** PES University, Bangalore**

(Established under Karnataka Act No. 16 of 2013)

**APRIL 2022: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER**

**UE20MA251- LINEAR ALGEBRA**

**MATLAB Assignment**

Session: Jan-May 2022

**Name of the Student :** \_\_\_\_\_\_\_\_**\_\_\_\_Sriram R\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SRN**  **: \_\_\_\_\_\_\_\_\_\_PES1UG20CS435\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Branch** **: Computer Science and Engineering**

**Semester & Section** **: Semester IV Section H**

**FOR OFFICE USE ONLY:**

**Marks Allotted**  **: / 05**

Name of the Course Instructor :**Prof. Jyothi R**

Signature of the Course Instructor : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Gaussian Elimination**

C = [1 2 -1; 2 1 -2; -3 1 1]  
b= [3 3 -6]'  
A = [C b];  
n= size(A,1);  
x = zeros(n,1); %variable matrix [x1 x2  
... xn] column  
for i=1:n-1  
for j=i+1:n  
m = A(j,i)/A(i,i)  
A(j,:) = A(j,:) - m\*A(i,:)  
end  
end  
x(n) = A(n,n+1)/A(n,n)  
for i=n-1:-1:1  
summ = 0  
for j=i+1:n  
summ = summ + A(i,j)\*x(j,:)  
x(i,:) = (A(i,n+1) - summ)/A(i,i)  
end  
end

Output:

x = 3, y = 1, z = 2

****

**Gauss Jordan Method**

A =[1,1,1;4,3,-1;3,5,3];

n =length(A(1,:));

Aug =[A,eye(n,n)]

for j=1:n-1

for i=j+1:n

Aug(i,j:2\*n)=Aug(i,j:2\*n)-Aug(i,j)/Aug(j,j)\*Aug(j,j:2\*n)

end

end

for j=n:-1:2

Aug(1:j-1,:)=Aug(1:j-1,:)-Aug(1:j-1,j)/Aug(j,j)\*Aug(j,:)

end

for j=1:n

Aug(j,:)=Aug(j,:)/Aug(j,j)

end

B=Aug(:,n+1:2\*n)

Output :

****

**LU Decomposition**

%LU Decomposition

Ab = [1 1 1;1 2 2;1 2 3];

%% Forward Elimination

n= length(A);

L = eye(n);

% With A(1,1) as pivot Element

for i =2:3

alpha = Ab(i,1)/Ab(1,1);

L(i,1) = alpha;

Ab(i,:) = Ab(i,:) - alpha\*Ab(1,:);

end

% With A(2,2) as pivot Element

i=3;

alpha = Ab(i,2)/Ab(2,2);

L(i,2) = alpha

Ab(i,:) = Ab(i,:) - alpha\*Ab(2,:);

U = Ab(1:n,1:n)

Output :

****

**Four Fundamental Subspaces**

% Bases of four fundamental vector spaces of matrix A.

A=[1,2,3;2,-1,1];

% Row Reduced Echelon Form

[R, pivot] = rref(A)

% Rank

rank = length(pivot)

% basis of the column space of A

columnsp = A(:,pivot)

% basis of the nullspace of A

nullsp = null(A,'r')

% basis of the row space of A

rowsp = R(1:rank,:)'

% basis of the left nullspace of A

leftnullsp = null(A','r')

Output :

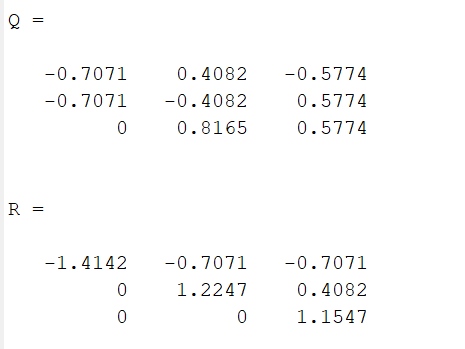
****

**QR Factorisation**

1.

Command: >> [Q,R]=qr([1,1,0;1,0,1;0,1,1])

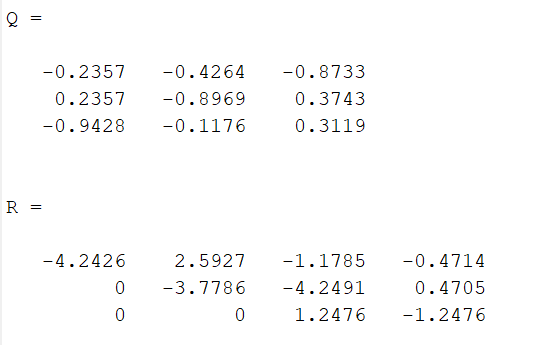
Output:



2.

