LU

DECOMPOSITION

PROJECT

CREATED BY :-

RUTVIK MARAKANA

INTRODUCTION TO LU DECOMPOSTION

- ## *IMPORTANT DEFINITIONS*

  1. **Decomposition** of a matrix A is an equation that expresses A as a product of two or more matrices [3].
  2. **Triangular** matrix is a special kind of square matrix. A square matrix is **lower triangular** if all entries above the main diagonal are 0. A square matrix is **upper triangular** if all entries below the main diagonal are 0.
  3. **Echelon form of a matrix**

     A matrix is in echelon form when **1)** All nonzero rows are above any rows of all zeros. **2)** Each leading entry of a row is in a column to the right of the leading entry of the row above it. **3)** All entries in a column below a leading entry are zeros [4].

- **LU Decomposition** of a matrix A is the decomposition of A into a lower triangular matrix(L) and Echelon form of A(U). LU Decomposition is denoted as **A=LU**. It was introduced by a Polish mathematician Tadeusz Banachiewicz in 1938 [2].

- **ASSUMPTIONS IN LU DECOMPOSTION**

  1. A is a m x n matrix [3].
  2. A can be reduced to echelon form without row interchanges [3].
  3. L has 1's on the main diagonal [3].

- **STEPS FOR SOLVING A LINEAR EQUATION**

  Assume a linear equation given in matrix form, Ax=b. We want to solve the equation for x, given A and b.

  The steps to do so are as follows :-

  1. Decompose A into L and U such that A=LU. The resultant now is LUx=b [3].
  2. Then we transform the equation to Ly = b and solve for y [3].
  3. Lastly, we solve the equation Ux = y for x [3].

ALGORITHM OF LU DECOMPOSITION

- First write A as product of L and U i.e., A=LU.

  Here, L is the lower triangular matrix with 1's on diagonal. The entries in L below the main diagonal are denoted as $l_{\text{(row number) (column number)}}$.

  U is the echelon form of A. The entries in U that are non-zero are denoted as $u_{\text{(row number) (column number)}}$.
- Now multiply L and U and get the resultant matrix.
- Equate each element of the resultant matrix with the corresponding element in matrix A.
- From the above step, one will get all the unknown elements of L and U and so the LU Decomposition of A.

(2) Given,

$$A = \begin{bmatrix} -3 & 1 & 2 \\ 6 & 2 & -5 \\ 9 & 5 & -6 \end{bmatrix}$$

By the definition of LU Decomposition,

$$A = LU$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

$$= \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21}u_{11} & l_{21}u_{12}+u_{22} & l_{21}u_{13}+u_{23} \\ l_{31}u_{11} & l_{31}u_{12}+l_{32}u_{22} & l_{31}u_{13}+l_{32}u_{23}+u_{33} \end{bmatrix}$$

Now by comparing corresponding values of A and LU,

$u_{11} = -3$, $u_{12} = 1$ and $u_{13} = 2$.

Now, $l_{21}u_{11} = -3l_{21} = 6 \Rightarrow l_{21} = -2$

$l_{21}u_{12} + u_{22} = (-2)(1) + u_{22} = 2 \Rightarrow u_{22} = 4$

$l_{21}u_{13} + u_{23} = (-2)(2) + u_{23} = -5 \Rightarrow u_{23} = -1$

$l_{31}u_{11} = -3l_{31} = 9 \Rightarrow l_{31} = -3$

$l_{31}u_{12} + l_{32}u_{22} = (-3)(1) + 4l_{32} = 5 \Rightarrow l_{32} = 2$

$l_{31}u_{13} + l_{32}u_{23} + u_{33} = (-3)(2) + (2)(-1) + u_{33} = -6 \Rightarrow u_{33} = 2$

So,
$$L = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 2 & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} -3 & 1 & 2 \\ 0 & 4 & -1 \\ 0 & 0 & 2 \end{bmatrix}$$

Now,
$A\vec{x} = \vec{b}$ where $\vec{b} = (0, 3, 8)$

$LU\vec{x} = \vec{b}$

Let $U\vec{x} = \vec{y}$

So, $L\vec{y} = \vec{b}$ where $\vec{y} = (y_1, y_2, y_3)$

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 8 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ -2y_1 + y_2 \\ -3y_1 + 2y_2 + y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 8 \end{bmatrix}$$

So, $y_1 = 0$

$-2y_1 + y_2 = (-2)(0) + y_2 = 3 \Rightarrow y_2 = 3$

$-3y_1 + 2y_2 + y_3 = (-3)(0) + (2)(3) + y_3 = 8 \Rightarrow y_3 = 2$

So, $\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 2 \end{bmatrix}$

Now solve $U\vec{x} = \vec{y}$ where $\vec{x} = (x_1, x_2, x_3)$

So,

$$\begin{bmatrix} -3 & 1 & 2 \\ 0 & 4 & -1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} -3x_1 + x_2 + 2x_3 \\ 4x_2 - x_3 \\ 2x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 2 \end{bmatrix}$$

