LU and CR Elimination

A Project Report

submitted by

NANDHAKUMARAN M (ME21B2008)

SRIKRISHNAN S (ME21B2008)

RAGHAVENDRAN S(ME21B2007)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING KANCHEEPURAM

ABSTRACT

The reduced row echelon form $\mathrm{rref}(A)$ has traditionally been used for classroom examples: small matrices A with integer entries and low rank r. This paper creates a column-row rank-revealing factorization $\boldsymbol{A} = \boldsymbol{CR}$, with the first r independent columns of A in C and the r nonzero rows of $\mathrm{rref}(A)$ in R. We want to reimagine the start of a linear algebra course, by helping students to see the independent columns of A and the rank and the column space.

If B contains the first r independent rows of A, then those rows of $\boldsymbol{A} = \boldsymbol{CR}$ produce $\boldsymbol{B} = \boldsymbol{WR}$. The r by r matrix W has full rank r, where B meets C. Then the triple factorization $A = CW^{-1}B$ treats columns and rows of $A (C$ and $B)$ in the same way.

KEYWORDS: Elimination; factorization; row echelon form; matrix; rank

TABLE OF CONTENTS

ABSTRACT **i**

LIST OF FIGURES **1**

1 Problem Definition and Motivation **2**

1.1 Problem Definition 2

1.2 Motivation 2

2 Overview of mathematical methods **3**

2.1 LU Decomposition 3

2.2 CR Elimination 3

2.3 Gauss-Jordan Elimination 3

2.4 Magic Factorization 3

3 LU Decomposition **4**

3.1 Algorithm 4

3.2 Code 5

3.3 Output 6

4 CR Elimination **8**

4.1 Algorithm 8

4.2 Code 8

4.3 Output 8

5 Magic Factorization **10**

5.1 Algorithm 10

5.2 Code 10

5.3 Output 11

6 Implementation **12**

6.1 Objective 12

6.2 Code 12

6.3 Output 14

7 Conclusion and Future scope 18

7.1 Conclusion 18

7.2 Future Scope 18

CHAPTER 1

Problem Definition and Motivation

1.1 Problem Definition

Our aim is to provide a new perspective of LU decomposition, to consider the corresponding column operations needed to achieve the same result as row operations. It also illustrates the practical applications of these concepts in linear algebra and numerical methods, showing the benefits of using pivoting to improve the stability and accuracy of LU decomposition and CR elimination algorithms and the introduction of triple factorization to give equal treatment to columns and rows.

1.2 Motivation

Re-imagining the start of a linear algebra course by introducing a new way of looking at matrices and their factorization, providing a new type of factorization that breaks down a matrix into independent columns and non-zero rows. We hope that with this new approach, students can grasp the fundamental concepts of Linear Algebra and develop a more profound intuition.

CHAPTER 2

Overview of mathematical methods

2.1 LU Decomposition

Matrix factorization technique that decomposes a square matrix into two triangular matrices: a lower triangular matrix L and an upper triangular matrix U, such that **A = LU**. It is typically computed using Gaussian elimination

2.2 CR Elimination

The factorization **A = CR** reveals the first great theorem of linear algebra. The column rank r equals the row rank. Proof. The r rows of R are independent (from its sub-matrix *I*) and all rows of A are combinations of the rows of R (because of **A = CR**).

2.3 Gauss-Jordan Elimination

Method for solving systems of linear equations by using row operations to transform a matrix into an equivalent matrix in row-echelon form and then applying further row operations to transform it into reduced row-echelon form.

2.4 Magic Factorization

$$A = CW^{-1}B \tag{2.1}$$

where, C: *r* independent columns of A

B: *r* independent rows of A

W: Intersection of C and B

CHAPTER 3

LU Decomposition

3.1 Algorithm

LU decomposition Given an $n \times n$ matrix A, we want to find upper and lower triangular matrices such that $A = LU$, with $L_{ii} = 1$ for all i Assume that this is possible: $A_{ij} = \sum_{k=1}^{n} L_{ik}U_{kj}$ For $k = 0, 1, 2, \ldots, n-1$, define rank-one matrices $M^{(k)}$ with elements