Command: >> [Q,R]=qr([1,1,1,1;-1,4,4,-1;4,-2,2,0])

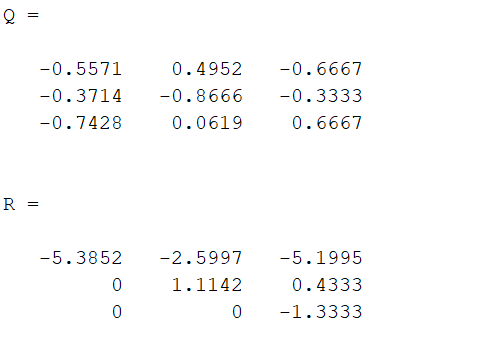
Output:



3.

Command: >> [Q,R]=qr([3,2,4;2,0,2;4,2,3])

Output:



**Projection Matrices and Least Square**

1.

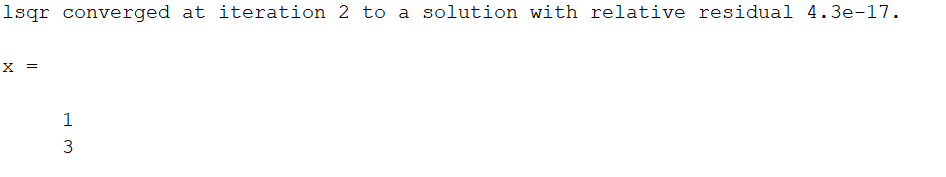
Command:

>> A=[1,0;0,1;1,1]

>> b=[1;3;4]

>> x=lsqr(A,b)

Output:



2.

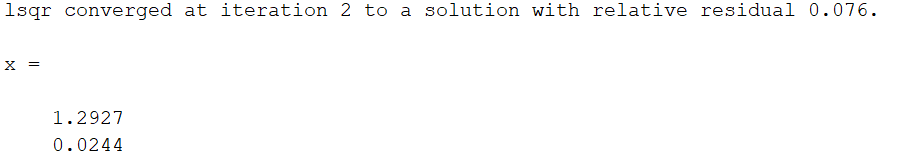
Command:

>> A=[1,0;0,2;3,1]

>> b=[1;0;4]

>> x=lsqr(A,b)

Output:



3.

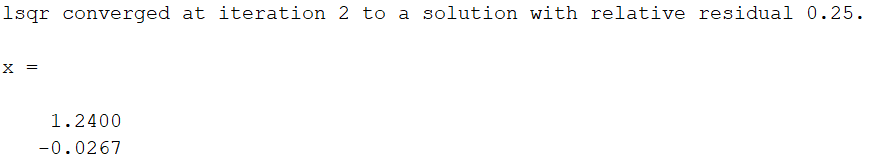
Command:

>> A=[1,2;3,1;4,1]

>> b=[1;5;4]

>> x=lsqr(A,b)

Output:



**Grams Schmidt Organisation**

1.

Command:

>> A=[1,1,2;0,0,1;1,0,0]

>> Q=zeros(3)

>> R=zeros(3)

>> for j=1:3

>> v=A(: , j)

>> for i=1:j-1

>> R(i,j)=Q(:,i)'\*A(:,j)

>> v=v-R(i,j)\*Q(:,i)

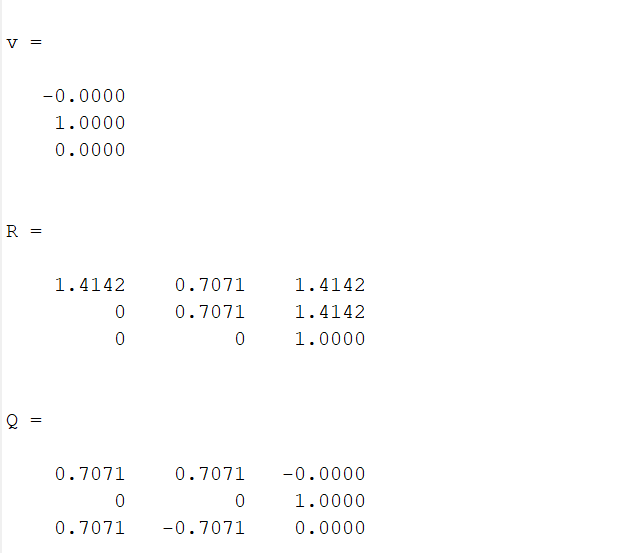
>> end

>> R(j,j)=norm(v)