By comparing values of $U\vec{x}$ with $\vec{b}$, we get

$2x_3 = 2 \Rightarrow x_3 = 1$

$4x_2 - x_3 = 4x_2 - 1 = 3 \Rightarrow x_2 = 1$

$-3x_1 + x_2 + 2x_3 = -3x_1 + 1 + 2(1) = 0 \Rightarrow x_1 = 1$

$$\text{So, } \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Check if $\vec{x} = (1, 1, 1)$ correct.

$$\text{So, } A\vec{x} = \begin{bmatrix} -3 & 1 & 2 \\ 6 & 2 & -5 \\ 9 & 5 & -6 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -3 + 1 + 2 \\ 6 + 2 - 5 \\ 9 + 5 - 6 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 8 \end{bmatrix} = \vec{b}$$

(3) Given,

$$A = \begin{bmatrix} 2 & 1 & -1 \\ -4 & -2 & 5 \\ 6 & 2 & 11 \end{bmatrix}$$

To find the ref(A),

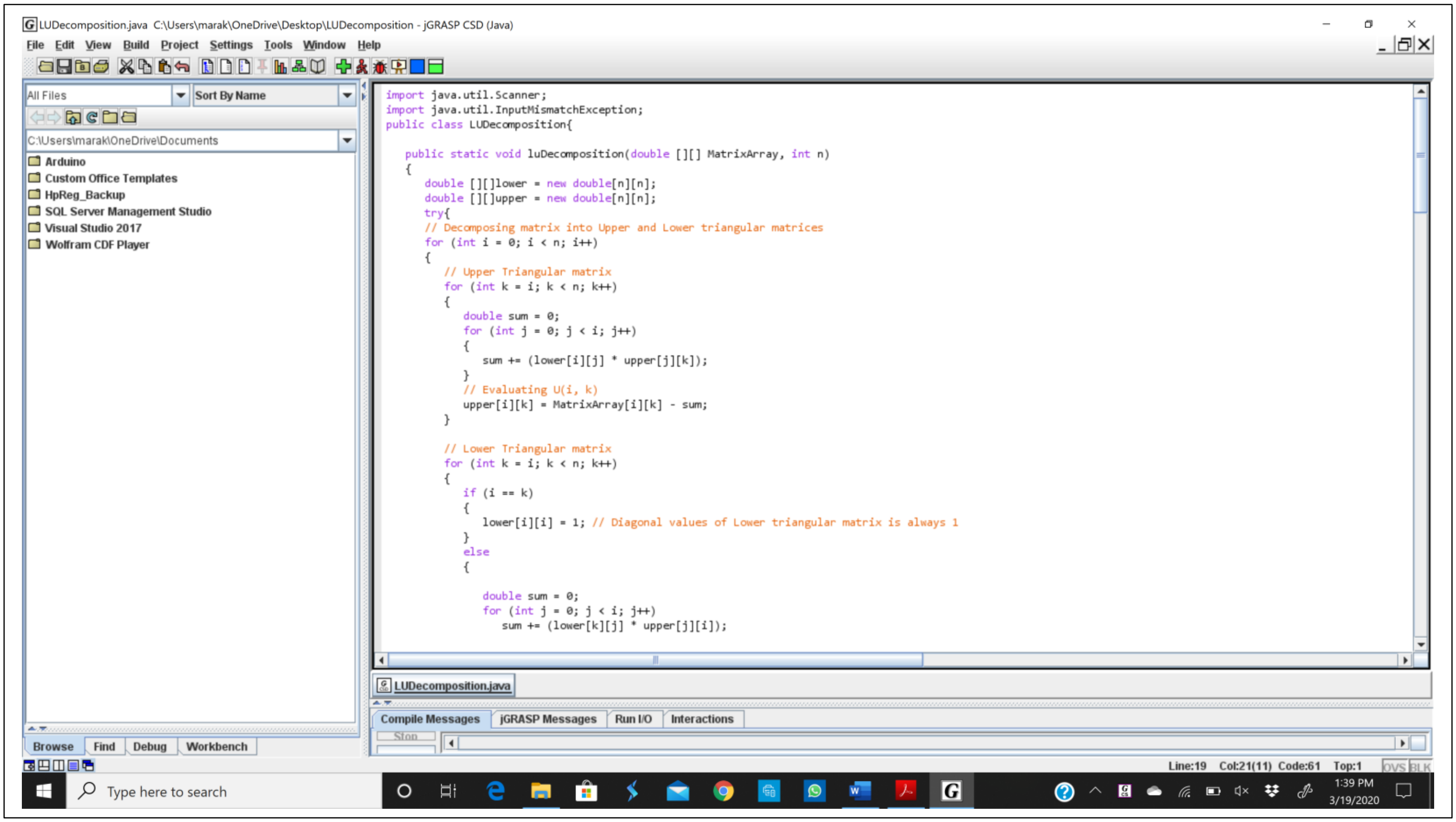
Do $R_2 + 2R_1 \to R_2$, $R_3 - 3R_1 \to R_3$

