Publication quality graphics, Exercises 2

18-Apr-2013

Create a new notebook called "GraphicsEx2" to complete the following exercises. Use a markdown cell to indicate the start of each new question or part of a question. Also include comments, explanations, and answers to any questions, using markdown cells. Be sure to save your work to github when you are done and/or leave for the day.

1. Download the American Math Society's short math guide (`ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf`) for LaTeX formatting of math equations and symbols. Use it as a reference to reproduce the following equations in markdown cells in your notebook. Be careful with parentheses and brackets - if they span over larger than one line, yours should too.

(a)

$$\psi_k(x) = A\left(\frac{ik - a\tanh(ax)}{ik + a}\right)e^{+ikx} \tag{1}$$

(b)

$$v_1 = m_2\sqrt{\frac{2G}{M}\left(\frac{1}{r} - \frac{1}{r_0}\right)} \tag{2}$$

(c)

$$f(\theta) = \frac{\hbar}{p}\sum_{\ell=0}^{\infty}(2\ell+1)e^{i\delta_\ell}\sin\delta_\ell P_\ell(\cos\theta) \tag{3}$$

(d)

$$\frac{d\sigma}{d\Omega} = \frac{k\pi^2(\pi-\theta)}{mu_0^2\theta^2(2\pi-\theta)^2\sin\theta} \tag{4}$$

(e)

$$\langle f\rangle = \frac{1}{\tau}\int_0^\tau f(t)dt. \tag{5}$$

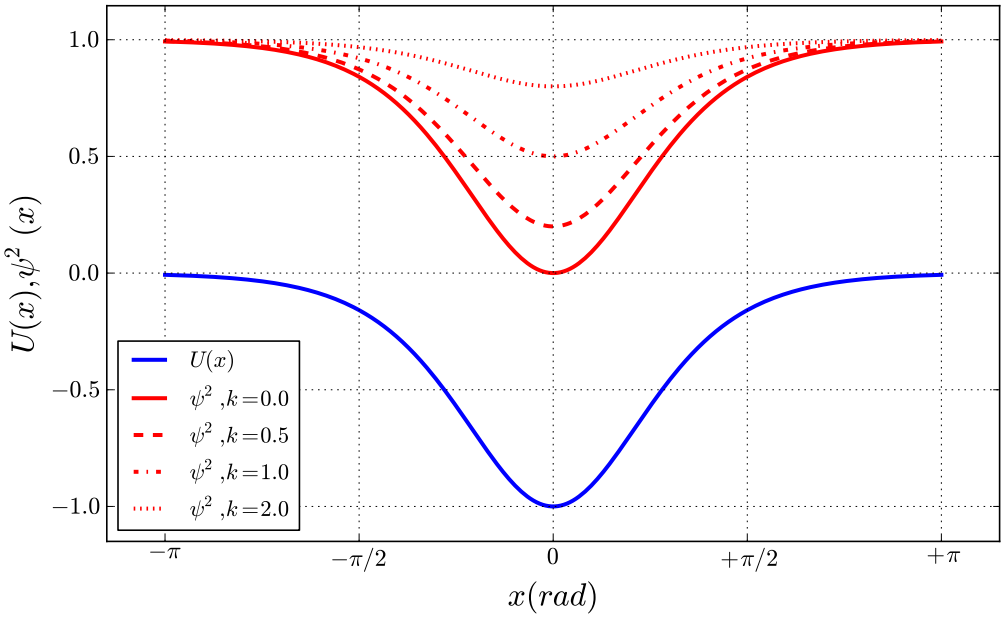
(f)

$$\Omega_S = \omega\frac{\sin\alpha}{\sin\theta} = \frac{L}{\lambda_1} = \omega\frac{\sqrt{\lambda_3^2 + (\lambda_1^2 - \lambda_3^2)\sin\alpha}}{\lambda_1} \tag{6}$$

2. In PHYS 212, many of you encountered the wave function shown in question 1(a) (Equation (1)). You plotted this along with the potential energy from the positive energy solutions for Schroedinger's equation:

$$U(x) = -\text{sech}^2(ax)$$

Reproduce the following graph of these two functions for the four values of k shown. Let A and $a = 1$.