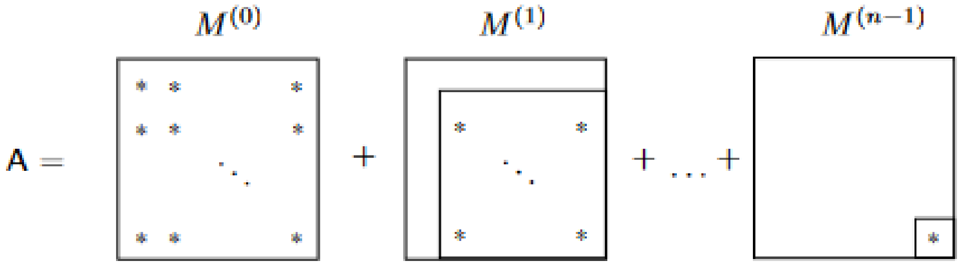
$$M_{ij}^{(k)} = L_{i,k+1}U_{k+1,j}$$

The first k rows of $M^{(k)}$ are full of zeros, as are the first k columns (by the triangular structure of L and U)

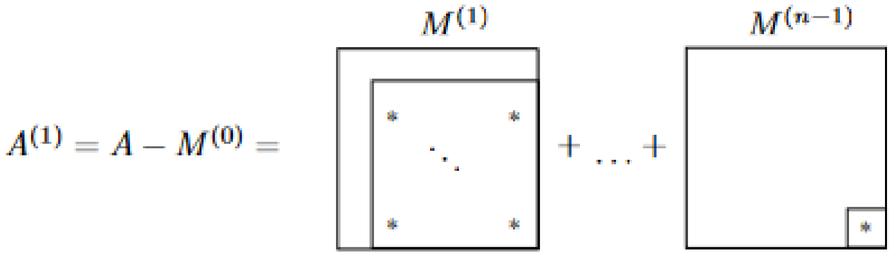
Define also $A^{(k)} = A - \sum_{r=1}^{k} M^{(r)}$, with $A^{(0)} = A$; also $A^{(n)} = 0$

The first row of $M^{(0)}$ is equal to the first row of A, and similarly for the first column Since $L_{11} = 1$, the first row of $M^{(0)}$ is the first row of U (since $M_{1j}^{(0)} = L_{11}U_{1j}$) Similarly the first column of L is U_{11} times the first column of $M^{(0)}$, and U_{11} has already been computed.

We know the first column of L and the first row of U. Hence we can compute all elements of $M^{(0)}$ (as $M_{ij}^{(0)} = L_{i1}U_{1j}$) $A^{(1)}$ is a known matrix, and its second row is the same as the second row of $M^{(1)} \ldots$ and similarly the second column.



* indicates a non-zero element



The second row of $M^{(1)}$ is the second row of U, and we can also compute the second column of L

In this way, we can work out all elements of L and U :

Let A$^{(0)} =$ A

Iterate $k = 1, 2, \dots, n$

k th row of U : $\quad U_{kj} = A_{kj}^{(k-1)} \quad j = k, \dots, n$

k th column of L : $\quad L_{ik} = A_{ik}^{(k-1)} / A_{kk}^{(k-1)} \quad i = k, \dots, n$

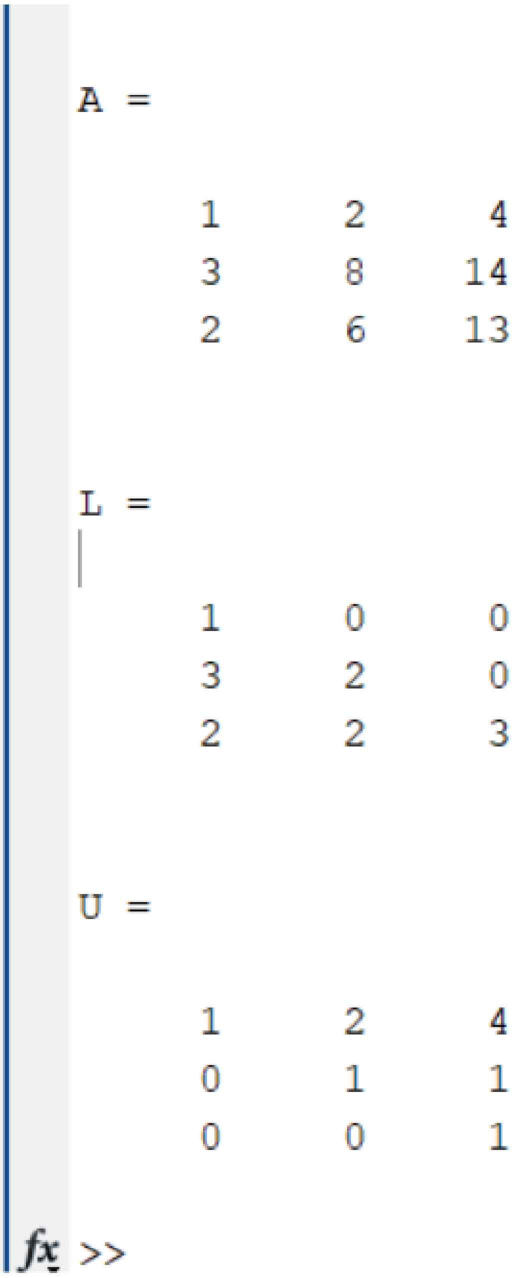
subtract $M^{(k-1)}$: $\quad A_{ij}^{(k)} = A_{ij}^{(k-1)} - L_{ik} U_{kj} \quad i, j \geq k$

This all works (and the decomposition exists) if $A_{kk}^{(k-1)} \neq 0$ for all k, otherwise it fails.

