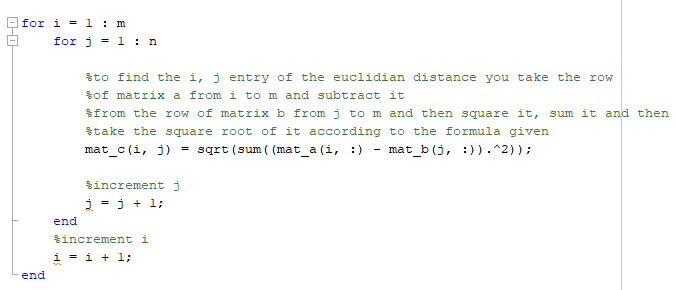
James Murphy

CS 381 - Image Processing

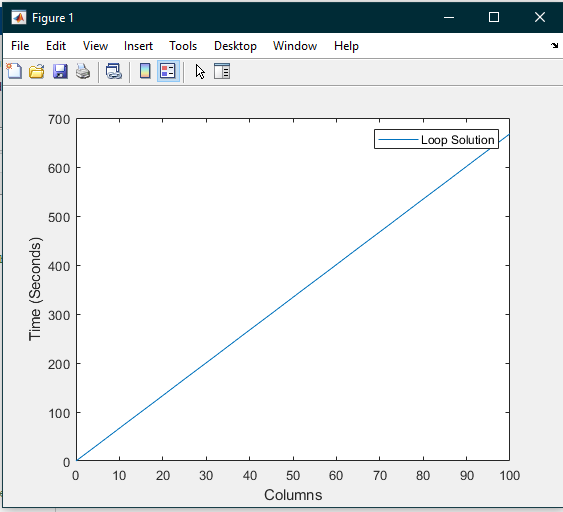
Homework #1

Problem 1 : Part 1

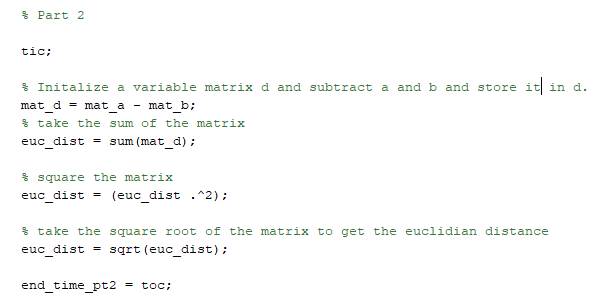
For part Problem 1 Part 1 we’re finding the euclidian distance between 2 matrices. For my implementation I took 2 matrices and randomly generated entries in a 100 x 100 matrix. To find the Euclidian distance you have to take the sum of the row from i to column n and subtract it from the row of matrix\_b using the j loop as the row variable. That way it’s mat\_A(row1, n) - mat\_B(row1, n) , mat\_A(row1, col) - mat\_B(row2, col) …. mat\_A(row2, col) - mat\_B(row1, col) … mat\_A(i, col) - mat\_B(j, col). Sum those values together, square it and then take the square root of it and then store those values in the i, j pair of a new matrix c.



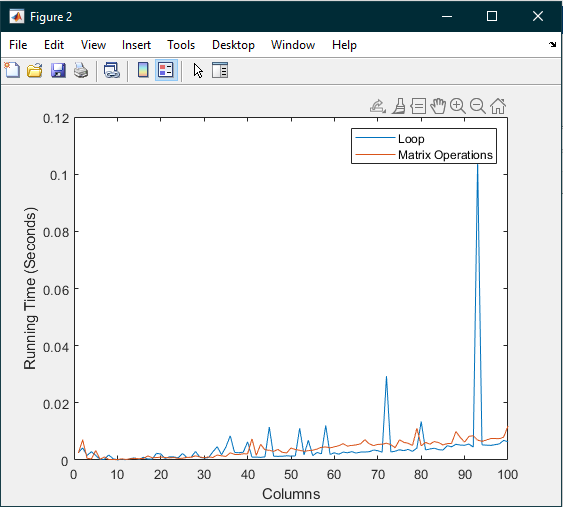
Graph of Part 1 :



Problem 1 : Part 2 : We take the randomly generated matrices a and b and subtract them and store it in d. Then we sum the values of matrix d together and store into a new matrix euc\_dist. Take the euc\_d matrix, square it and then sum it.