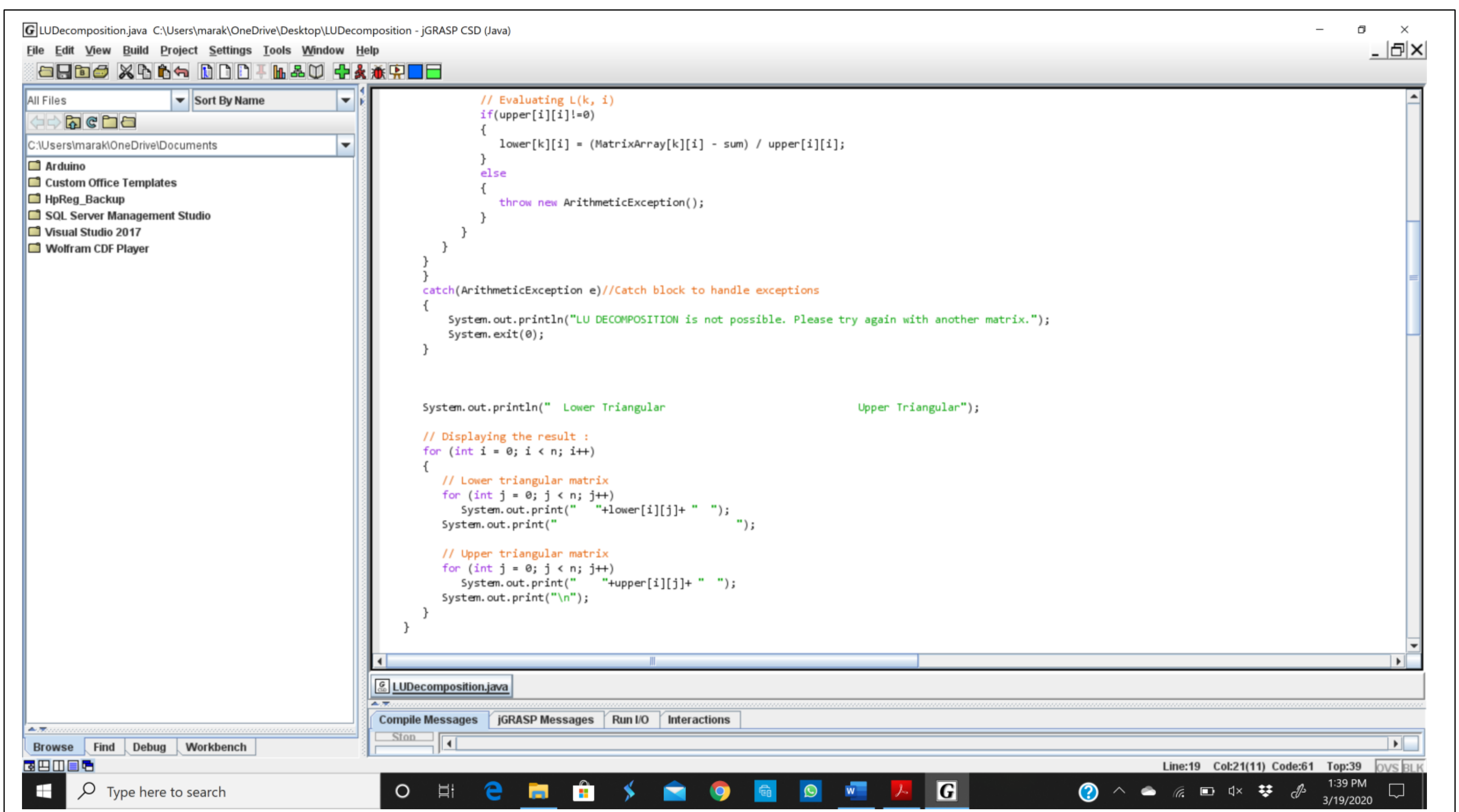
$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 0 & 3 \\ 0 & -1 & 14 \end{bmatrix}$$

