Assignment 1

General Instructions

- Solutions due by 11^{th} February 2025 for Group A (Prof. Pallab's class); 13^{th} February 2025 for Group B (Prof. Sreehari's class)
- Hand calculations can be done as homework and should be submitted together with the report.
- Write your name and roll-number on your report

Hand calculations

1. (a) Consider solving $Ax = b$, with A and b complex and order $(A) = n$. Convert this problem to that of solving a real square system of order $2n$. *Hint:* Write $A = A_1 + iA_2$, $b = b_1 + ib_2$, $x = x_1 + ix_2$, with $A_1, A_2, b_1, b_2, x_1, x_2$ all real. Determine the number of operations required for solving the complex system $Ax = b$ using real arithmetic and as a real square sytem of order $2n$.

   (b) Compare these number of operations with those based on directly solving $Ax = b$ using Gaussian elimination and complex arithmetic. Note: One multiplication in complex arithmetic is equivalent of four multiplications in real arithmetic.

   (c) Reduce the real square system of order $2n$ using Gaussian elimination on the blocks of sub-matrices and express the unknown vectors in the following format: $[\ldots]\{x_1\} = \{\ldots\}$ and $[\ldots]\{x_2\} = \{\ldots\}$. (15 points)

2. For a linear system $Ax = b$, with $A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$, it is clear that the first and the second rows must be swapped before applying Gaussian elimination. Computers are finite precision machines and rounding-off errors can cause failure of Gaussian elimination. This general problem is better understood by applying Gaussian elimination for a slightly perturbed version of A - $\begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix}$. Solve this system exactly with $b = [1, 0]^T$. Compare the solution with that obtained by using floating point arithmetic with a precision of 10^{-16}. (15 points)

3. We learnt two ways to get $A = LU$ in class:

   (a) From Gaussian Elimination and the multiplication factors

   (b) Using Doolittle's algorithm.

   Consider the matrix:

$$\begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{pmatrix} \tag{1}$$

   Get L and U matrices both ways keeping $l_{ii} = 1$ for $i = \{1, 2, 3, 4\}$. Are they the same? (10 points)

Programming

1. Write a program that performs Gaussian elimination for a square system of size $n \times n$. Consider writing it as a modular program with separate functions or subroutines that perform forward-elimination and backward-substitution. To test the program, solve the system $Ax = b$ of order n, with $A = [a_{ij}]$ defined by

$$a_{ij} = \max(i, j).$$

   Also define $b = [1, 1, ..., 1]^T$. Solve the system to obtain the solution vector x, for $n = 32, 128, 512, 1024$. Plot a graph between n and $1/\sum_{i=1}^{n} {x_i}^2$. (30 points)

2. Algorithm for Gaussian elimination (which you would have used for the previous problem) is given below -

   $U = A, \quad L = I$

   for $k = 1$ to $m - 1$

   for $j = k + 1$ to m

   $l_{jk} = u_{jk}/u_{kk}$

   $u_{j,k:m} = u_{j,k:m} - l_{jk} u_{k,k:m}$

   This algorithm has triply nested loops; two explicit *for* loops, and the third loop is implicit in the vectors $u_{j,k:m}$ and $u_{k,k:m}$. Rewrite this algorithm with just one explicit *for* loop indexed by k. Inside this loop, U will be updated at each step by a certain rank-one outer product. This *outer product* form of Gaussian elimination may be a better starting point than the origninal algorithm if one wants to optimize computer performance. Write a Gaussian elimination program using this algorithm and check if it indeed has benefits over the original program (30 points).