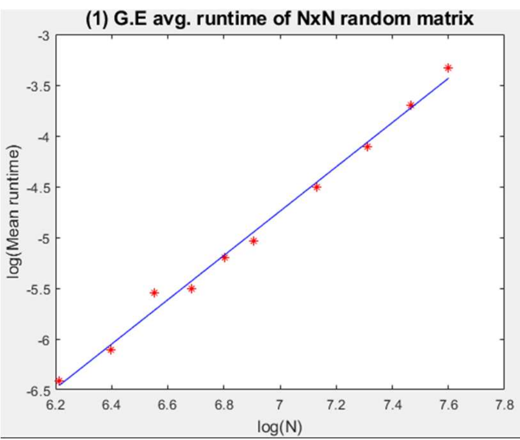
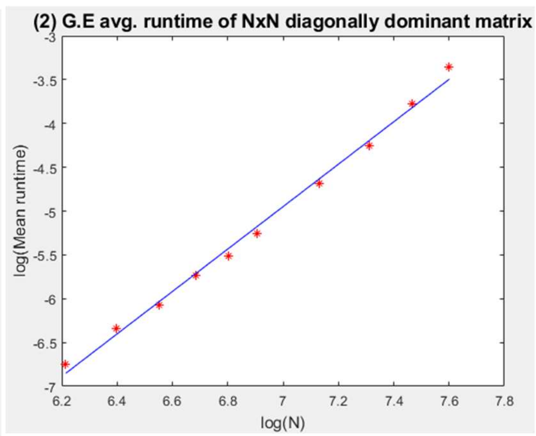
Computing Assignment 3

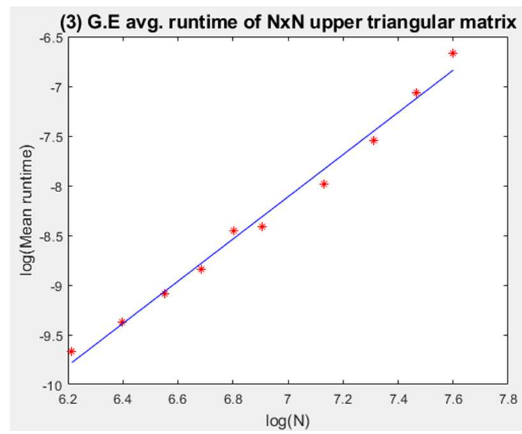
Daniel Todd
301428609
D100

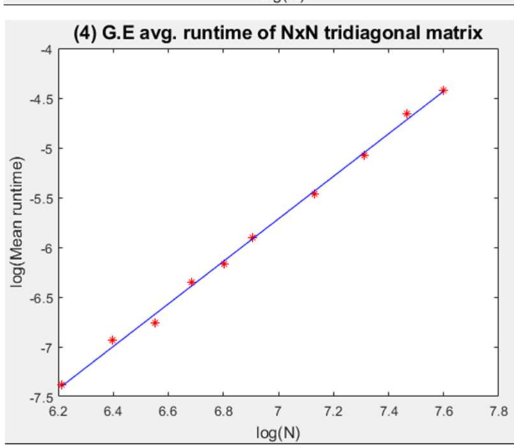
What I did:

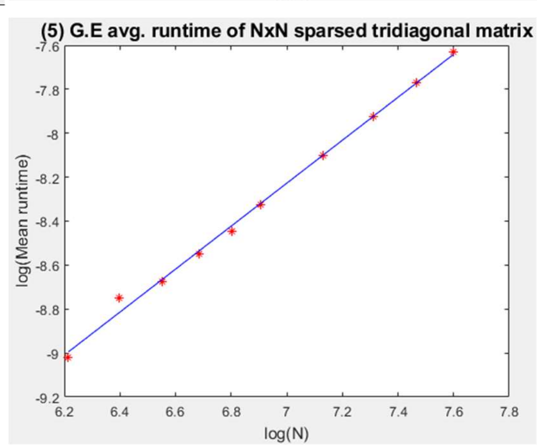
For my values of N, I chose 500, 600, 700, 800, 900, 1000, 1250, 1500, 1750, 2000, as we want large N for accurate flop counts and any number greater than 2000 increases computing time substantially. Then, for my number of trials (M) I chose 500, as this is the maximum number of trials that yields total a total runtime of under 5 minutes and seems high enough to get consistent results. I then, for each matrix provided in the assignment, calculated the average runtime of Gaussian Elimination (G.E) for each N over M. I then plotted log(N) vs log(mean runtime) for each matrix provided in the assignment. After plotting, I fitted a liner line to each graph, and computed the slopes of each line. The graphs containing the linear lines of these slopes can be found below, as well as the computed slopes of each line.












```
>> CA3
Slope of best fit line for matrix 1: 2.1807
Slope of best fit line for matrix 2: 2.4222
Slope of best fit line for matrix 3: 2.1229
Slope of best fit line for matrix 4: 2.1374
Slope of best fit line for matrix 5: 0.97749
```

Comparison of results:

We know the flop count of Gaussian Elimination on a random matrix to be O($\frac{2}{3}n^3$) . If a program's runtime was linearly related to the amount of operations the program performs, this would imply that the time to execute gaussian elimination on a random matrix would grow cubically. However, given the data produced we can see that the time to run the program grows logarithmically, as we can plot a linear line between log(N) and log(mean runtime). We also know the flop count of Gaussian Elimination on a random, diagonally dominant matrix to be O($\frac{1}{3}n^3$) , as no pivoting needs to take place, and the flop count on a random, upper triangular matrix to be O(n^2) . This means that the number of operations performed during Gaussian Elimination relative to a random matrix should be halved for diagonally dominant matricies, and reduced a significant amount for upper triangular matricies. However, the runtime data produced shows an increase in runtime for diagonally dominant matricies, and only a slight decrease in runtime for upper triangular matricies. Given the observed disconnect betweeen the runtime of Gaussian Elimination on these matricies and theoretical flop counts, we can only assume that the flop count of a program is not linearly related to the runtime of a program, and that there is some other factor at play.