COMPUTER SCIENCE 349A, SPRING 2019
ASSIGNMENT #4 - 20 MARKS

DUE FRIDAY MARCH 22, 2019 (11:30 p.m. PST)

This is a really large class and the logistics of grading assignments are challenging. Me and the markers require your help in making this process go smoothly. Please ensure that your assignments conform to the following requirements - any violation will result in getting a zero for the particular assignment.

- All assignments should be submitted electronically through the ConneX course website and shoud be **SINGLE PDF FILES**. No other formats will be accepted. Handwritten answers are ok but they will need to be scanned and merged into a single pdf file together with any code examples and associated plots.
- The assignment number, student name and student number should be clearly visible on the top of every page of your assignment submission.
- **PLEASE DO NOT COPY THE ASSIGNMENT DESCRIPTION IN YOUR SUBMISSION**
- The asnwers to the questions should be in the same order as in the assignment specification.
- Some of the questions of the assignments are recycled from previous years but typically with small changes in either the description or the numbers. Any submission that contains numbers from previous years in any questions will be immediately graded with zero.
- Any assignment related email questions should have a subject line of the form CSC349A Assignment X, where X is the number of the corresponding assignment.
- The total number of points for this assignment is 20.

1. Consider the following system of linear equations of order $n = 3$.

$$\begin{aligned} 2x_1 + x_2 - x_3 &= 1 \\ 5x_1 + 2x_2 + 2x_3 &= -4 \\ 3x_1 + x_2 + x_3 &= 5 \end{aligned}$$

   (a) **(2 points)** Solve using naive Gaussian elimination. Show all your work.

   (b) **(2 points)** Solve using Gaussian elimination with partial pivoting. Show all your work.

   (c) **(2 points)** Compute the determinant of A. Show your work and justify your method of choice.

2. (a) **(2 points)** Let A be an $n \times n$ nonsingular lower triangular matrix; that is,

$$\begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

where $a_{ii} \neq 0$ for $1 \leq i \leq n$. Let $b = [b_1, b_2, \ldots, b_n]^T$ denote a (column) vector with n entries. A system of linear equations $Ax = b$ can be solved by "forward" substitution, which is very similar to back-substitution but starts with the first equation. Fill in the blanks in the following algorithm (use pseudocode, not MATLAB) so that it will solve such a system of linear equations.

$x_1 \leftarrow$ ____________________

for $i =$ ____________________

  sum $\leftarrow$ ____________________

  for $j =$ ____________________

    ____

  end for

  ____

end for

DELIVERABLES: Your complete pseudocode, it can be hand written or typed.

(b) **(2 points)** Give a floating-point operation (flop) count for this forward substitution algorithm. That is, determine the total number of floating-point additions, subtractions, multiplications and divisions (as a function of n) that this algorithm will execute.

DELIVERABLES: All the steps in your count, including your reasoning.

(c) **(2 points)** Convert your pseudocode to MATLAB code and write a function that takes as input the nonsingular lower triangular matrix A, and column vector b and returns the results of "forward" substitution. The input matrix should be the full matrix including the zeros. Show how your function can be called and what the result is for the following matrices A and b:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 \\ 4 & 5 & 6 & 0 \\ 7 & 8 & 9 & 10 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 5 \\ 15 \\ 34 \end{bmatrix}$$

DELIVERABLES: A copy of the m-file function, a diary of the steps taken to solve for the given A and b and the solution x.

3. Consider the function $f(x) = \cos(x)$ in the interval $[0, 8]$. You are given the following 3 points of this function:

1	0.5403
2	-0.4161
6	0.9602

   (a) **(2 points)** Calculate the quadratic Lagrange interpolating polynomial as the sum of the $L_0(x), L_1(x), L_2(x)$ polynomials we defined in class. The final answer should be in the form $P(x) = ax^2 + bx + c$, but with a, b, c known.

   DELIVERABLES: All your work in constructing the polynomial. This is to be done by hand not MATLAB.

   (b) **(2 points)** Plot $f(x) = \cos(x)$ in the interval $[0, 8]$ by creating vectors

```
x = 0:0.1:8;
y = cos(x);
```

   Use the MATLAB *hold on* command to retain the plot and plot the three data points provided above as stars. There is a similar example in the Lecture Notes for lecture 15. To plot the first star for example use

```
plot(1, 0.5403, '*')
```

   DELIVERABLES: The commands and the resulting plot from MATLAB.

   (c) **(2 points)** Plot each polynomial $L_0(x)f(x_0), L_1(x)f(x_1), L_2(x)f(x_2)$ for these specific values using the same axis as part (b):

```
x = 0:0.1:8
```

Note: Use `fplot` for this part.

DELIVERABLES: The commands and the resulting plot from MATLAB.

(d) **(2 points)** Make a new figure with $f(x)$ and the points as stars as well as the sum of $L_0(x)f(x_0), L_1(x)f(x_1), L_2(x)f(x_2)$ using the same

```
x = 0:0.1:8
```

but without showing $L_0(x), L_1(x), L_2(x)$.

DELIVERABLES: The commands and the resulting plot from MATLAB.