1 Experiment No. 7

2 Experiment Title

Solving system of linear equations and matrix inversion using LU Decomposition Method

3 Objective

The objectives of this lab are:

- To gather knowledge about solving a system of linear equations using the LU Decomposition Method.
- To implement and show output using the LU Decomposition Method for solving linear equation and inverse matrix.

4 Theory

The LU factorization method is a technique used to solve systems of linear algebraic equations of the form:

$$AX = B \tag{7.1}$$

Although it is a valid and sound method, solving the system repeatedly with the same coefficient matrix A but with different right-hand-side vectors B can be inefficient. LU decomposition addresses this by separating the time-consuming elimination of matrix A from the manipulation of the vector B. Thus, once A is decomposed, multiple right-hand-side vectors can be processed efficiently.
The LU decomposition splits A into a lower triangular matrix L and an upper triangular matrix U:

$$A = LU \tag{7.2}$$

Substituting into Eq. (7.1), we get:

$$LUX = B \tag{7.3}$$

Let:

$$UX = D \tag{7.4}$$

Then,

$$LD = B \tag{7.5}$$

Thus, the LU factorization method consists of two main steps:

1. **LU Decomposition Step:** The coefficient matrix A is decomposed into a lower triangular matrix L and an upper triangular matrix U.
2. **Substitution Step:**
   (a) Forward substitution is applied to Eq. (7.5) to solve for the intermediate vector D.
   (b) Back substitution is then used on Eq. (7.4) to solve for X.

4.1 LU Decomposition

Gauss elimination can be used to decompose A into L and U. Let:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

The forward elimination process reduces A to:

$$U = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix}$$

Meanwhile, the multipliers used in elimination form the matrix L:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix}$$

where:

$$L_{21} = \frac{a_{21}}{a_{11}}, \quad L_{31} = \frac{a_{31}}{a_{11}}, \quad L_{32} = \frac{a'_{32}}{a'_{22}}$$

4.2 Finding X by Substitution

After obtaining L and U, we solve $LD = B$ using forward substitution:

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \Rightarrow \begin{cases} d_1 = b_1 \\ d_2 = b_2 - l_{21}d_1 \\ d_3 = b_3 - l_{31}d_1 - l_{32}d_2 \end{cases}$$

Then we solve $UX = D$ using back substitution:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \Rightarrow \begin{cases} x_3 = \frac{d_3}{u_{33}} \\ x_2 = \frac{d_2 - u_{23}x_3}{u_{22}} \\ x_1 = \frac{d_1 - u_{12}x_2 - u_{13}x_3}{u_{11}} \end{cases}$$

4.3 Finding Inverse of a Matrix

The inverse A^{-1} satisfies:

$$AA^{-1} = I \tag{7.6}$$

If $AX = B$ and $B = I$, then $X = A^{-1}$. Using $A = LU$ and substituting:

$$AX = LUX = LD = B$$

For a 3×3 matrix:

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [B_1 \; B_2 \; B_3]$$

Steps to find A^{-1}:

1. For each column B_i of B, solve $LD_i = B_i$ by forward substitution.

2. Then solve $UX_i = D_i$ by back substitution.

3. Repeat for $i = 1, 2, 3$, and form the inverse matrix:

$$A^{-1} = [X_1 \ X_2 \ X_3]$$

For example, to find D_1:

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_{11} \\ d_{21} \\ d_{31} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \end{bmatrix} \Rightarrow \begin{cases} d_{11} = b_{11} \\ d_{21} = b_{21} - l_{21}d_{11} \\ d_{31} = b_{31} - l_{31}d_{11} - l_{32}d_{21} \end{cases}$$

Then solve:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} = \begin{bmatrix} d_{11} \\ d_{21} \\ d_{31} \end{bmatrix} \Rightarrow \begin{cases} x_{31} = \frac{d_{31}}{u_{33}} \\ x_{21} = \frac{d_{21} - u_{23}x_{31}}{u_{22}} \\ x_{11} = \frac{d_{11} - u_{12}x_{21} - u_{13}x_{31}}{u_{11}} \end{cases}$$

Repeat this process for all columns to obtain A^{-1}.

5 Algorithm

5.1 LU Decomposition

Input: Square matrix A of size $n \times n$
Output: Lower triangular matrix L and upper triangular matrix U

1. Initialize L as an identity matrix of size $n \times n$

2. For $k = 1$ to $n - 1$:

   (a) For $i = k + 1$ to n:
      i. $L[i, k] \leftarrow A[i, k]/A[k, k]$
      ii. $A[i, :] \leftarrow A[i, :] - (A[i, k]/A[k, k]) \cdot A[k, :]$

3. Set $U \leftarrow A$

4. Return L and U

5.2 Solving System Using LU Decomposition

Input: Matrices L, U, and B ($n \times 1$)
Output: Solution vector X
Step 1: Forward Substitution (Find D)

1. Initialize D and D_{sum} as zero vectors

2. For $j = 1$ to n:

   (a) $D_{\text{sum}}[j] \leftarrow \sum_{k=1}^{j-1} D[k] \cdot L[j, k]$
   (b) $D[j] \leftarrow B[j] - D_{\text{sum}}[j]$

