PROGRAMMING ASSIGNMENT 5

Problem 1. In this problem we will explore random matrices.

(a) We start by sampling square $m \times m$ matrices whose entries are i.i.d random variables with the normal distribution $N(0, 1/m) = N(0, 1)/\sqrt{m}$. Use `normal` from `np.random` to create 100 samples for each $m = 2^j$ with $j = 1, \ldots, 8$. Compute the vector `w` of eigenvalues using `LA.eig`. If you inspect them, you will notice that they are usually complex numbers. So to visualize them we will use the command `plt.scatter(w.real,w.imag)`. You will need to use `plt.subplot` to generate 8 plots (one for each value of j). Make sure that the axis are scaled by the same factor.

Here are the first few lines of code:

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from numpy.random import normal
from numpy import linalg as LA


N=8
M = 100

plt.figure(figsize=(10,5*N/2.0))

for j in range(N):
    plt.subplot(N/2.0,2,j+1)
    m = 2**j
    for k in range(M):
```

What do you notice?

(b) Now instead of plotting the eigenvalues compute the expected spectral radius (i.e., the largest eigenvalue in absolute value), operator 2-norm, and smallest singular value $\sigma_{\min}$ (and hence the condition number), as $j = 1, \ldots, 6$. Also, make a conjecture about the limiting behavior of these three quantities. (You'll want to accumulate these quantities in the inner loop and then divide by the number of samples).

(c) Finally, estimate the tail of the probability distribution for $\sigma_{\min}$ by computing the proportion of random $m \times m$ matrices with $\sigma_{\min} \leq 2^{-1}, 4^{-1}, 8^{-1}, 16^{-1}, 32^{-1}$. Compare the plots for $m = 2^j$ and $j = 1, \ldots, 8$. What might you conjecture?