PHY 250L – Spring 2018
Computing tutorial 6

Welcome back to the PHY 250L computing tutorial! This week you began to learn the syntax of MATLAB, but keep in mind that you haven't learned anything new about *computing* techniques... just a new way of instructing the computer.

As you work through the problems below, please consult the MATLAB notes from class (see Sakai).

Problems for 4.5.2018 The following problems should be completed (in MATLAB, duh) and uploaded to Sakai by 09:45 on 4.4.2017. Each problem should correspond to its own MATLAB program (*i.e.*, each problem will correspond to a single file). The preferred names for the files are indicated in each problem.

1. `real_zeta.m`, 30 points
   The Riemann zeta function is defined for **complex** numbers s as

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} \tag{1}$$

   Write a program that outputs approximations of $\zeta(s)$ using the first 20 terms of ζ for the following **real** values of s:

$$s \in 0,\ 0.2,\ 0.4,\ ...,\ 5 \tag{2}$$

   Think of a simple way to generate an array of these s values – you'll lose two points if you have to hard-code in the values. It would be a good idea to write a separate function that computes the 20-term approximation for a given value of s, and then simply apply this function to your array of s values.

2. `integ1.m`, 20 points
   Write a MATLAB script that approximates the following integral

$$\int_0^5 \sin(2t)\, e^{-t^2/20}\, dt \tag{3}$$

   with an integration step of $dt = 0.0001$. Use a while loop.

3. `oned_walk.m`, 30 points
   In this problem, you'll generate the trajectory of an object that makes discrete movements along a single axis. This may seem like it's too simple to be of use, but this problem comes up frequently in physics – old-timers often refer to it as the "drunkard's walk" because of the following analogy:

   Picture a drunk person leaning against a lamppost in the middle of a very long sidewalk. We'll imagine the x-axis as being aligned with this sidewalk and the lamppost at $x = 0$. At time $t = 0$, the person (let's call him/her Terry) decides that it is time to go home, and Terry sets out making a step of length 0.5 m every second. There is only one problem: Terry is so compromised that each step is a random one! The

probability that Terry steps in the positive-x direction (at each step) is p_+, and the probability of a step in the negative-x direction is p_-. Each step is a random event and doesn't depend on the previous position.

Write a program that generates an array whose i^{th} element is Terry's position after i steps with $p_+ = p_- = 0.5$. Using this array, determine the number of times that Terry returned to the lamppost during her/his long stumble. Terry's walk should last for 500 seconds.

We'll return to this code next week to do some statistical analysis of the random walk. Yesssssssssss.