CS 475/675 Spring 2025: Crowdmark Assignment 1

Due May 16, at 11:59 pm Eastern.
The purpose of this assignment is to give you practice at coding functions in MATLAB.

The function names in this assignment all end in `slow`, to avoid conflict with the names of built-in functions in MATLAB.

Submit all components of your solutions (written/analytical work, code/scripts, figures, plots, output, etc.) to CrowdMark in PDF form in the appropriate section for each question.

You must also separately submit a single zip file containing any and all code/scripts you write to the Crowdmark Assignment 1 DropBox on LEARN, in runnable format (that is, .m).

You are NOT required to do any error checking on your function inputs - you may assume that each function input is as it is described in the question.

It may save you time if you use the provided solution templates.

For this assignment only, do NOT vectorize your computations - instead compute each entry in your desired output matrix on its own.

1. **(10 marks)** Create a MATLAB function, `negateslow.m`, that accepts an $m \times n$ matrix, A, as a parameter, and returns the negative of A. For example, if $A = \begin{bmatrix} 1 & 2 \\ -2 & 1 \\ 2 & -3 \end{bmatrix}$, then `negateslow(A)` should return $\begin{bmatrix} -1 & -2 \\ 2 & -1 \\ -2 & 3 \end{bmatrix}$.

   Submit to Crowdmark:

   - a copy of your `negateslow.m` code, and
   - the output from processing an explicitly displayed input matrix, A.

   Submit to the LEARN Dropbox:

   - `negateslow.m`.

2. **(10 marks)** Create a MATLAB function, `transposeslow.m`, that accepts an $m \times n$ matrix, A, as a parameter, and returns the transpose of A. For example, if $A = \begin{bmatrix} 1 & 2 \\ -2 & 1 \\ 2 & -3 \end{bmatrix}$, then `transposeslow(A)` should return $\begin{bmatrix} 1 & -2 & 2 \\ 2 & 1 & -3 \end{bmatrix}$.

   Submit to Crowdmark:

   - a copy of your `transposeslow.m` code, and
   - the output from processing an explicitly displayed input matrix, A.

   Submit to the LEARN Dropbox:

   - `transposeslow.m`.

3. **(10 marks)** Create a MATLAB function, `forwardsolveslow.m`, that accepts a lower triangular $n \times n$ matrix, L (with all its diagonal entries non-zero), and an $n \times 1$ column vector b as parameters, and returns the $n \times 1$ column vector y which is the solution of the system $Ly = b$. For example, if $L = \begin{bmatrix} 2 & 0 & 0 \\ -3 & 1 & 0 \\ 4 & -3 & 7 \end{bmatrix}$ and $b = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}$, then `forwardsolveslow(L,b)` should return $\begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}$.

   Submit to Crowdmark:

   - a copy of your `forwardsolveslow.m` code, and
   - the output from processing an explicitly displayed pair of input matrices, L and b.

   Submit to the LEARN Dropbox:

   - `forwardsolveslow.m`.

4. **(10 marks)** Create a MATLAB function, `backwardsolveslow.m`, that accepts an upper triangular $n \times n$ matrix, U (with all its diagonal entries non-zero), and an $n \times 1$ column vector y as parameters, and returns the $n \times 1$ column vector x which is the solution of the system $Ux = y$. For example, if $U = \begin{bmatrix} 2 & 1 & -1 \\ 0 & 1 & -2 \\ 0 & 0 & 3 \end{bmatrix}$ and $y = \begin{bmatrix} 1 \\ -5 \\ 6 \end{bmatrix}$, then `backwardsolveslow(U,y)` should return $\begin{bmatrix} 2 \\ -1 \\ 2 \end{bmatrix}$.

   Submit to Crowdmark:

   - a copy of your `backwardsolveslow.m` code, and
   - the output from processing an explicitly displayed pair of input matrices, U and y.

   Submit to the LEARN Dropbox:

   - `backwardsolveslow.m`.

5. **(10 marks)** Create a MATLAB function, `LUslow.m`, that accepts an $n \times n$ matrix, A (with all its leading principal submatrices non-singular) as a parameter, and returns the LU factorization of $A = LU$, in which U is upper triangular, and L is unit lower triangular. For example, if $A = \begin{bmatrix} 2 & 6 & 2 \\ -3 & -8 & 0 \\ 4 & 9 & 2 \end{bmatrix}$, then `LUslow(A)` should return $L = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{3}{2} & 1 & 0 \\ 2 & -3 & 1 \end{bmatrix}$ and $U = \begin{bmatrix} 2 & 6 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 7 \end{bmatrix}$.

   Submit to Crowdmark:

   - a copy of your `LUslow.m` code, and
   - the output from processing an explicitly displayed input matrix, A.

   Submit to the LEARN Dropbox:

   - `LUslow.m`.