3.2 Code

```
clc, clear
A=[1,2,4; 3,8,14;2,6,13]
[ L,U ] = LUdecompo(A)
function [ L,U ] = LUdecompo(A)
%LUdecomp: decompose square matrix A as A=LU
% where L is lower triangle and U is upper triangle
   [m, n]=size(A);
   if m ~= n, error(’Input must be a square matrix.’), end
   L=zeros(n); U=zeros(n);
   % remember A^(0) is A
   AofK = A;
   for k = 1:n
       % at this point AofK is A^(k-1)
       for j = k:n
           U(k,j) = AofK(k,j);
       end
       % check that we don’t divide by zero...
       if U(k,k) == 0
           error(’** A^(k-1)_{k,k}==0 in LU decomp’)
       end
       for i = k:n
           L(i,k) = AofK(i,k)/U(k,k);
       end
       % now modify AofK so that we can use it in the
       % next iteration
       for i = k:n
           for j = k:n
               AofK(i,j) = AofK(i,j) - L(i,k)*U(k,j);
           end
       end
   end
end
```

3.3 Output



```
A =

     1     2     4
     3     8    14
     2     6    13


L =

     1     0     0
     3     2     0
     2     2     3


U =

     1     2     4
     0     1     1
     0     0     1

>>
```

CHAPTER 4

CR Elimination

4.1 Algorithm

A is a matrix with m x n and rank r

Finding row-reduced echelon form Ro->rref(A)

Pivot Values gives us the rank jb->rref(A)

Rref returns both row-reduced echelon matrix and non-zero pivots

r -> length of jb

C is the matrix with first r independent columns We Iterate row wise through A

C -> Iteration of A through all rows where column number = jb [A(:,jb)]

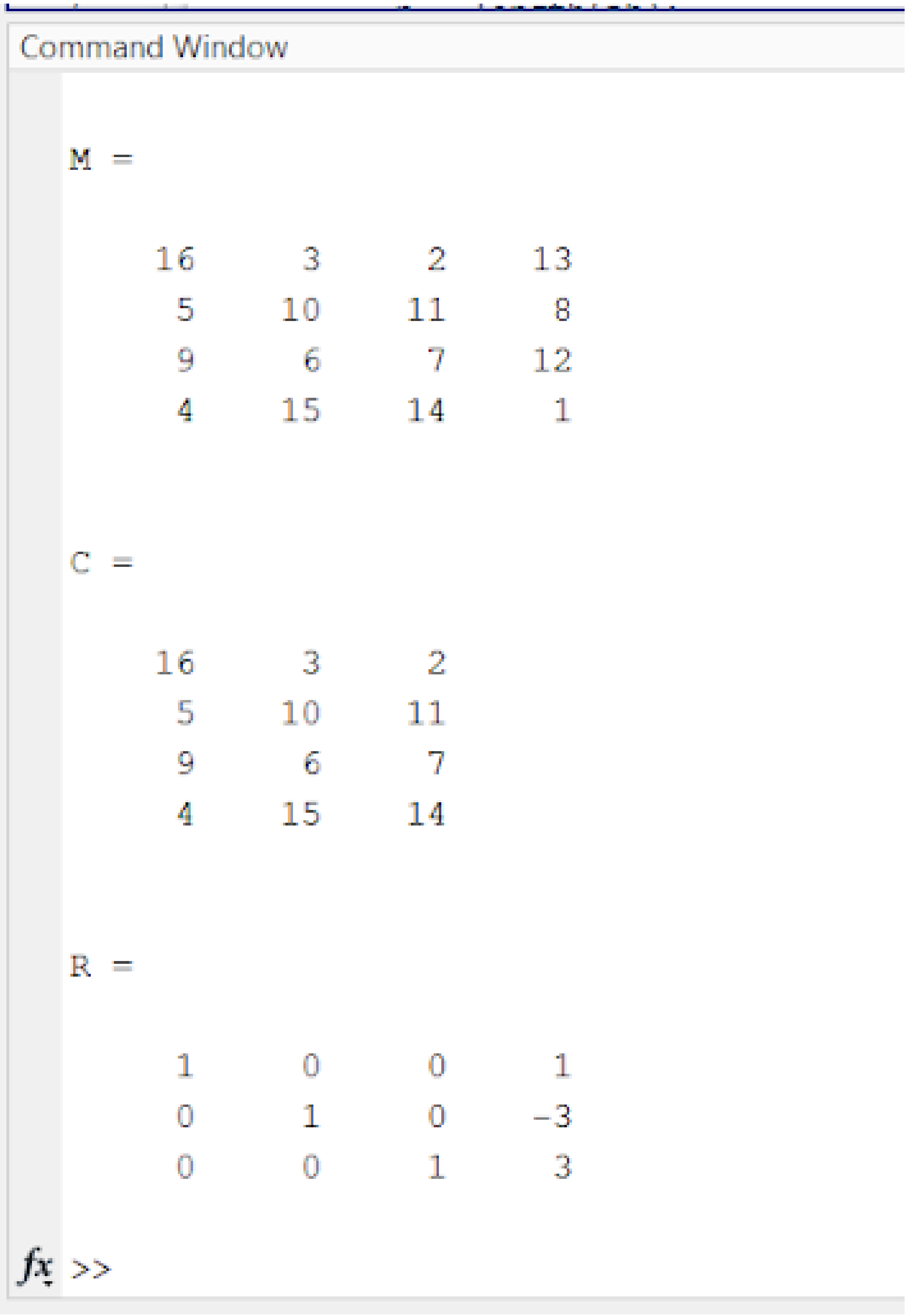
R is the matrix with the first r independent rows We iterate through matrix Ro column wise