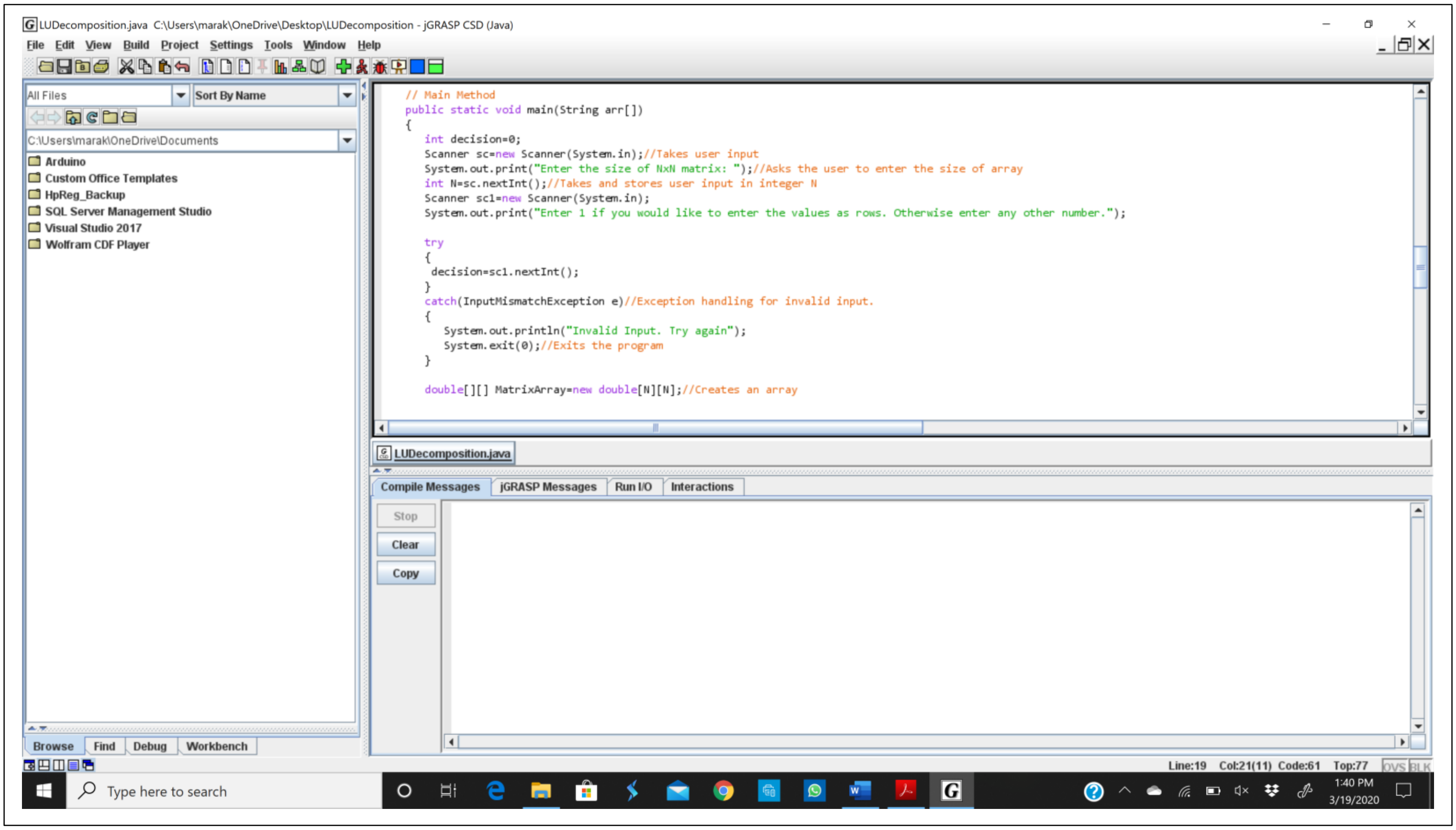
Since the first non-zero entry in row 2 is in column 3, we definitely need a row swapping to find the ref(A).

As a result, LU Decomposition of A is not possible.

