MA398 Matrix Analysis and Algorithms: Exercise Sheet 1

1. (a) Given the matrix
$$A = \begin{bmatrix} 1 & -1 & 1 \\ 2 & 3 & -1 \\ 3 & -2 & -9 \end{bmatrix}, \quad b = \begin{bmatrix} 8 \\ 2 \\ 9 \end{bmatrix}$$
Use the Gaussian elimination method described in the lecture notes to solve the system $Ax = b$. Show each step of your work.

   (b) Implement a Python function called **gaussian_elimination(A, b)** that takes as input a numpy array A and a vector b and outputs the solution vector x. Test your function on the matrix and vector given in part (a).

2. (Forward substitution) Formulate the algorithm **FS** (forward substitution) to solve the system $Lx = b$ where $L \in \mathbb{C}^{n\times n}$ is unit lower triangular and $b \in \mathbb{C}^n$, and show that the algorithm computes the correct result, with a computational cost of n^2.

3. (LU decomposition)

   (a) Find the LU decomposition of the matrix
$$A = \begin{bmatrix} 2 & -1 & 3 \\ 4 & 2 & 1 \\ -6 & -1 & 2 \end{bmatrix},$$
and use it to solve $Ax = b$ with $b = (7, 8, -3)$.

   (b) Implement a Python function called **lu_factorization(A)** that takes as input a numpy array A and outputs the lower triangular matrix L and the upper triangular matrix U. Test your function on the matrix given in part (a).

   (c) Show that the multiplication of your L and U matrices gives back the original matrix A.

4. (Operation count) How many operations (divisions and multiplications) are necessary to perform an LU decomposition without pivoting?

5. (Diagonal dominance) A matrix $A = (a_{ij})_{i,j=1}^n \in \mathbb{C}^{n\times n}$ is called <u>strictly diagonal dominant</u> if
$$|a_{ii}| > \sum_{j\neq i} |a_{ij}| \quad \text{for all } i \in \{1, \ldots, n\}.$$
Show that such a matrix is invertible and its LU factorisation exists.
For this purpose, show that the remaining matrix $(u_{ij}^{(k)})_{i,j=k+1}^n$ after step k of the Gaussian elimination without pivoting still is strictly diagonal dominant.

6. (a) Let $A = (a_{ij})_{i,j=1}^n \in \mathbb{C}^{n\times n}$ be a matrix of bandwidth $w \in \{0, \ldots, n-1\}$, i.e.,
$$a_{ij} = 0 \quad \text{if } |j - i| > w.$$
Give an example of a 4×4 matrix of bandwidth $w = 2$ but not $w = 1$ which fulfils the strong row sum criterion (also known as strict diagonal dominance).

   (b) Assume that the LU factorisation of a matrix $A \in \mathbb{C}^{n\times n}$ of bandwidth $w = 1$ can be computed with the algorithm LU (without pivoting!). Show that then the computed matrices L and U are of bandwidth $w = 1$, too.

   (c) Formulate a specialised version of the algorithm LU for band matrices of bandwidth $w = 1$ where only the necessary operations are carried out. Ensure and check that the number of elementary executable operations is $O(n)$ as $n \to \infty$.