R->Iteration of R through rows ranging from 1 to r over all columns [Ro(1:r,)]

4.2 Code

```
function [C,R] = cr(M)
    [R,jb] = rref(M);
    r = length(jb);
    R = R(1:r,:);
    C = M(:,jb)
end
```

4.3 Output



```
M =

    16     3     2    13
     5    10    11     8
     9     6     7    12
     4    15    14     1


C =

    16     3     2
     5    10    11
     9     6     7
     4    15    14


R =

     1     0     0     1
     0     1     0    -3
     0     0     1     3

>>
```

CHAPTER 5

Magic Factorization

5.1 Algorithm

A is a matrix with mxn and rank r

Cols non zero pivots of matrix A

cols->rref(A) Rows non zero pivots of matrix A Transpose rows->rref(A')

C is the independent column space

We iterate column wise through A

C->Iteration of A through all rows where column number = cols [A(:,cols)]

B is the independent row space

We iterate row wise through A

B->Iteration of A where row number = rows through all columns [A(rows,:)]

W is the intersection of C and B

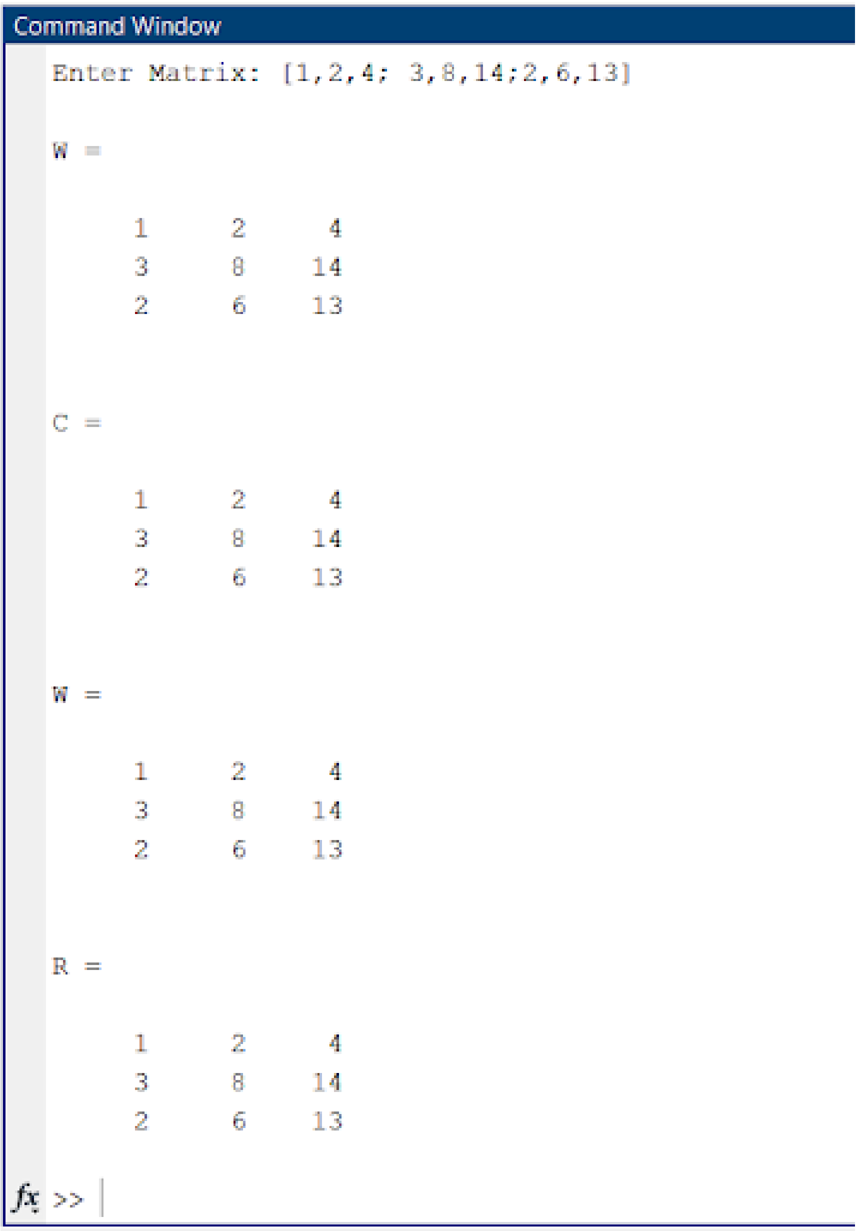
We iterate through both rows and columns simultaneously