Step 2: Backward Substitution (Find X)

1. Initialize X and sum as zero vectors
2. For $j = n$ down to 1:
   (a) $\text{sum}[j] \leftarrow \sum_{k=j+1}^{n} U[j,k] \cdot X[k]$
   (b) $X[j] \leftarrow (D[j] - \text{sum}[j])/U[j,j]$

5.3 Matrix Inversion Using LU Decomposition

Input: Matrices L, U, and identity matrix B ($n \times n$)
Output: Inverse matrix A^{-1}
Step 1: Forward Substitution (Find D)

1. Initialize D and D_{sum} as zero matrices
2. For $i = 1$ to n:
   (a) For $j = 1$ to n:
      i. $D_{\text{sum}}[j,i] \leftarrow \sum_{k=1}^{j-1} D[k,i] \cdot L[j,k]$
      ii. $D[j,i] \leftarrow B[j,i] - D_{\text{sum}}[j,i]$

Step 2: Backward Substitution (Find A^{-1})

1. Initialize A^{-1} and sum as zero matrices
2. For $i = 1$ to n:
   (a) For $j = n$ down to 1:
      i. $\text{sum}[j,i] \leftarrow \sum_{k=j+1}^{n} U[j,k] \cdot A^{-1}[k,i]$
      ii. $A^{-1}[j,i] \leftarrow (D[j,i] - \text{sum}[j,i])/U[j,j]$

5.4 Verification

Input: L, U, A^{-1}, original matrix A
Steps:

1. Compute $L_d = L -$ MATLAB's L
2. Compute $U_d = U -$ MATLAB's U
3. Compute $A_d = A^{-1} -$ MATLAB's inv(A)
4. Verify if L_d, U_d, and A_d are zero matrices

5.5 Solving Non-linear Equation Using LU Decomposition Method

5.5.1 MATLAB Code:

```
clc;
clear;
close all;

A = [4, -2, 1; 20, -7, 12; -8, 13, 17];
B = [11; 70; 17];
n = size(A,1);
L = zeros(n);
U = A;

for k = 1:n-1
L(k,k) = 1;
for i = k+1:n
L(i,k) = U(i,k) / U(k,k);
U(i,:) = U(i,:) - L(i,k) * U(k,:);
end
end
L(n,n) = 1;
LU = L * U;
disp('Lower Triangular Matrix L:');
disp(L);
disp('Upper Triangular Matrix U:');
disp(U);
disp('Display the L*U')
disp(LU)

%for finding D
D = zeros(n,1);
for j = 1:n
Dsum = 0;
for k = 1:j-1
Dsum = Dsum + D(k) * L(j,k);
end
D(j) = B(j) - Dsum;
end

%for finding X
X = zeros(n,1);
for j = n:-1:1
sum = 0;
for k = j+1:n
sum = sum + U(j,k) * X(k);
end
X(j) = (D(j) - sum ) / U(j,j);
end
disp('Solution Vector X:');
disp(X);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = size(L,1);
I = eye(n);        % Identity matrix
A_inv = zeros(n); % To store the inverse
```

Listing 1: Solving Non-linear Equation Using LU Decomposition in MATLAB.

```
        %  to solve LY = I
        Y = zeros(n,n);
        for col = 1:n
        for j = 1:n
        sum1 = 0;
        for k = 1:j-1
        sum1 = sum1 + L(j,k) * Y(k,col);
        end
        Y(j,col) = (I(j,col) - sum1) / L(j,j);
        end
        end
        %  to solve UX = Y
        for col = 1:n
        for j = n:-1:1
        sum2 = 0;
        for k = j+1:n
        sum2 = sum2 + U(j,k) * A_inv(k,col);
        end
        A_inv(j,col) = (Y(j,col) - sum2) / U(j,j);
        end
        end
        disp('Inverse Matrix of A:');
        disp(A_inv);
        disp('Verification:');
        m=A*A_inv;
        disp(m);
```

Listing 2: Solving Non-linear Equation Using LU Decomposition in MATLAB.

5.6 Result Shown in Command Window

```
        Lower Triangular Matrix L:
        1      0      0
        5      1      0
        -2      3      1
        Upper Triangular Matrix U:
        4     -2      1
        0      3      7
        0      0     -2

        Display the L*U
        4     -2      1
        20     -7     12
        -8     13     17
        Solution Vector X:
        1
        -2
        3
        Inverse Matrix of A:
        11.4583    -1.9583     0.7083
        18.1667    -3.1667     1.1667
        -8.5000     1.5000    -0.5000
        Verification:
         1.0000          0          0
        0.0000     1.0000          0
        0          0     1.0000
```

Listing 3: Command Window for LU Decomposition

6 Discussion

In this experiment, the LU Decomposition method was used to solve systems of linear equations inverse of square matrix. That method was implemented in MATLAB and tested on standard systems. Initially input A and B were assumed, and Gauss eliminations method was used to decompose matrix A to lower and upper triangular matrix. Then Gauss eliminations was used to solve the linear equation and to find inverse of the matrix.
Due to lower and upper triangular matrices, it was easier to solve linear equation as well as finding inverse matrix. The results from the method was verified with MATLAB's built-in solver that found to be accurate.