Ay190 – Worksheet 9

Scott Barenfeld

Date: February 7, 2014

I worked with Daniel DeFellipis.

1 Problem 1

My script reads A and $\mathbf{b}$ from the provided tables. It prints the shape of A and $\mathbf{b}$ to make sure A is $N \times N$ and $\mathbf{b}$ is of length N. It also computes the determinant of A to make sure it is non-zero and checks if any diagonal elements of A are 0 so the rows can be rearranged. Finally, it prints $\mathbf{b}$ to make sure $\mathbf{b}$ is not $\vec{0}$. These are the conditions required for Gaussian elimination to be implemented and the system to be solved. The LSEs are 10, 100, 200, 1000, and equations, and can be solved.

2 Problem 2

I used a Gaussian elimination script found at `https://gist.github.com/num3ric/1357315`. To test this script, I used the following LSE:

$$4a + 3b + c = 10a + 2b - c = 53a - b + 2c = 1$$

The solution is left as an exercise for the reader. Just kidding. It's

$$a = 1,\ b = 2,\ c = 0$$

After finding and fixing a few mistakes in the script, it produced the correct solution.

I then used this script on the LSEs provided. Because I like trees (even electronic ones), I'm only including my solution for the system of 10 equations. I find:

$$\boxed{x = [-5.51, -3.32, -2.98, 2.14, 3.59, 7.28, 8.87, -6.81, 3.04, 5.37]}$$

Using Python's timeit() function, I find Gaussian elimination with the script I'm using takes at least 90 minutes (I stopped it part way through solving the 2000 equations LSE, sorry Mr. TA, bacon burgers were calling).

3 Problem 3

As a faster method, I used Python's built-in numpy.linalg.solve() function, which solves all five LSEs in only 14 seconds. Python is good at stuff.