>> Q(:,j)=v/R(j,j)

>> end

Output:



2. Command

>> A=[0,1,1;1,1,0;1,-1,2;1,0,-1]

>> Q=zeros(4,3)

>> R=zeros(3)

>> for j=1:3

>> v=A(: , j);

>> For i=1:j-1

>> R(i,j)=Q(:,i)'\*A(:,j)

>> v=v-R(i,j)\*Q(:,i)

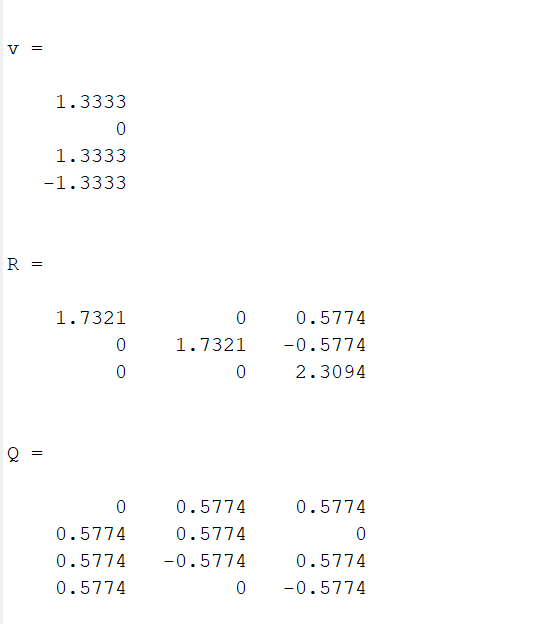
>> end

>> R(j,j)=norm(v)

>> Q(:,j)=v/R(j,j)

>> end

Output:



3.

Command:

>> A=[0,2,3;1,1,0;3,-4,2;1,5,-1]

>> Q=zeros(4,3)

>> R=zeros(3)

>> for j=1:3

>> v=A(: , j)

>> for i=1:j-1

>> R(i,j)=Q(:,i)'\*A(:,j)

>> v=v-R(i,j)\*Q(:,i)

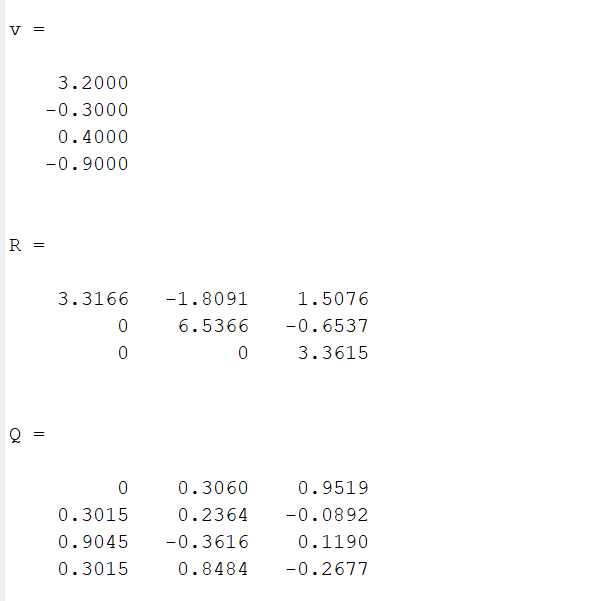
>> end

>> R(j,j)=norm(v)

>> Q(:,j)=v/R(j,j)

>> end

Output:



**GAUSS JORDAN INVERSE**

1.

**Command:**

A =[1,1,1;4,3,-1;3,5,3];

n =length(A(1,:));

Aug =[A,eye(n,n)]

for j=1:n-1

for i=j+1:n

Aug(i,j:2\*n)=Aug(i,j:2\*n)-Aug(i,j)/Aug(j,j)\*Aug(j,j:2\*n)

end

end

for j=n:-1:2

Aug(1:j-1,:)=Aug(1:j-1,:)-Aug(1:j-1,j)/Aug(j,j)\*Aug(j,:)

