Homework 2

Goal: To learn to write Matlab scripts and create 2D plots.

Due: Feb. 19

Read and work through the calculations in Spencer and Ware, Chapters 2 and 5. Explore the Plotting examples on Matlab-Monkey.com.

Problems:

1. ***Radius of a Black Hole.*** A black hole is defined as an object that is compressed such that nothing (including light) can escape its gravity. The Schwarzschild radius R_S of an object is the minimum radius to which an object of mass M must be compressed before it collapses in on itself to form a black hole. The Schwarzschild radius R_S is given by:

$$R_S = \frac{2GM}{c^2},$$

   where G is the Gravitational constant and c is the speed of light. The average density is then

$$\rho = \frac{M}{\frac{4}{3}\pi R_S^3}.$$

   Write a Matlab script that prompts the user for the mass of the object and displays the Schwarzschild radius and density. You may find it helpful to use example A02.m on the Matlab-Monkey.com website as a guide. Make sure you label your results with the proper units using the formatted `fprintf()` command. Format your result so that it will display a meaningful result no matter what mass is given. What is the Schwarzschild radius for the mass of the Earth? What is the corresponding density? What black hole mass gives a density less than that of air (i.e. 1 kg/m^3)?

2. ***Sunspot Numbers.*** Sunspots are cooler regions on the surface of the sun caused by magnetic field effects. Astronomers have observed the number of sunspots since the 1700s. In this problem you will plot the number of sunspots over time. Here are the steps:

   (a) You will first need to download a data file containing the sunspot numbers from the website http://www.sidc.be/silso/. You want the monthly mean total sunspot numbers in CSV (Comma Separated Values) format. Copy the file to your Matlab directory. Feel free to rename the file to something easier if you like, but be sure to keep the `.csv` extension.

   (b) Write a program to read in the data file. Since the data file contains only numbers and "`;`" separators, you can use the `load()` command (see Matlab-Monkey.com examples A11A.m and plot11.m). You can find descriptions of the data columns by clicking on `info` on the website next to where you downloaded the file. You'll want to plot the monthly mean sunspot number vs. the date in fractions of a year. (Once you load in the data into a matrix, you can double-click on the matrix in the Workspace window to examine its contents.)

   (c) Plot the data and see what it looks like. You'll want to play with the different plotting options to bring out the variations in the data. Here are some things to consider:

      - label the axes
      - set the axis limits and tick label spacing appropriately
      - save the plot to a `.pdf` file using the commands in Matlab-Monkey.com example `plot04.m`. You can set the aspect ratio of the plot here (i.e. how wide and how tall you want the plot).

- use markers to plot the data (don't connect the points with a line). Adjust the size of the data points appropriately.

(d) Use the `movmean(y,n)` command to smooth out the variations in the sunspot numbers. This command produces a vector in which each element equals the average of `n` data points centered around the point of interest. It works best when n is an odd number. It returns

$$f(x_i) = \frac{1}{n} \sum_{j=i-(n-1)/2}^{i+(n-1)/2} y(x_j)$$

where n is an odd number. Determine an appropriate value of n that averages out the month-to-month variations and highlights the longer-term periodic variations. Overlay this moving mean (in a different color) on the original data. Create a legend labeling the original, raw data and the smoothed data.

(e) Make a second plot similar to the first showing only the last two sunspot cycles.

3. ***Underdamped Oscillations (part 1).*** The differential equation for a damped harmonic oscillator is

$$\ddot{x} + 2\beta\dot{x} + \omega_0^2 x = 0$$

where β is the damping parameter and k is the spring constant. The natural frequency of the oscillator in the absence of damping is $\omega_0 = \sqrt{k/m}$. When $\omega_0^2 > \beta^2$, the pendulum is underdamped and the solution is

$$x(t) = Ae^{-\beta t}\cos(\omega_1 t - \delta)$$

where A is the initial amplitude and $\omega_1 = \sqrt{\omega_0^2 - \beta^2}$ is the angular frequency of the oscillator. Plot $x(t)$ for $A = 1$, $\omega_0 = 1$, $\delta = 0$, $\beta = 0.1$ for $0 < t < 100$. Plot the curve as a smooth curve. On the same plot draw a curve showing the damping envelope $x(t) = Ae^{-\beta t}$ as a dashed line in a different color. Label your axes and make a title giving the values of ω_0 and β. Create a legend to label the two curves. Save the result to a pdf file and adjust the page size so the plot looks nice (not too stretched one way or another).

4. ***Underdamped Oscillations (part 2).*** Let's make your previous plot a bit fancier. Use the `subplot()` command to create a grid of 4 plots (2 rows and 2 columns). Use the same values as in problem 1, but but pick a different β value. In the top left plot, plot $x(t)$ as before. In the top right plot, plot a phase diagram $\dot{x}$ vs. x. Differentiate $x(t)$ by hand to obtain $\dot{x}$. Pick a different value of β and repeat the above steps to make plots on the bottom row. Label each plot with the ω_0 and β values. Choose β values to show rapid and slow decays. You can adjust the maximum time as needed to produce a nice plot.

5. ***Quantum Harmonic Oscillator.*** The first three wave functions of the quantum harmonic oscillator are given by

$$\psi_0(y) = \left(\frac{\alpha}{\pi}\right)^{1/4} e^{-y^2/2} \tag{1}$$

$$\psi_1(y) = \left(\frac{\alpha}{\pi}\right)^{1/4} \sqrt{2}ye^{-y^2/2} \tag{2}$$

$$\psi_2(y) = \left(\frac{\alpha}{\pi}\right)^{1/4} \frac{1}{\sqrt{2}}\left(2y^2 - 1\right) e^{-y^2/2} \tag{3}$$

where $\alpha = m\omega/\hbar$ and $y = \sqrt{\alpha}x$. Use the `subplot()` command to create two columns of plots. In the left column plot $\psi_0(y)$, $\psi_1(y)$ and $\psi_2(y)$ vs. y. In the right column plot $|\psi_0(y)|^2$, $|\psi_1(y)|^2$ and $|\psi_2(y)|^2$. Set $\alpha = 1$. Shade the area under each curve for the $|\psi_n(y)|^2$ plots using the command `area()`. Set the scales of each of the ψ plots to the same range. Do the same for the $|\psi|^2$ plots. On each plot draw dashed vertical lines at the classical turning points: i.e. $y_{tp} = \sqrt{2n+1}$.