a very natural single parameter which can tell us whether our particular data set is likely to have been produced by random noise on top of the law of nature we are proposing, or whether it is unlikely that noise could have ever produced our data set, but rather it must be that the underlying natural law is in fact very different from our model. That parameter is the chi-square statistic. The value of this is reported by the non least-squares curve fitting algorithm we employ. However, its absolute value has no meaning, and is thus useless as a test of goodness of the model, unless we supply error bars with our data points. Again, this you have seen in lecture. In the remainder of this lab, we will endeavor to estimate error bars for our data points; then rerun the curve fitting. This time the chi-square statistics will give us meaningful information on whether our data point are or are not consistent with our hypothesized model fitfunc. We will purposely try a model that we expect to fail, just to see what things look like in this case.

From our knowledge of how error bars are connected with the standard deviation of the measurements of data, as discussed in lecture, we know that the standard error bars for our data should be none other than the standard deviation of the underlying noise. Nonetheless, let us go through this again, as it is a crucial point.

1. Generate 10000 observation of pure noise, with no bias.

2. Histogram these observations.

3. Find the standard deviation of these noise measurements.

4. Plot a gaussian distribution with a sigma of 1 and a mean of zero, and overlay with the histogram.

5. This means that 68% of the time, the instrument will deliver a reading that fluctuates away from the True value by an amount less than 1. However, it is always a possibility that on any one measurement the fluctuation could be smaller, or bigger. Even in this set, there are fluctuations as large as 3. Those larger fluctuations happen with less frequency, though. How often do you get a fluctuation larger than +3? Let us calculate this probability.
from theory by integrating the gaussian PDF from 3 to infinity.

6. Calculate the expected number of noise events which exceed 3 for 10000 measurements, and compare to the actual number in your dataset.

7. Now we will see how the standard uncertainty bars for an individual measurement are related to the standard deviation of a large set of independent, repeated measurements. Here, make 200 measurements at each of 10 points along the x axis, and plot every single one.

8. Now let's replace the scatterplot with histograms at each point, viewed from (above), where the height of the histogram bar is given by false color. We see they are scattered about a mean value with a normal distribution having a standard deviation of, here, 1 mV.

9. Calculate the standard deviation of each of these groups.

10. We proved in class that, in the absence of a systematic bias, the mean of such a large set of data is the best estimate of the true underlying value. However, when we report a single set of measurements, we must communicate that we expect it (and any other subsequent set of measurements) to vary randomly from the underlying value, with a 32% chance of disagreeing by more than 1 standard deviation. In this step, then, let us plot the true underlying values as blue x’ (note: we know the true value because we created the law of nature in this simulation), with the standard deviation of the histogram in the previous step now indicated with a vertical line, centered on the x, called the “error bar.” Then overlay with ONE data set, say the first of the 1000, and see how many of those points from that data set line within, and how many lie without, of the error bars.

11. For our 7 data points, how many do we expect, on average, to lie beyond the 1 standard deviation error bar?

12. Of course we normally do not know the underlying True values. Make a plot where we put the error bars on our measurements. Leave the underlying values plotted as x’s for reference!

13. Now perform a nonlinear least squares fit to the data with uncertainties. Let’s do it for the original data set from the start of the tutorial. Construct an “uncertainty vector”, which is an array having the length the number of samples, but with the y-uncertainty at every point.

14. Now perform the fit using the uncertainty vector. Notice the chi-square value.

15. Calculate the p-value of the chi-square of the fit (the `chi2.sf()` function in python). If it is around 0.5 then we say that these data are consistent with the model being a True representation of Nature.

16. Let us try to fit a model with a fixed offset of 15.

17. Examine the p-value of this chi-squared value. Is our data consistent with this new model? The answer is no – these data very strongly exclude this model as a possible law of nature. It also means that the best fit parameters reported by the fitting algorithm are totally meaningless, even though they come with nice looking uncertainties. This is the most crucial point of this exercise. If a model does not represent your data well, then the parameter estimates coming from a fit of that model to the data are worthless!

III. LAB ASSIGNMENT

Your assignment is to load in your individually assigned data set, and fit it to a model of your own devising. By plotting the data you will see that the data exhibits one peak. Model the peak as a gaussian function. Perform a nonlinear fit of the data to a model from which you can extract its position along the x axis and its width. Hint – you will need more free parameters than these two!! Please assume vertical error bars of ±0.7 volts on every point. Be advised that you will need to provide your python code with an initial guess of the best fit parameters, otherwise the fitting algorithm will likely return an error or a nonsense result. Here is an example of providing an initial guess vector `pguess` for a two parameter fit:

```python
pguess=[12,5]
(popt, punc, rc, d) = curve_fit_custom(peakfunc, filexlist, filedataset, pguess, yuncertaintyvector)
```

Plot a smooth curve of the fitted model against the data points, to make sure it looks like it is correctly capturing the peak structure of the data. What is the chi-square value of the fit, and what does it mean for your model?