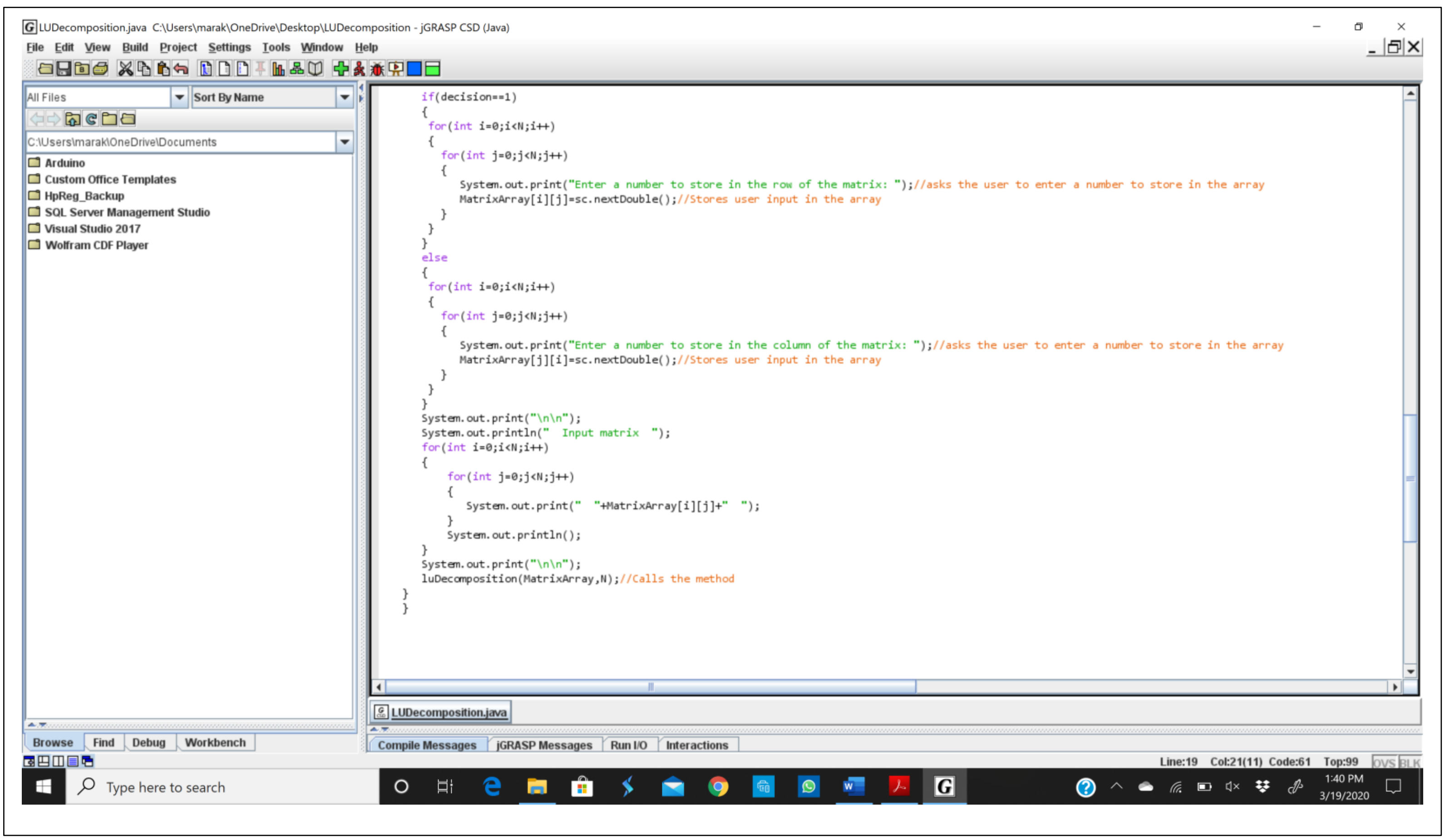
LU DECOMPOSITION JAVA PROGRAM SCREENSHOTS





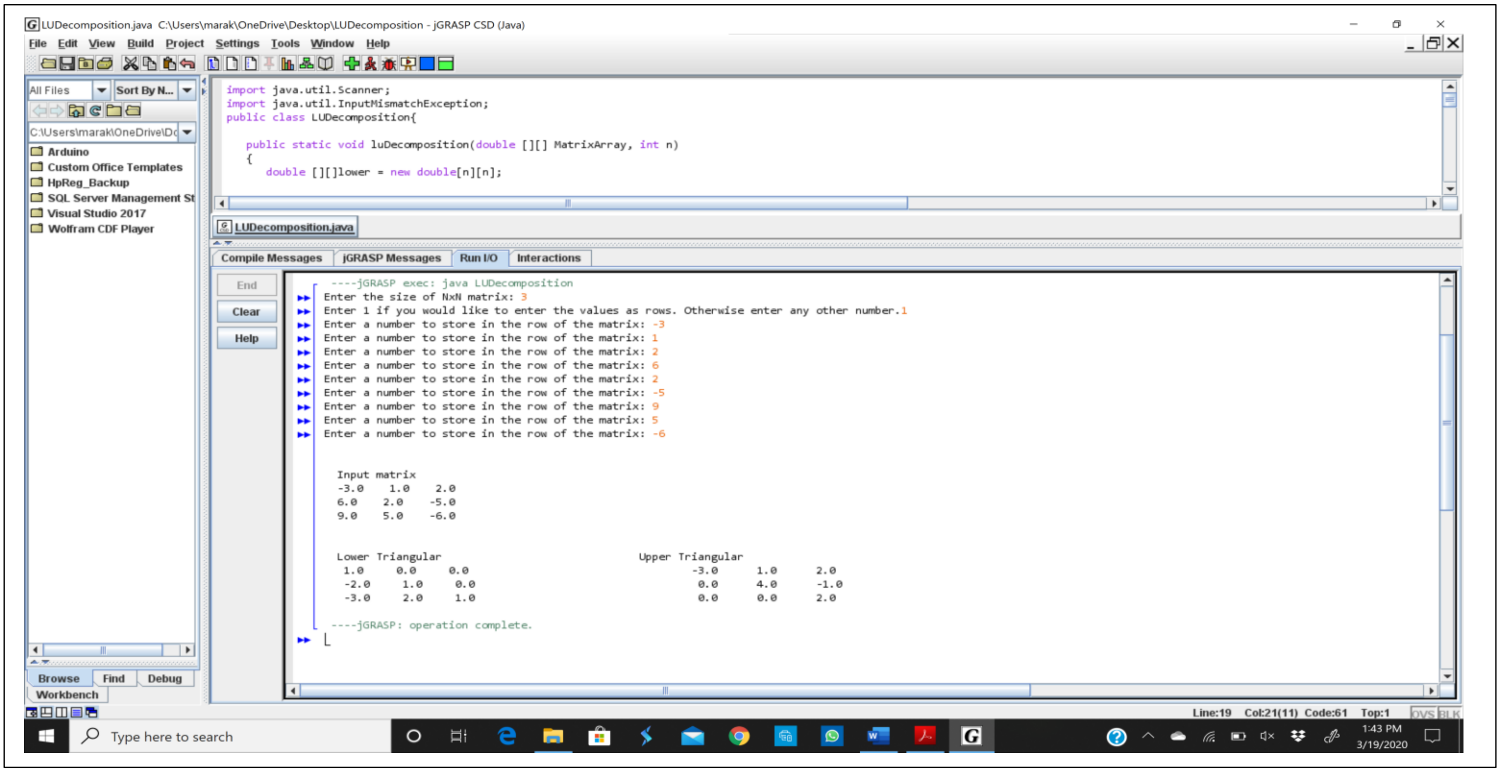


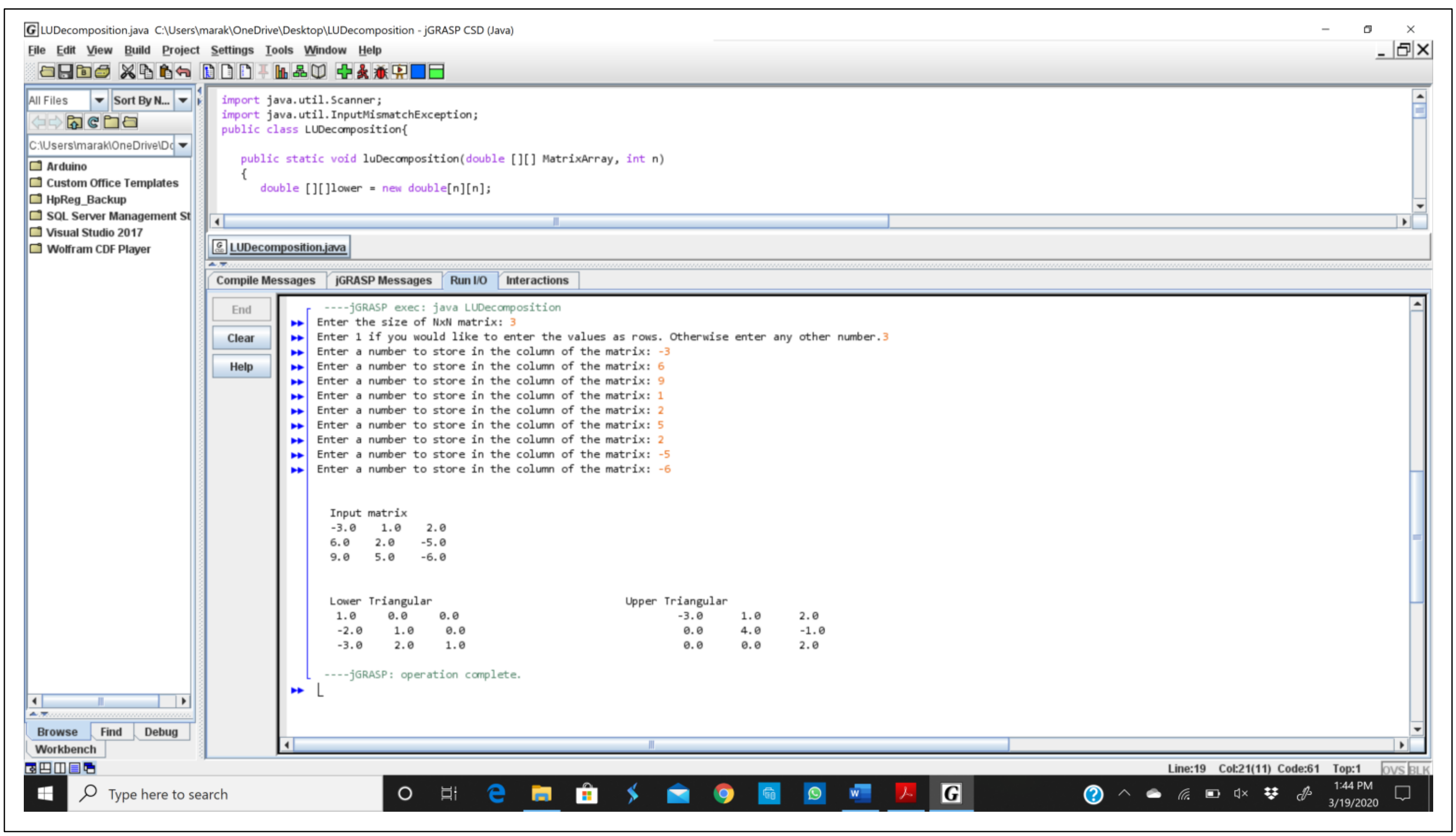
```java
// Main Method
public static void main(String arr[])
{
  int decision=0;
  Scanner sc=new Scanner(System.in);//Takes user input
  System.out.print("Enter the size of NxN matrix: ");//Asks the user to enter the size of array
  int N=sc.nextInt();//Takes and stores user input in integer N
  Scanner sc1=new Scanner(System.in);
  System.out.print("Enter 1 if you would like to enter the values as rows. Otherwise enter any other number.");

  try
  {
   decision=sc1.nextInt();
  }
  catch(InputMismatchException e)//Exception handling for invalid input.
  {
     System.out.println("Invalid Input. Try again");
     System.exit(0);//Exits the program
  }

  double[][] MatrixArray=new double[N][N];//Creates an array
```

