EAS 596, Fall 2019, Homework 5
Due Friday 11/8, **3:30 PM**, Box outside Jarvis 326

Work all problems. Show all work, including any M-files you have written or adapted. Make sure your work is clear and readable - if the TA cannot read what you've written, they will not grade it. All electronic work (m-files, etc.) **must** be submitted through UBlearns and obey the following naming convention: `ubitname_hw5_pN.m`, replacing ubitname with your ubitname and `N` with the problem number. Any handwritten work may be submitted in class.

All two point problems will be graded according to the following scheme:

- 2 Points: Solution is complete and correct.
- 1 Points: Solution is incomplete or incorrect, but was using correct ideas and concepts.
- 0 Points: Using incorrect ideas and concepts.

All four point problems will be graded according to the following scheme:

- 4 Points: Solutions are complete and correct. Code runs with no need for modification.
- 3 Points: One mistake in the code and it is easily found. Code runs after the modification.
- 2 Points: Two to three minor mistakes in the code, which are easily found. Code runs after the modification.
- 1 Points: Many mistakes in the code. No attempt will be made to modify it to run.
- 0 Points: Code has major conceptual issues.

Note about the programs: You are allowed to use MATLAB point-wise operations such as `./` along with functions such as `floor`, `size`, `zeros`, `eye`, `norm`, `dot`, etc. You are not allowed to use builtin functions `lu` or `qr` unless explicitly told to.

1. (4 pts) Write your own MATLAB function that accepts a matrix A and computes the reduced LU decomposition of A (without partial pivoting). The function call must be `[L, U] = ubitname_hw5_p1(A)`. Do *not* have your code output any text to the terminal or return

any results other than `L` and `U`. If your code does not follow these guidelines you will be penalized. Hint: Test your code by creating a random matrix, computing the LU-Decomposition, and then checking the difference between `A-L*U`. Please upload your function to UBlearns. This problem does not require a hardcopy submission.

2. (4 pts) Write a MATLAB script which produces the time needed to compute the LU factorization for random square matrices from a size $n = 10$ to $n = 1000$ using the function written in Problem 1. You may find the functions `logspace` useful to determine your matrix sizes and `tic` and `toc` useful for timing. You must use at least 50 matrix sizes. Use the following MATLAB command to generate the random matrix (this ensures its strictly diagonally dominant and does not require partial pivoting): `A = rand(n,n)+n*eye(n)`, where n is the size of the linear system. This script should produce a single plot of the time required versus the matrix size on a log-log plot using the `loglog` command. Include a line showing the long-term trend of the time required. Does the trend hold for small n? Why or why not? Be sure to *only* include the time required for the LU decomposition, and not any extraneous functions, such as the generation of the random matrix. Properly label the resulting figure along with the x- and y-axis. Your script should *not* produce any output other than the required plot. Please upload your script to UBlearns. This problem does require a hardcopy submission.

3. (4 pts) Write MATLAB functions to compute a QR decomposition of a matrix A using a) Classical Gram-Schmidt and b) Modified Gram-Schmidt. The function call for each must be `[Q, R] = ubitname_hw5_p3a(A)` or `[Q, R] = ubitname_hw5_p3b(A)`, as appropriate. Use each of your two MATLAB functions to compute a QR decomposition of the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 7 \\ 4 & 2 & 3 \\ 4 & 2 & 2 \end{bmatrix}$$

Compare your results. Please upload your functions to UBlearns. The QR decomposition using your code and a comparison must be submitted via hardcopy.

4. (4 pts) This problem will study the loss of orthogonality that can occur using the Gram-Schmidt procedure for ill-conditioned matrices. Write a MATLAB script that will compute the QR decomposition of the Hilbert matrix from sizes 2 all the way through 10 (use the `hilb` command in MATLAB) using your classical Gram-Schmidt and modified Gram-Schmidt QR functions, along with the MATLAB's `qr` function. Plot $\|Q^T Q - I\|$ versus matrix size on a `semilogy` plot for each method and comment on your results. Make sure the properly title your plot using the `title` command, identify the results using the `legend` command, and properly label the x- and y-axis. Upload your script to UBlearns and submit comments on the results via hardcopy. It is not necessary to submit a copy of the graph on hardcopy.

5. Consider the data in following table:

$$x = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$
$$y = \begin{bmatrix} -0.02 & 1.1 & 5.0 & 8.5 & 17.3 & 26.1 & 42 \end{bmatrix}$$

Create a MATLAB script which does the following:

(a) (2 pts) Find a least-squares solution to fitting the data using a linear function $f(x) = a_0 + a_1 x$.

(b) (2 pts) Now fit the data using a quadratic polynomial $f(x) = a_0 + a_1 x + a_2 x^2$.

Plot the data and the each resulting regression line. Be sure to properly title plot and identify the data and each fit.

(2 pts) Which of the two functions do you think is the more appropriate fit of the data? You must support your answer numerically.

Submit the MATLAB script to UBlearns and an explanation of which function is more appropriate via hardcopy.

6. (2 pts)

(a) Determine all eigenvalue and eigenvectors of the matrix $\mathbf{A} = \begin{bmatrix} 5 & -4 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 5 \end{bmatrix}$.

(b) Is it possible diagonalize matrix $\mathbf{A}$? Why or why not?

This problem must be submitted via hardcopy. You must determine the eigenvalues by hand, although you can check them using computational resources.