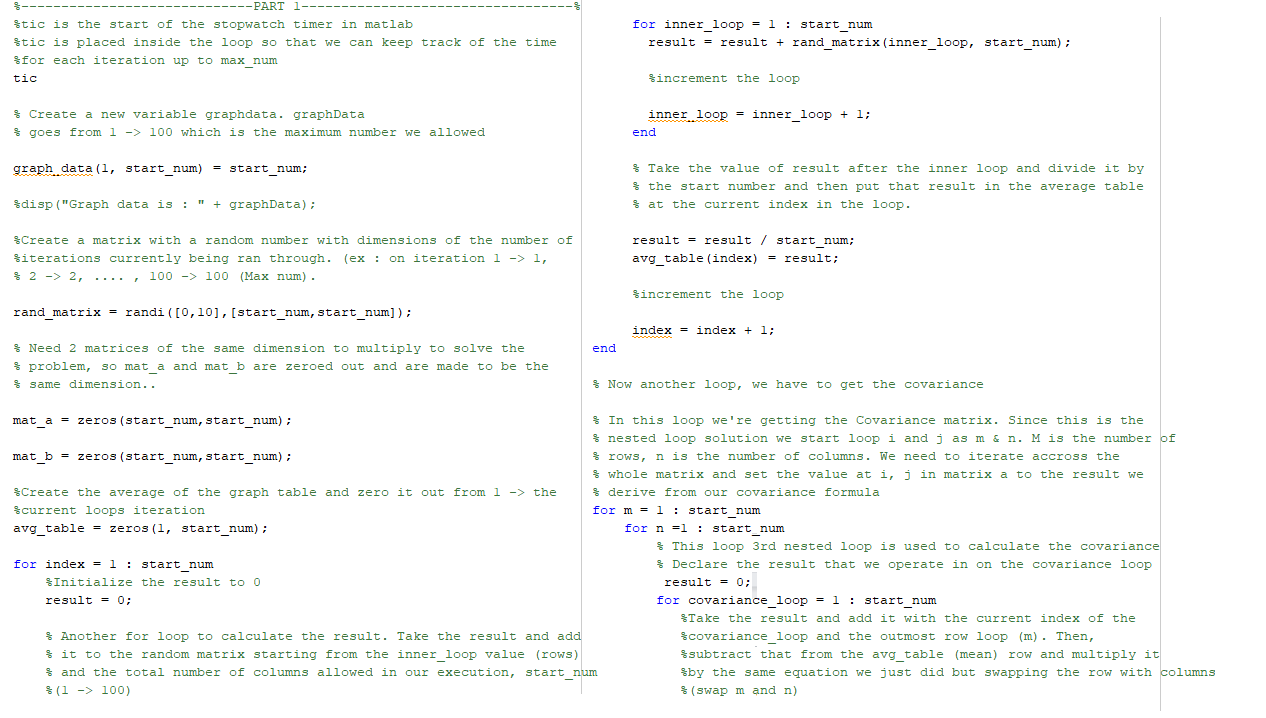


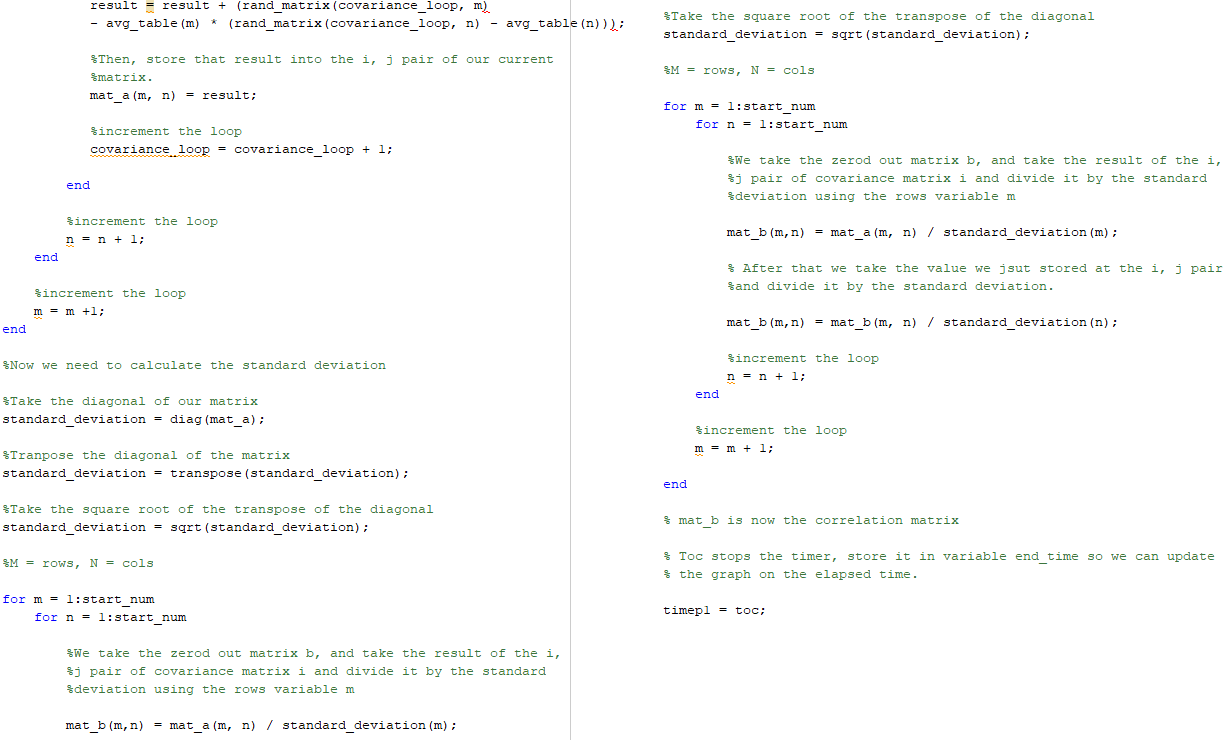
Graph Problem 1 Part 2 :