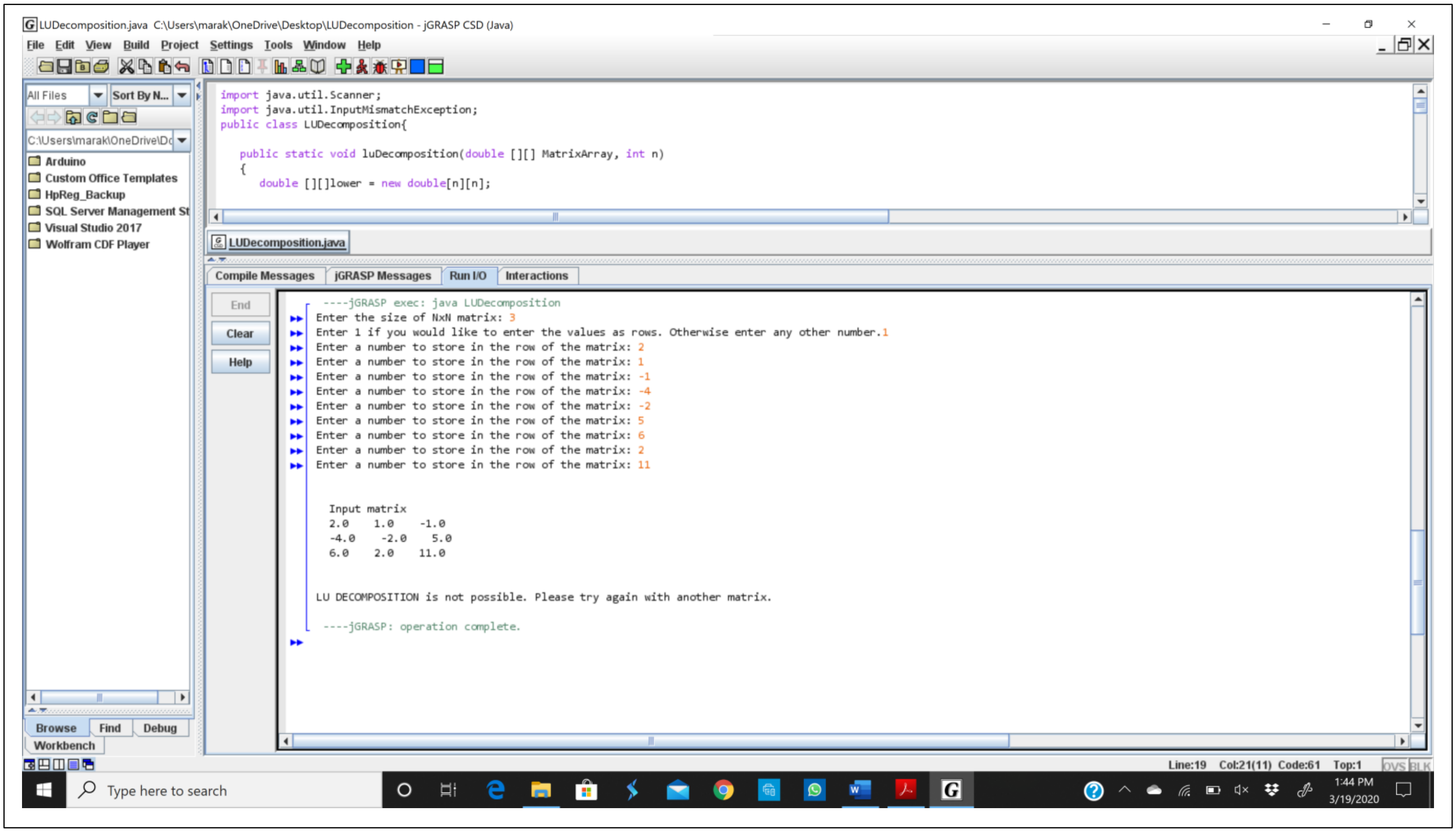


```java
  if(decision==1)
  {
   for(int i=0;i<N;i++)
   {
     for(int j=0;j<N;j++)
     {
        System.out.print("Enter a number to store in the row of the matrix: ");//asks the user to enter a number to store in the array
        MatrixArray[i][j]=sc.nextDouble();//Stores user input in the array
     }
   }
  }
  else
  {
   for(int i=0;i<N;i++)
   {
     for(int j=0;j<N;j++)
     {
        System.out.print("Enter a number to store in the column of the matrix: ");//asks the user to enter a number to store in the array
        MatrixArray[j][i]=sc.nextDouble();//Stores user input in the array
     }
   }
  }
  System.out.print("\n\n");
  System.out.println("  Input matrix  ");
  for(int i=0;i<N;i++)
  {
    for(int j=0;j<N;j++)
    {
      System.out.print("  "+MatrixArray[i][j]+"  ");
    }
    System.out.println();
  }
  System.out.print("\n\n");
  luDecomposition(MatrixArray,N);//Calls the method
 }
}
```

