3

Extracting the data you want from a text file

Reading:
Langtangen, Chapter 2
Matplotlib Web Site

3.1 Statement of the problem

Here's a problem that you frequently encounter as a scientist: you use a data acquisition system to measure some phenomenon, and although the data that gets written to disk is merely alphanumeric text (technical term is ascii—American Standard Code for Information Interchange for those of you who are interested), the format of this ascii file is not of the form that scientific graphics tools can easily plot.

What do you do?

One option is to import it into a spreadsheet, and use the functionality of a spreadsheet to extract the data points you want. I won't do this, because I don't know enough about spreadsheet functionality—I suspect most physicists are in the same boat on this.

Another option is to manually read the data file and type it by hand. This is fine for a small data file, but introduces the inevitable typo(s), and is a totally absurd approach to a data file with millions of data points.

What we need is a way to read in a data file, extract what we need and write out a new data file in a more convenient format; and this method should work equally well for a small file or a data set with millions of points.

3.2 An example

Let's make this more concrete with an example. Table 3.1 shows a short section of a 1000 line data file. The data is from an optical switch used to measure the period of a pendulum.

Table 3.1. A short sample from the data set.

time (s)	state	period (s)
0.5389116	1	—
0.6551832	0	
2.2663992	1	
2.3827892	0	
4.0025032	1	3.4635916
4.118984	0	
5.7299832	1	
5.8465832	0	
7.4661176	1	3.4636144
7.5827832	0	

A laser beam is sent through the air to photodiode, and a digital signal is recorded each time the pendulum enters or leaves a laser beam. Figure 3.1 schematically shows the pendulum breaking the laser beam at time t_a, leaving the beam at t_b, breaking it at t_c, and leaving the beam at t_d. The next breaking of the beam at time t_e (not shown) will then allow one to calculate the period as $T = t_e - t_a$.

The problem here is that the format of this data file is not readily readable by many plotting programs. What we'd like to do is filter through this data only extracting the data where we have a period measurement. When you do so, you'll then have a two column data set with time and period values only. Then, it is a simple matter to read the data file and plot it with (for example) a tool like *gnuplot* or (as in Figure 3.2), *Matplotlib*.

3.3 Details for this Assignment

1. Read the file PeriodData.dat, and extract only lines with actual period values. Create an output file called Filtered.dat (which will go in your data folder—see Appendix B for submission guidelines). I've posted one way (not the most elegant, but it works) to do this at http://people.usm.maine.edu/pauln/261downloads.html in file filter.py

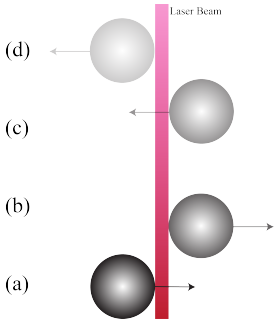

Fig. 3.1. The idea behind an optical switch—each time the pendulum enters or leaves the beam, a digital timing signal is recorded. In the figure, the pendulum is drawn a four different times; on the 5th crossing (not shown), one will have enough information to calculate the period.

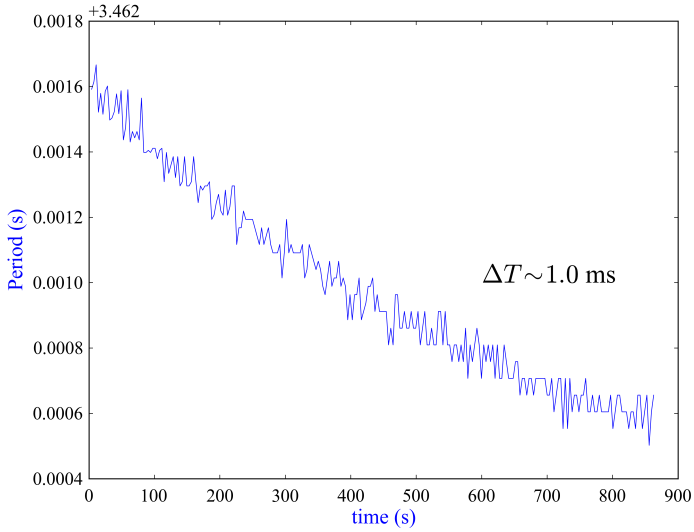

Fig. 3.2. A plot (using Matplotlib) of the period vs time data from the full data set. Notice that Matplotlib automatically pulled out 3.462 s from the period axis on the left, so that the vertical tics are space 0.4 ms apart, and over the course of 850 seconds, the period of the pendulum only changed by about 1 mS.

2. Plot this data from within your script using Matplotlib. I suggest you go to the Matplotlib gallery, find a simple x-y plot similar to what you want, and examine the code needed to create the plot.
3. Once you create the plot on the screen, you can click the disk icon on the plot to save it to disk (in your `LaTeX/Figures` folder) as a .png or a .pdf file.
4. As a part two to this exercise, still using the data from periodData.txt (at http://people.usm.maine.edu/pauln/261downloads.html) create a data file with all possible periods extracted; i.e. you can calculate the period as the difference in time between every 4th crossing:

$$\text{Period}_i = t_{i+4} - t_i$$

In this scheme, you'll end up with roughly three times as many period measurements. Create a new output file called tripleFiltered.dat, and plot this data file as you did with filtered.dat. Keep in mind that you will have to modify your program to produce this data file.

5. Now write a short LaTeX report about what you did. This is **not** a formal report, but simply an exercise to get your Linux/Python/LaTeX feet wet. When you're done, you'll have had experience with the three main tools we'll work with all semester, so the rest of the term will polish and deepen your familiarity with these tools.
6. Don't forget to submit your completed assignment according to the format specified in Appendix B. Your python scripts for part 1 and part 2 should of course be in your code folder. You do not have to use the code I posted to do part 1—if you have a better way, please feel free to ignore my code, and I'll include a handout of all the different methods people used when I hand back your assignment submissions.