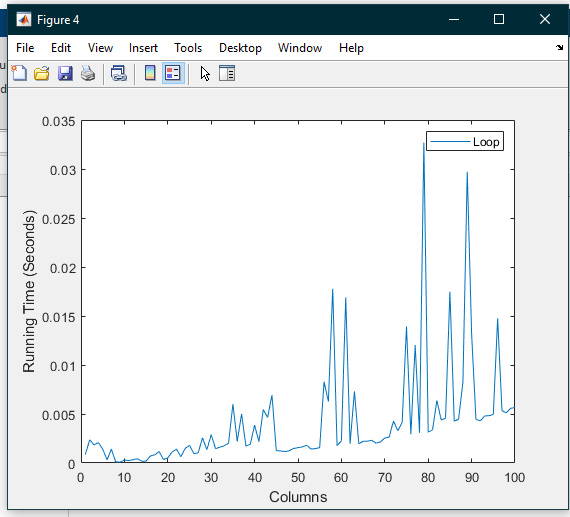
Problem 2 : Part 1 :

Sorry for the long code, it can’t fit in one image and I commented on it heavily. We create a variable max\_num and set it to 100. Then create a random matrix with values. We need 2 matrices of the same dimension so we zero out a matrix of size 100 x 100. We then need to find the mean / average. So a table is created for it and it’s zeroed out from 1 -> 100. Then a loop is created to calculate the result. The result is taken and added with itself and the random matrix from 1 -> n which is 100. Then we take the result and divide it by 100 since we did 100 iterations and set the current index in the loop to that value. After getting the average table we can now find the covariance. We initialize another result variable and take the result and add it with the (random\_matrix(loop index, and the row index) - the average in our table at m \* the random\_matrix(loop index, and the column index - the average in our table at n). Then we store that result into the i, j pair of the matrix. After that we need to calculate the standard deviation. So we take the diagonal of the matrix we just stored the i, j pairs in transpose it and square root that matrix. Then we take matrix b and divide both matrix a and b at m, n in the loop by the standard deviation at m and n respectively.



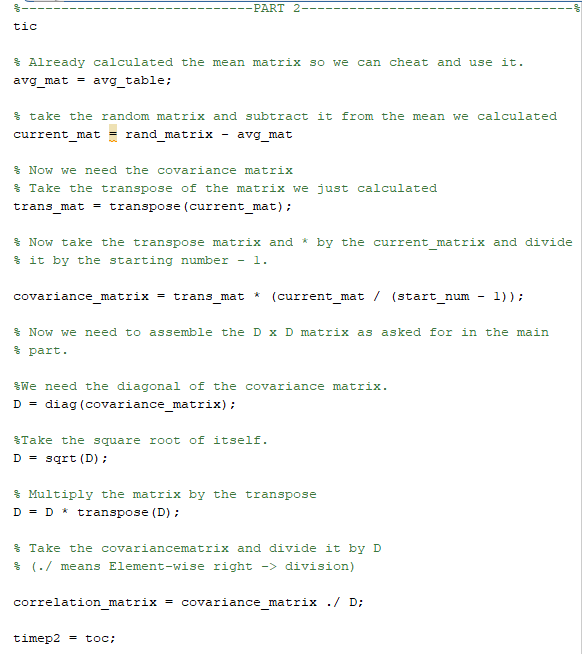


Graph of Problem 2 Part 1 :



Problem 2 Part 2 :

In part 2 of the problem, I was able to ‘cheat’ because we already had the mean / average matrix calculated from the first part of the problem so I used that. I created a new matrix called “current\_mat” and subtracted it from the rand\_matrix previously created from the first part of this problem. I took the transpose of current\_matrix. To calculate the covariance matrix we needed to take the transpose matrix and multiply it by the current\_matrix / the iteration in the “overall” loop the whole problem is in - 1. Next to calculate D we take the diagonal of the covariance matrix, square root it and then multiply it by the transpose of D. After that to calculate the correlation matrix we take the covariance matrix and divide it by D.



Graph of Problem 2 Part 2 :

