Homework 1 for
MSAN 502, Review of Linear Algebra

Due: Wednesday, July 20, 11:59pm

Strang Problems (4th edition)

1.1: 1,5,16,26
1.2:1,4ab,5,6,8,22
1.3: 6,7
2.1:12,16,18,19,28
2.2: 2, 6,7,11a,12,13,24
2.3: 1abc, 2, 3, 7, 13, 18, 27
2.4: 1, 2ab,5,14,22,
2.5: 5,8,9,12

Programming Warm-up

Problem 1

A. Familiarize yourself with the `numpy.linalg` package. Define a 10×10 matrix A whose entries are the numbers from 1 to 100 ordered consecutively going across rows (so $a_{1,1} = 1, a_{2,1} = 11, a_{10,1} = 91$, etc). Let $v = (1, 2, \ldots 10)$, and $b = (1, 1, \ldots 1)$. Compute Av and solve $Ax = b$ and $Ax = v$ using `linalg.solve`. Verify your solution by computing Ax.

B. Depending on how part A went, some of you are not quite ready to report to your director that you ”can't" do the above. You decide to “shake" your matrix by some noise, after all data usual comes in noisy. Use the `rand` command to build a 10×10 matrix $R(\epsilon)$ whose entries are i.i.d. random variables $X_{i,j} \sim U(-\epsilon, \epsilon)$. Verify now that you “can" compute the questions above for various values of $\epsilon > 0$ for the new matrix $A + R(\epsilon)$.

C. **Challenge:** How can you set up a brief study to understand if this noise idea is a good one to answer $Ax = b$ and $Ax = v$? Discuss your reasoning and discuss your results.

Problem 2
Write a python function Eliminate.py that carefully performs our elimination procedure as described in class. If it hits any fail points (including the temporary "permutation failure") it can just kick out and return an error. If it completes have the function perform back substitution as well. The final output should be the form of the upper triangular matrix as well as the solution vector. Compare your solver to the work you did on problem 1.

A. Can you keep track of what happens when trying to solve $Ax = v$ and $Ax = b$. If and when does your algorithm fail?

B. Compare the speed of Elimnate.py vs using `linalg.solve`. To do so modify your $R(\epsilon)$ from above so it can be of any size $n \times n$. Then report the average run time of 10 solver runs of $R(.5)x = w$ (where w is a vector of all ones) for $n = 100, 1000, 5000, 10000$. You may have to quit on the last one.