end

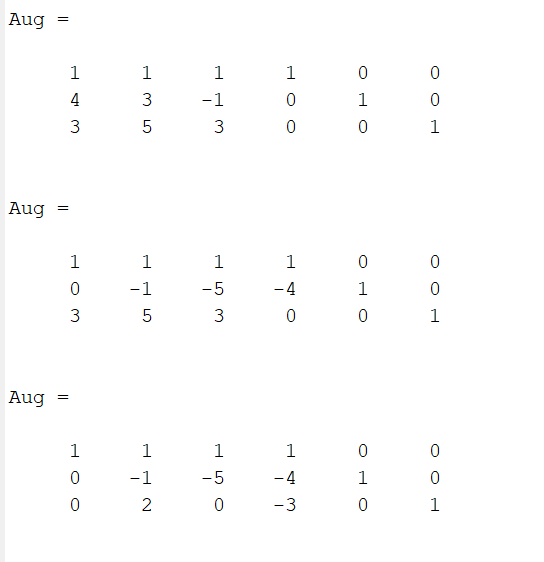
for j=1:n

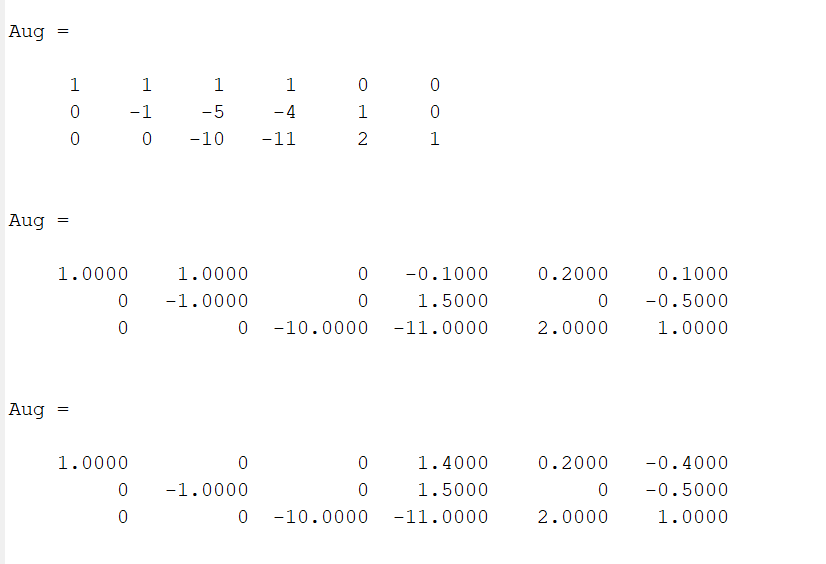
Aug(j,:)=Aug(j,:)/Aug(j,j)

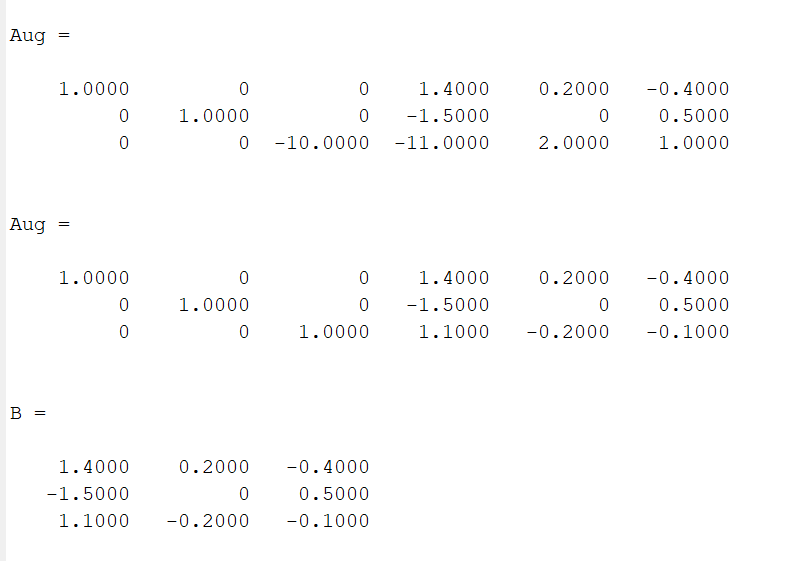
end

B=Aug(:,n+1:2\*n)

Output:







**LU DECOMPOSITION:**

1.