W->Iteration of A where row number = rows and column number = cols [A(rows,cols)]

NOTE: Cols and Rows are vectors with multiple elements

5.2 Code

```
function [C,W,B] = cab(M)
    [~,cols] = rref(M);
    C = M(:,cols);
    [~,rows] = rref(M');
    B = M(rows,:);
    W = M(rows,cols)
end
```



```
Enter Matrix: [1,2,4; 3,8,14;2,6,13]

W =

     1     2     4
     3     8    14
     2     6    13


C =

     1     2     4
     3     8    14
     2     6    13


W =

     1     2     4
     3     8    14
     2     6    13


R =

     1     2     4
     3     8    14
     2     6    13

fx >> 
```

Figure 5.1: Output for magic factorization

5.3 Output

Please refer figure 5.1 for the output.

CHAPTER 6

Implementation

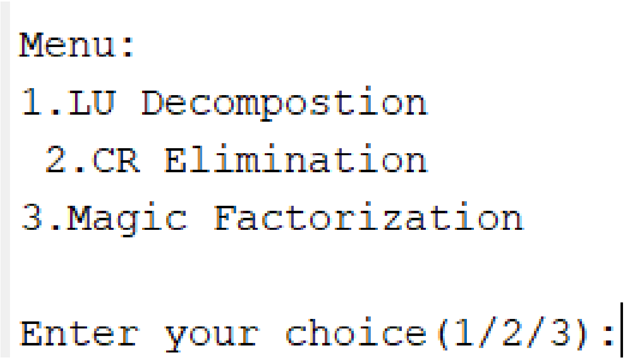
6.1 Objective

Our objective is to provide a model for students where they can explore all three methods and compare them with each other.
This integrated choice helps them choose between the three methods of their will (refer to figure 6.1).

6.2 Code

```
clc, clear
disp('Hello Student')
A=input("Enter Matrix:")
fprintf('Menu:\n1.LU Decompostion\n 2.CR Elimination\n3.Magic Fa
choice = input("Enter your choice(1/2/3):")

if choice==1
    [ L,U ] = LUdecompo(A)
elseif choice==2
    [C,R] = cr(A)
elseif choice==3
    [C,W,R] = cab(A)
end

function [ L,U ] = LUdecompo(A)
%LUdecomp: decompose square matrix A as A=LU
```

```
% where L is lower triangle and U is upper triangle
   [m, n]=size(A);
   if m ~= n, error(’Input must be a square matrix.’), end
   L=zeros(n); U=zeros(n);
   % remember A^(0) is A
   AofK = A;
   for k = 1:n
       % at this point AofK is A^(k-1)
       for j = k:n
           U(k,j) = AofK(k,j);
       end
       % check that we don’t divide by zero...
       if U(k,k) == 0
           error(’** A^(k-1)_{k,k}==0 in LU decomp’)
       end
       for i = k:n
           L(i,k) = AofK(i,k)/U(k,k);
       end
       % now modify AofK so that we can use it in the
       % next iteration
       for i = k:n
           for j = k:n
               AofK(i,j) = AofK(i,j) - L(i,k)*U(k,j);
           end
       end
   end
end


function [C,R] = cr(A)
        [R,jb] = rref(A);
        r = length(jb);
```

