

Lab #7

Physics 91SI, Spring 2013

Objective: In this lab, you will use the scientific Python libraries to produce more advanced, publication quality plots.

As usual, log on to corn and clone over the starter repository:

```
hg clone /afs/ir.stanford.edu/class/physics91si/src/lab7 lab7
```

Remember to `hg commit` often to save your changes, and submit your code at the end of the lab.

Part 1: Interactive Plot Evolution

For this part of the lab, you're going to follow along as I go through an interactive example on the projector. You all have a Python script named `3d_plot.py` and the necessary data (in pickle files, which we'll learn in a few weeks) in your `lab7` folder. Together, we're going to read through the script, see what plot it produces, and then add to that script in order to produce a clearer and richer plot that would be ultimately publication-quality!

Please follow along as I talk about bad aspects of the original plot and suggest different things we can add to the Python script to change the plot and how the data is presented. You should add changes to your script as I do them on the projector, and run your script each time I run mine.

Part 2: Improve Plots from Last Lab

Now that you're acquainted with the different options in Matplotlib and have heard a brief intro to the principles of good plotting, you should go back and improve on the plots you made in Lab 6. Please copy your two Python scripts from Lab 6 into your Lab 7 folder. If you haven't finished working on `data_analysis.py` since Tuesday, take the time to finish it now. (At the end, if you have time, you can play around with `wave.py` and the plots it produces.)

You should now add additional aspects to your plots in `data_analysis.py`, such as labels, titles, legends, change colors/markers, etc.

For `data_analysis.py`, you should create a single figure in `main()` and add two

subplots to it (you decide if they look better stacked vertically or horizontally). On the first subplot, display the noisy wave packet, and on the second one, display the cleaned-up wave packet. You should add appropriate titles/text to each set of axes that describe what data is shown and what changed between them. You should also make sure that both subplots have the same axis ranges or that the wave packet is centered in both sets of axes, so the comparison is clear and unbiased. Play around with having different colors or legends on each plot, or whatever else you think would make it look better! Experiment!

If you have time, do something similar for your plot from `wave.py`. Add labels and a title. Play around with line styles and colors. Maybe add a subplot that focuses on the y-value of a single x-point and just displays a dot moving vertically, representing something like a boat floating on a wavy ocean. You could do that subplot with a simple call to `ax.scatter()` for your subplot axis object. We encourage you to experiment as much as possible and get your own feel for what looks good and what doesn't.