**Math 40a: Introduction to Applied Mathematics**
Lab 5

Upload your write-up and Python files to Latte to submit your assignment. The write-up should be self contained—the graders should be able to evaluate your lab based on the write-up without having to look at your separate code files.

## Overview

Today's computer lab has two aims:

1. Simulate random walks and verify the $\sigma = \sqrt{2Dt}$ outlined in class.
2. Write a project proposal.

## Standard Random Walks

We will simulate a random walk as in class. In the following code (random_walk.py), we consider a walker at $x = 0$ and flip a coin to let the walker move to the left or to the right:

```
#!/usr/bin/python
from random import random
# Starting position
a=0

# Number of steps
steps=20

# Carry out steps
for i in range(steps):
  # Print current position
  print(i,a)
  # Carry out a random step
  if random()<0.5:
    a+=1
  else:
    a-=1

# Print final position
print(steps,a)
```

Now we would like to run similar code for many different walkers starting at 0 and compile statistics over time. The file random_walkers.py does this.

```
#!/usr/bin/python
from random import random
import numpy as np
import matplotlib.pyplot as plt

# Time steps
tmax = 200

# Max spatial steps
nmax = 2*tmax

# Number of walkers and initial positions
nwalkers = 100
x = [0]*nwalkers

# Bin random walk positions into an empty 2D array (time, bin)
bins = np.zeros((tmax,nmax+1)) # tmax rows x nmax+1 columns
```

```
bins[0,int(nmax/2)] = nwalkers # all walkers initially at x=0

# Intialize statistics
meanx = [0]*tmax
widthx = [0]*tmax
# Time iteration
for i in range(1,tmax): # Walker iteration
  for j in range(nwalkers):
    if random()>0.5:
      x[j] += 1
    else:
      x[j] -= 1
    # Increment count for location of x[j]
    bins[i,x[j]+int(nmax/2)] += 1 # recenter bins around x=0

  # Statistics
  meanx[i] = sum(x)/float(len(x))
  widthx[i] = np.std(x)

# Sample plots
def walker_hist(i):
  plt.figure(i)
  bw = float(np.shape(bins)[1]-1)/float(np.shape(bins)[1])
  binx = -0.5*(np.shape(bins)[1]-1)+bw*(0.5+np.arange(np.shape(bins)[1]))
  biny = bins[i,:]
  plt.bar(binx,biny,width=bw) #width here is just a plot style option
  plt.xlim([-nmax/40, nmax/40])

walker_hist(0),walker_hist(1),walker_hist(100)
plt.figure()
plt.loglog(widthx)
plt.show()
```

Please note the following aspects of this program:

- The program simulates the trajectories of nwalkers random walkers. This is so that we can do the random walk many times.
- The program contains two loops. Within the loop the program repeats the task a specified number of times. The outer loop goes is over time steps. The inner loop moves each walker separately.

1. Run the program random_walkers.py. It outputs several histograms containing the positions of the walkers after different number of steps. Show how the histograms change with the number of steps by changing the limits of these plots using the plt.xlim command as in the walker_hist function. You may want to increase the number of walkers to get better statistics.

   The variable widthx saves the width of the distribution as a function of time—this is the standard deviation of the distribution in the positions of the walkers at each time. Plot widthx to show how the width of the distribution changes over time.

Next, we'll convince ourselves that the width grows according to our famous law

$$\sigma(t) = \sqrt{2Dt}.$$

A good way to do this is to use a log-log plot. Note that if we calculate

$$\log(\sigma(t)) = \log(\sqrt{2Dt}) = \frac{1}{2}\left(\log(2) + \log(D) + \log(t)\right).$$

Thus if we plot $\log(\sigma)$ versus $\log(t)$ we should find a straight line with a slope of $1/2$. We also might check the value of the diffusion constant: in class we argued that $D = a^2/2\tau$ , where $a$ is the lattice spacing and $\tau$ is the timestep. In our simulation both the lattice spacing $a = 1$ and $\tau = 1$ so that $D = 1/2$.

2. Do the following three things: 1) Make a log-log plot of $\sigma(t)$ and verify that the slope is 1/2 in accordance with the prediction. Start with the command plt.loglog(widthx). 2) Verify that the diffusion constant is correct. 3) Plot the average of the walkers over time. What happens to the mean? Does this make sense?

## Final project proposal

Write a short (one to two paragraph) description of your project proposal. Groups can consist of 1 to 3 people; all members need to submit the proposal along with this lab (the same proposal for the entire group). It should include:

1. Group Member Names (First and Last)
2. Which paper your project is based on.
3. A specific question that your project will attempt to answer.
4. One or two preliminary steps you plan to take along the way to answering the question.

Note that the project you turn in on the last day of class may end up being different from what you have written in your proposal. This is expected and completely acceptable – you should be willing to make adjustments along the way and not view this proposal as a binding contract.