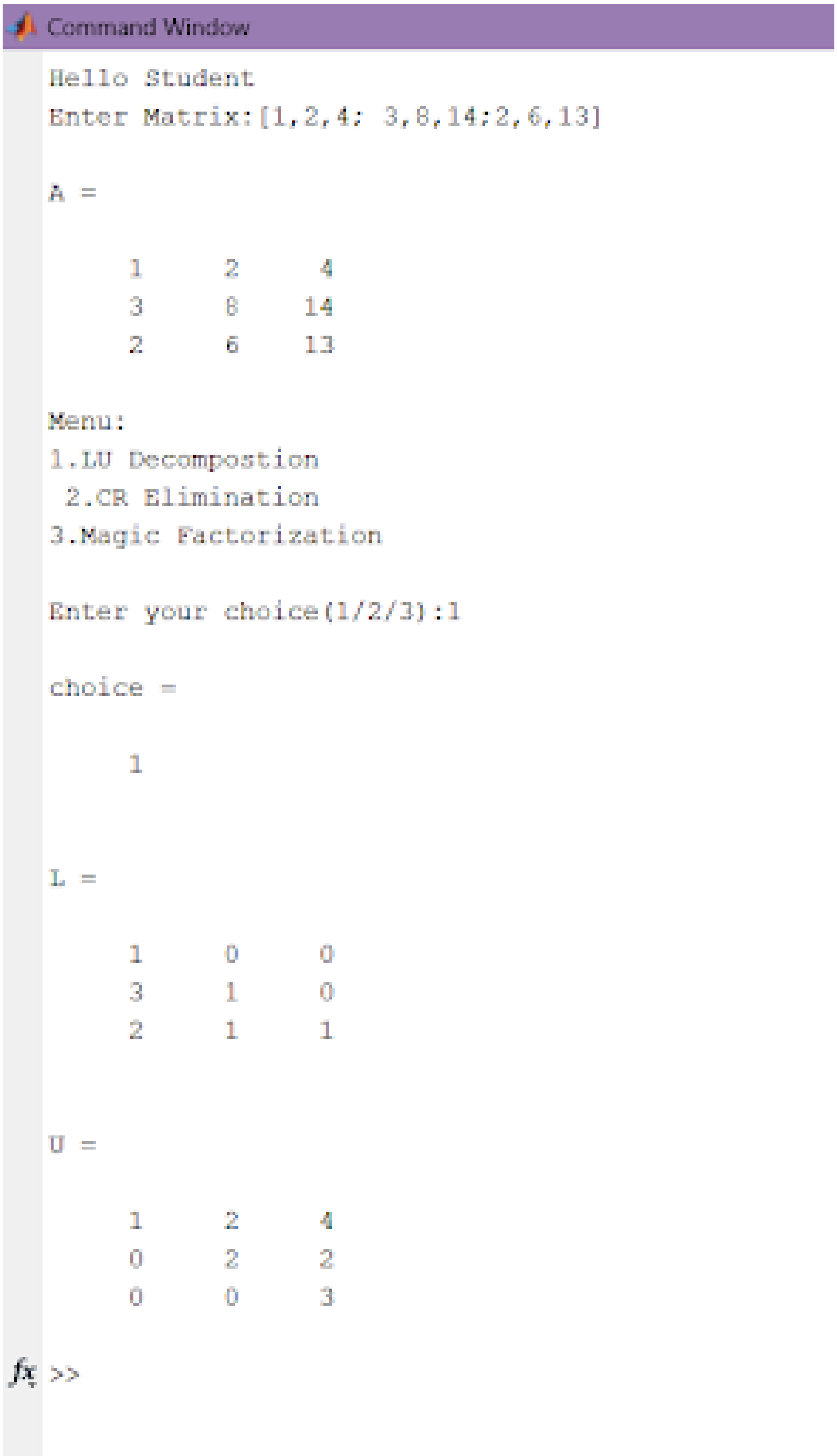


```
Menu:
1.LU Decompostion
 2.CR Elimination
3.Magic Factorization

Enter your choice(1/2/3):
```