Note the x-axis labels are fractions of π, not numbers. You'll need to look at the documentation for `xticks()` and `yticks()` to do this.

3. On PolyLearn, you will find a file called `sunspots.txt`, which contains the observed number of sunspots on the Sun for each month since January 1749. The file contains two columns of numbers, the first being the month and the second being the sunspot number.

(a) Write some code that reads in the data and makes a graph of sunspots as a function of time. Format the x-axis labels to show the month and year. You will have to look up documentation for how to do this. Be sure to think about how to make this graph *publication quality*, i.e. it should have labels, a legend, and customized (i.e. *NOT* default)

color and data representation choices. The following can be used to extract the data from the file into numpy arrays:

```
data = loadtxt("sunspots.txt",float)
x = data[:,0]
y = data[:,1]
```

(b) Now write code to display only the first 1000 data points on the graph.

(c) Finally, add code to calculate the running average of the data. Plot both the original data and the running average on the same graph, again over the range covered by the first 1000 data points. Be sure to include a legend and make it "pretty" (shorthand for *publication quality*).

4. There is another file on PolyLearn called `stars.txt`, which contains the temperatures and magnitudes of several thousand stars in the Milky Way near our solar system. You will use this data to construct a scatter plot of stellar magnitude vs. temperature, called a Hertzsprung-Russell diagram, after the astronomers who first plotted such data this way. The diagram is one of the fundamental tools of stellar astrophysics. For historical reasons, H-R diagrams are normally plotted with both the magnitude and temperature axes *decreasing* rather than increasing.

(a) Read in the data from `stars.txt` using similar commands as in question 3. Plot the data as is, with magnitude and temperature increasing in the usual way. Be sure to properly format your graph as publication quality.

(b) Now use the `xlim()` and `ylim()` commands to flip your graph so that it is a correct H-R diagram. There are several regions of the diagram where stars seem to cluster. Do a web search on H-R diagrams to find out what each of these regions are called and add some annotations to your graph to identify them.

5. Create a new notebook file called `PlotTemplates` in your main repository directory (not in the `Week3` directory).

(a) Add a markdown cell at the top with the heading "Publication ready plot template" and the description, "Simple graphing template for matplotlib functions and data plots.".

(b) Add another markdown cell beneath that one with a smaller heading "Line plots".

(c) Add one cell to create the data arrays for the following functions over the range $\{-5 < y < 5\}$:

$$E = \cosh(y)$$
$$|\mathbf{p}| = \sinh(y)$$

Use regular python `#comment` formatting to describe the steps in this cell.

(d) Add one more cell for all of the commands and customizations needed to make a publication-ready graph of the data from part (b). Remove the bounding box on the figure and center the horizontal and vertical axis lines (called spines) at $(0, 0)$. Use generic labels for the lines and axes. Remember, this is just a template. Again, use regular python `#comment` formatting to describe the steps in this cell. You'll need to get the current axis and manipulate the spines. Try these out:

```
ax = gca()
ax.spines['right'].set_color('none')
```

(e) Repeat step (b) using the heading "Subplots", and step (d), to create two side-by-side or top-and-bottom plots of your data from part (c) using the `subplot()` command. Be sure to include comments.

(e) Go back and look at question 4. In that question, you made a nicely formatted scatter plot. Now add new cells to repeat parts (b) through (d) for this new category ("Scatter plots") of figure. You can just create some dummy `(x,y)` data with the random number generator like this:

```
n = 200
X = random.normal(0,1,n)
Y = random.normal(0,1,n)
```

N.B. This notebook will be your personalized toolbox for plotting. Each time you create a new kind of plot in this class, you will add a new entry to this file. In this way, you will build up a library of plotting templates that you can use in current or future classes. Take a look at your solution to question 3. You might want to add an entry to your `PlotTemplates` file to remember how to plot time series data with dates on the x-axis...