Lab-7: Solving Linear System

1. Prerequisites:

- **a.** Open the colab file shared in BUX.
- **b.** Create a copy of that shared file.
- **c.** Rename the colab filename using the format Name_ID_Lab Section

2. Lab Tasks:

Task-1: Solving a linear system using an inverse matrix

We have a linear system

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
\cdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
$$

It is convenient to express this system in the matrix form

$$Ax = b$$

where A is an $n \times n$ square matrix with elements a_{ij}, and x, b are $n \times 1$ vectors.

We have to keep in mind that this system will have a unique solution iff A is non-singular, given by $x = A^{-1}b$.

A. You have to **implement** the `get_result_by_inverse_matrix(A, b)`, where *A* is a *n* x *n* matrix and *b* is a *n* x *1* vector.

B. **Check** if *A* is a singular matrix or not. If not, find its inverse. **[3]**

C. **Multiply** the inverse with the vector *b*. **[2]**

Task-2: Gaussian elimination method

Gaussian elimination method uses elementary row operations to transform the system to an upper triangular form $Ux = y$.

Elementary row operations include swapping rows and adding multiples of one row to another. They won't change the solution x but will change the matrix A and the right-hand side b.

The upper triangular matrix, U, is defined as

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & u_{nn} \end{bmatrix}$$

The algorithm of the Gaussian elimination method:

Algorithm of Gaussian elimination

Let $A^{(1)} = A$ and $b^{(1)} = b$. Then for each k from 1 to $n-1$, compute a new matrix $A^{(k+1)}$ and right-hand side $b^{(k+1)}$ by the following procedure:

1. Define the row multipliers

$$m_{ij} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, i = k+1, \cdots, n.$$

2. Use these to remove the unknown x_k from equations $k+1$ to n, leaving

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}, b_i^{(k+1)} = b_i^{(k)} - m_{ik}b_k^{(k)}, i, j = k+1, \cdots, n.$$

It is helpful to combine these matrices to form an Augmented matrix (matrix b is the fourth column). We will perform this row operations on the Augmented matrix. It takes care of both A and b matrixes at the same time.

After generating the upper triangular matrix, we have to apply **backward substitution method**. For any $n \times n$ upper triangular system, $Ux = b$, the solution is:

$$x_j = \frac{b_j - \sum_{k=j+1}^{n} u_{jk}x_k}{u_{jj}}, j = n, n-1, \cdots, 1.$$

Here we assumed that $detU \neq 0$.

A. You have to **implement** the `get_result_gaussian_elimination(n, A)` method, where n is the number of unknowns and A is the augmented matrix. **[2.5 + 2.5 = 5]**

Task-3: LU Decomposition

We will transform the $n \times n$ matrix A into a product of two triangular matrices: one lower triangular (L) and the other upper triangular (U).

$$A = LU$$

The algorithm of LU decomposition:

Algorithm of LU decomposition:

1. Initialize L to an identity matrix, I of dimension $n \times n$ and $U = A$.
2. For $i = 1, \ldots, n$ do Step 3
3. For $j = i + 1, \ldots, n$ do Steps 4-5
4. Set $l_{ji} = u_{ji} / u_{ii}$
5. Perform $U_j = (U_j - l_{ji} U_i)$ (where U_i, U_j represent the i and j rows of the matrix U, respectively)

- We know, The linear system in matrix form is $Ax = b$.
- Using the decomposition, we get $LUx = b$.
- Now, let $Ux = y \implies Ly = b$.
- Since L is lower triangular, we solve $Ly = b$ to obtain y by forward substitution.
- Since U and y are known, we solve $Ux = y$ to obtain x by backward substitution.

A. You have to **implement** the `lu(A)` method.

B. You have to **implement** the `forward_substitution(L, b)` method. **[5]**

C. You have to **implement** the `back_substitution(U, y)` method. **[5]**

Note: `lu_solve(A, b)` method has been completed for you.

Total Marks: 20