Figure 6.1: Choices

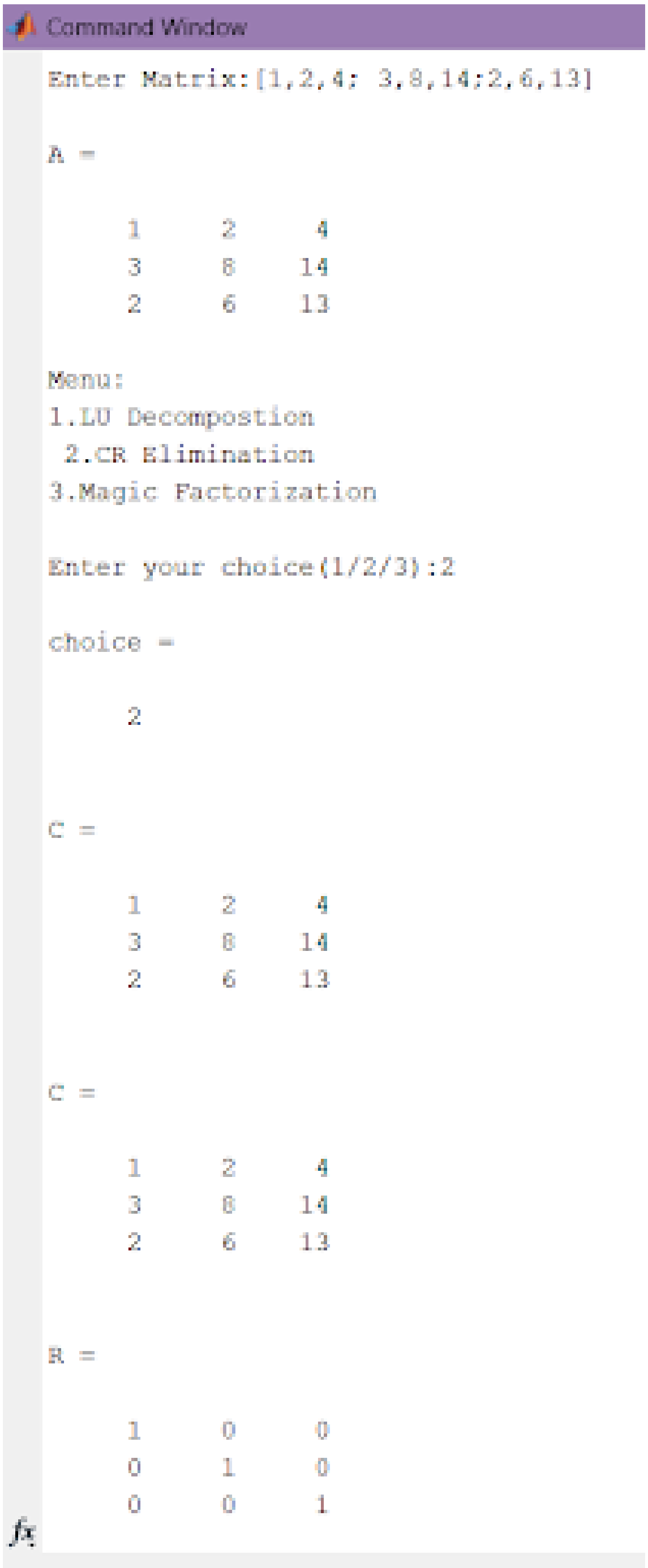
```
        R = R(1:r,:);
        C = A(:,jb)
end

function [C,W,R] = cab(A)
        [~,cols] = rref(A);
        C = A(:,cols);
        [~,rows] = rref(A');
        R = A(rows,:);
        W = A(rows,cols)
end
```