OUTPUTS OF THE JAVA PROGRAM






LUDecomposition.java C:\Users\marak\OneDrive\Desktop\LUDecomposition - jGRASP CSD (Java)
File Edit View Build Project Settings Tools Window Help
import java.util.Scanner;
import java.util.InputMismatchException;
public class LUDecomposition{
public static void luDecomposition(double [][] MatrixArray, int n)
{
double [][]lower = new double[n][n];
LUDecomposition.java
Compile Messages
jGRASP Messages
Run I/O
Interactions
End
Clear
Help
----jGRASP exec: java LUDecomposition
Enter the size of NxN matrix: 3
Enter 1 if you would like to enter the values as rows. Otherwise enter any other number.1
Enter a number to store in the row of the matrix: 2
Enter a number to store in the row of the matrix: 1
Enter a number to store in the row of the matrix: -1
Enter a number to store in the row of the matrix: -4
Enter a number to store in the row of the matrix: -2
Enter a number to store in the row of the matrix: 5
Enter a number to store in the row of the matrix: 6
Enter a number to store in the row of the matrix: 2
Enter a number to store in the row of the matrix: 11
Input matrix
2.0 1.0 -1.0
-4.0 -2.0 5.0
6.0 2.0 11.0
LU DECOMPOSITION is not possible. Please try again with another matrix.
----jGRASP: operation complete.
All Files
Sort By N...
Arduino
Custom Office Templates
HpReg_Backup
SQL Server Management St
Visual Studio 2017
Wolfram CDF Player
Browse
Find
Debug
Workbench
Line:19 Col:21(11) Code:61 Top:1
Type here to search
1:44 PM
3/19/2020

REFLECTION ON THE PROCESS OF LU DECOMPOSITION

According to me, LU Decomposition is of great use when solving a linear equation having many variables. This is because, finding the solution by using the inverse of the matrix, may be very hectic if the matrix has many entries. Also, without using a calculator, finding the reduced echelon form of the matrix can also be troublesome, especially when the matrix is too large. But all these problems are solved by using an LU Decomposition method. This is because the det(L)=1 and its easy to find det(U) because U has all zeros below the main diagonal. As a result, it is easy to find the inverse of L and U and so to solve the system of linear equations.

- **Strengths of LU Decomposition**

  1. As stated above, LU Decomposition makes it easier to solve a linear equation.
  2. While solving a linear equation, we don't need to know the vector b upfront because LU decomposition only uses matrix A [1]. While solving a linear system of equation using gaussian elimination, we use the augmented matrix to find the solution, so we need to know what the vector b is upfront [1].

- **Weaknesses of LU Decomposition**
    1. LU Decomposition does not exist if the echelon form of A can't be found without doing row swapping [1,3].
    2. For an invertible matrix A, LU decomposition only exists if all the leading principal minors of A are non-zero [2].
    3. If A is a singular matrix of rank k, then it admits an LU Decomposition if and only if the first k leading principal minors are non-zero [2].
- **Special types of LU Decomposition**
    1. PLU Decomposition

       Sometimes when LU Decomposition does not exist, we try to find a permutation matrix P such that PA has LU Decomposition. Here, P is a n*n permutation matrix with one entry whose value is "1" in every column and row and the other entries are "0" [1]. Since we are using the permutation matrix in this decomposition, it is called PLU Decomposition. The PLU Decomposition always exist.

2. LDU Decomposition

   An LDU Decomposition is the decomposition of the form A=LDU [2]. Here, D is the diagonal matrix. So, L and D are square matrices with same dimensions and U has the same dimensions as A.

BIBLIOGRAPHY

[1] “Linear Algebra.” *MuPAD Tutorial*, www.cfm.brown.edu/people/dobrush/am34/MuPad/LU.html, Accessed March 19,2020.

[2] “LU Decomposition.” *Wikipedia*, Wikimedia Foundation, 14 Mar. 2020, en.wikipedia.org/wiki/LU_decomposition, Accessed March 19,2020.

[3] “Matrix Factorizations.” *Linear Algebra and Its Applications*, *Fifth Edition*, by David C. Lay et al., Pearson, 2016, pp. 125–129, Accessed March 19, 2020.

[4] “Row Reduction and Echelon Forms.” Linear Algebra and Its Applications, Fifth Edition, by David C. Lay et al., Pearson, 2016, pp. 12–13, Accessed March 19, 2020.