HW4: MATH/CSCI-4800-02 Numerical Computing

Due by 3pm on March 1 (Friday)
The bonus problem #6 is due by 2pm on March 14

1. Find the LU factorization of A below in (1). Based on the L and U matrices you obtain, using back and forward substitutions to solve $Ax = b$ for x, with b also given in (1). No pivoting is needed. All the calculation you report should be by hand.

   Remark: Feel free to verify your L and U by matrix multiplication, and to verify your x by matrix-vector multiplication, either by hand or by computer. Verification step is not required for this homework problem.

$$A = \begin{bmatrix} 1 & -1 & 1 & 2 \\ 0 & 2 & 1 & 0 \\ 1 & 3 & 4 & 4 \\ 0 & 2 & 1 & -1 \end{bmatrix}, \qquad b = \begin{bmatrix} 3 \\ 2 \\ 11 \\ 1 \end{bmatrix}. \tag{1}$$

2. Given $B = (b_{ij}) \in \mathbb{R}^{4\times 4}$, how to use matrix multiplication to accomplish the following actions:

   - to exchange row 1 and row 4,and ALSO to turn row 2 to its negative; *your answer should be one single matrix, not two matrices*
   - to get the following matrix C.

$$C = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ 5b_{31} & 5b_{32} & 5b_{33} & 5b_{34} \\ b_{41} - 2b_{11} & b_{42} - 2b_{12} & b_{43} - 2b_{13} & b_{44} - 2b_{14} \\ b_{11} & b_{12} & b_{13} & b_{14} \end{bmatrix}. \tag{2}$$

3. Given

$$A = \begin{bmatrix} 1 & 2 \\ 1+\delta & 2 \end{bmatrix}.$$

   Consider three values of δ: $\delta = 10^{-2}, 10^{-6}, 10^{-8}$, and for each of them, answer the following questions.

   - Compute, by hand, the ∞-norm condition number of A, denoted as $\text{cond}(A, \infty) = ||A||_\infty ||A^{-1}||_\infty$.
   - Let $x = [7, 3]^T$, and set $b = Ax$. Using MATLAB backslash command to solve $Ax = b$ to get an approximate solution x_c. Report

$$\text{Error}_\sigma = \frac{||x - x_c||_\infty}{||x||_\infty},$$

     and discuss how the magnitude of Error_σ is related to the condition number $\text{cond}(A, \infty)$ of the respective matrix A, as well as the machine epsilon (denoted as ϵ_{mach} in class, or denoted as eps in MATLAB). *To compute Error_σ, feel free to use MATLAB built-in command, norm, check "help norm" in MATLAB.*

4. Tell whether each of the following matrices are symmetric positive definite (SPD). Why?

$$A_1 = \begin{bmatrix} 1 & 3 \\ 3 & 10 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & -1 \\ -1 & 0 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 2 & 4 \\ 4 & 3 \end{bmatrix}.$$

5. For the matrix A given below, show $x^T Ax > 0$ for any nonzero $x \in \mathbb{R}^4$ (*implying A is SPD*). Find, by hand, its Cholesky factorization.

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}.$$

Remark: matrices of such structure arise in numerical simulations of PDEs modeling heat conduction, wave propagation etc.

6. (Bonus Problem) Implement Jacobi and /or Gauss-Seidel method(s). Apply your codes to the following problem $Ax = b$, where

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 2 & 4 & 1 \\ -1 & 2 & 5 \end{bmatrix}, \qquad b = \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix}.$$

Propose a stoping criterion (*for instance, to ensure the relative or absolute error is bounded by some prescribed error level*). Plot the error history $E_j = ||x^{(j)} - x||/||x||$, where $x^{(j)}$ is the j-th iterate, and $x = [2, -1, 1]^T$ is the exact solution, $|| \cdot ||$ is a norm of your choice (refer to MATLAB command, *norm, plot, semilogy*). If you implement both Jacobi and Gauss-Seidel methods, you can plot the error history of these two methods in the same figure so you can easily compare the performance. Add suitable title, legend, etc to your plots. Summarize and discuss (or even explain) your observations.

IF you have confidence that your codes are free of bugs (*based on their performance on the 3×3 example above, or possibly on more tests you have done. This is a common practice to validate computer codes, that is, by applying them to simple test cases to gain confidence first, before applying them to more challenging problems.*), you can test your codes on a much larger problem, say a problem of 100×100 size. For instance, you can generate a strictly diagonally dominant matrix $A \in \mathbb{R}^{n \times n}$ (*with $n = 100$ or a value of your choice. This will need some thoughts. Here are some MATLAB commands that may be helpful, "diag", "rand". Feel free to write a small code to verify that your A is strictly diagonally dominant*). Pick an exact solution $x \in \mathbb{R}^n$ and set $Ax = b$. Apply your codes to this much larger matrix A and b, and plot and study the error history of the methods. Summarize and discuss (or even explain) your observations.