6.3 Output



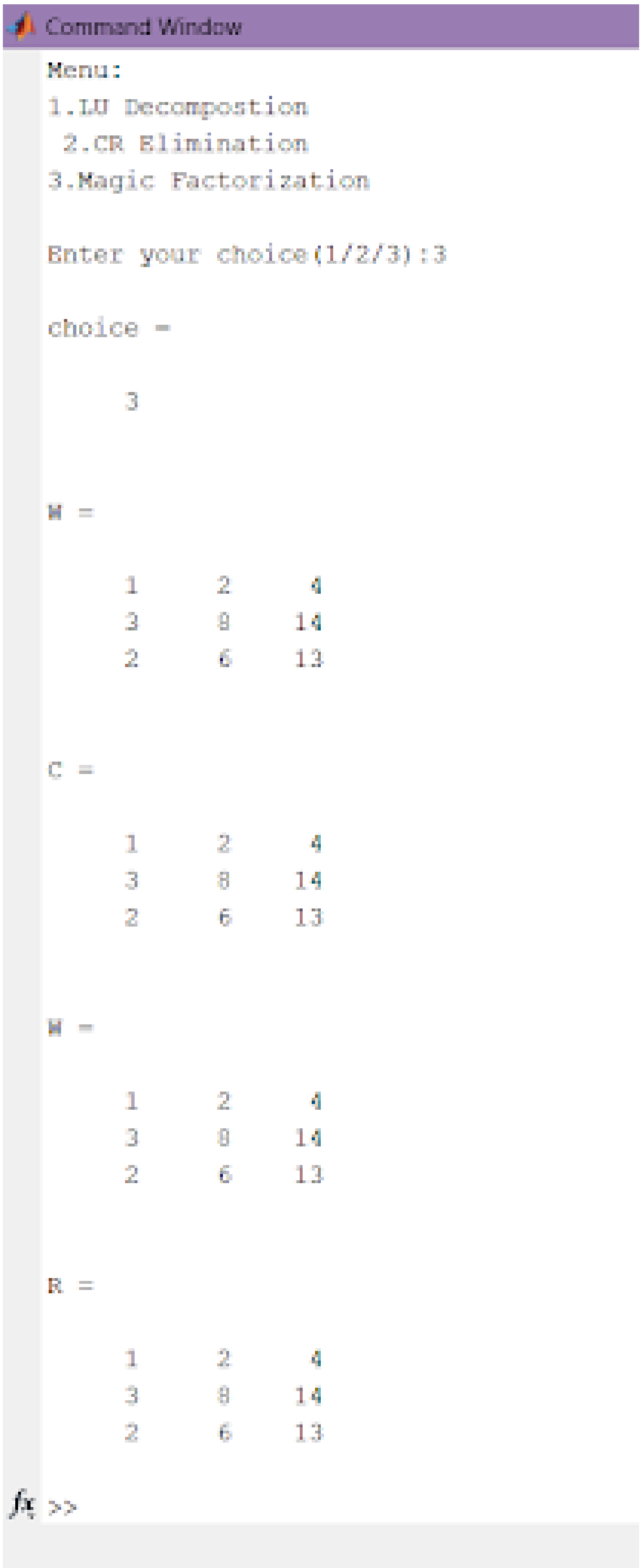
```
Hello Student
Enter Matrix:[1,2,4; 3,8,14;2,6,13]

A =

     1     2     4
     3     8    14
     2     6    13

Menu:
1.LU Decompostion
 2.CR Elimination
3.Magic Factorization

Enter your choice(1/2/3):1

choice =

     1


L =

     1     0     0
     3     1     0
     2     1     1


U =

     1     2     4
     0     2     2
     0     0     3

fx >>
```

Figure 6.2: LU Decomposition



```
Enter Matrix:[1,2,4; 3,8,14;2,6,13]

A =

     1     2     4
     3     8    14
     2     6    13

Menu:
1.LU Decompostion
 2.CR Elimination
3.Magic Factorization

Enter your choice(1/2/3):2

choice =

     2


C =

     1     2     4
     3     8    14
     2     6    13


C =

     1     2     4
     3     8    14
     2     6    13


R =

     1     0     0
     0     1     0
     0     0     1
```

Figure 6.3: CR Elimination



```
Menu:
1.LU Decompostion
 2.CR Elimination
3.Magic Factorization

Enter your choice(1/2/3):3

choice =

     3



W =

     1     2     4
     3     8    14
     2     6    13



C =

     1     2     4
     3     8    14
     2     6    13



M =

     1     2     4
     3     8    14
     2     6    13



R =

     1     2     4
     3     8    14
     2     6    13

>>
```

Figure 6.4: Magic Factorization

CHAPTER 7

Conclusion and Future scope

7.1 Conclusion

With this we were able to achieve a deeper understanding of LU and CR elimination , their differences with respect to methodology , computational resources and time and stabilization due to pivot variables.
Further we aimed to make a model where students can witness how a matrix behaves to three different decomposition techniques namely LU Decomposition , CR elimination and Magic/Triple Factorization.As intended this gives them a deeper essence of linear algebra to make their fundamentals strong.

7.2 Future Scope

In the future, LU decomposition and CR elimination techniques are poised to play significant roles in various fields. LU decomposition is expected to find applications in large-scale simulations, scientific computing, and real-time data analysis. With advancements in high-performance computing and parallelization techniques, LU decomposition can provide faster and more accurate solutions for complex systems of equations. On the other hand, CR elimination techniques will continue to evolve to ensure data integrity and reliability in communication and storage systems. As data volumes and transmission speeds increase, the demand for robust error detection and correction mechanisms will grow, making CR elimination crucial in domains like cybersecurity, network protocols, and data-driven modeling. Ongoing research and innovation in both areas will further enhance their capabilities and broaden their applications in the future.

REFERENCES

"LU Decomposition." Rosetta Code, 15 May 2023, rosettacode.org/wiki/LU decomposition.

Strang, Gilbert, and Cleve Moler. "LU and CR Elimination." SIAM Review, vol. 64, no. 1, 2022, pp. 181–190, https://doi.org/10.1137/20m1358694.