Command:

Ab = [1 1 -1;3 5 6;7 8 9];

n= length(A);

L = eye(n);

for i =2:3

alpha = Ab(i,1)/Ab(1,1);

L(i,1) = alpha;

Ab(i,:) = Ab(i,:) - alpha\*Ab(1,:);

end

i=3;

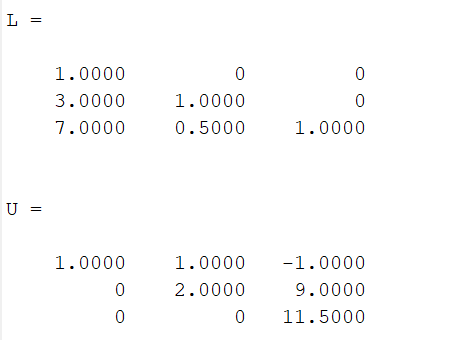
alpha = Ab(i,2)/Ab(2,2);

L(i,2) = alpha

Ab(i,:) = Ab(i,:) - alpha\*Ab(2,:);

U = Ab(1:n,1:n)

Output:



2.

Command:

Ab = [1 1 -2;3 4 6;7 -8 9];

n= length(A);

L = eye(n);

for i =2:3

alpha = Ab(i,1)/Ab(1,1);

L(i,1) = alpha;

Ab(i,:) = Ab(i,:) - alpha\*Ab(1,:);

end

i=3;

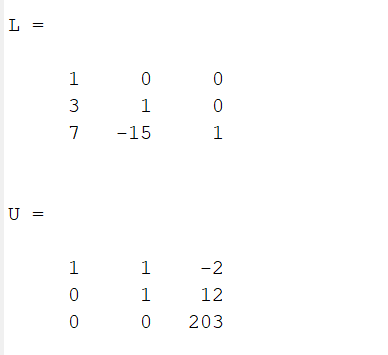
alpha = Ab(i,2)/Ab(2,2);

L(i,2) = alpha

Ab(i,:) = Ab(i,:) - alpha\*Ab(2,:);

U = Ab(1:n,1:n)

Output:



**Gauss Jordan Elimination**

1.

Command:

C = [1 2 -1; 2 1 -2; -3 1 1]

b= [3 3 -6]'

A = [C b];

n= size(A,1);

x = zeros(n,1); %variable matrix [x1 x2

... xn] column

for i=1:n-1

for j=i+1:n

m = A(j,i)/A(i,i)

A(j,:) = A(j,:) - m\*A(i,:)

end

end

x(n) = A(n,n+1)/A(n,n)

for i=n-1:-1:1

summ = 0

for j=i+1:n

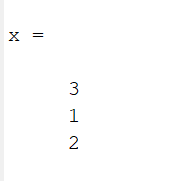
summ = summ + A(i,j)\*x(j,:)

x(i,:) = (A(i,n+1) - summ)/A(i,i)

end

end

Output:



2.

Command:

C = [2 1 -1; 2 5 7; 1 1 1]

b= [0 52 9]'

A = [C b];

n= size(A,1);

x = zeros(n,1); %variable matrix [x1 x2

... xn] column

for i=1:n-1

for j=i+1:n

m = A(j,i)/A(i,i)

A(j,:) = A(j,:) - m\*A(i,:)

end

end

x(n) = A(n,n+1)/A(n,n)

for i=n-1:-1:1

summ = 0

for j=i+1:n

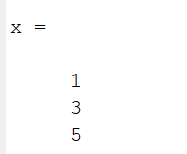
summ = summ + A(i,j)\*x(j,:)

x(i,:) = (A(i,n+1) - summ)/A(i,i)

end

end

Output:



**Eigen Values and Eigen Vectors**

1.

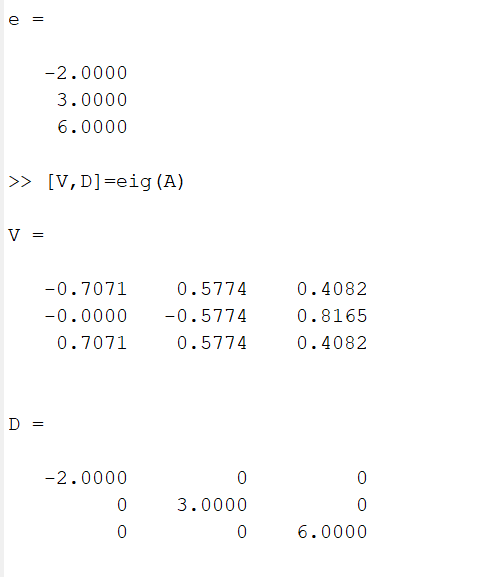
Command:

>> A=[1,1,3;1,5,1;3,1,1]

>> e=eig(A)

>> [V,D]=eig(A)

Output:



2.

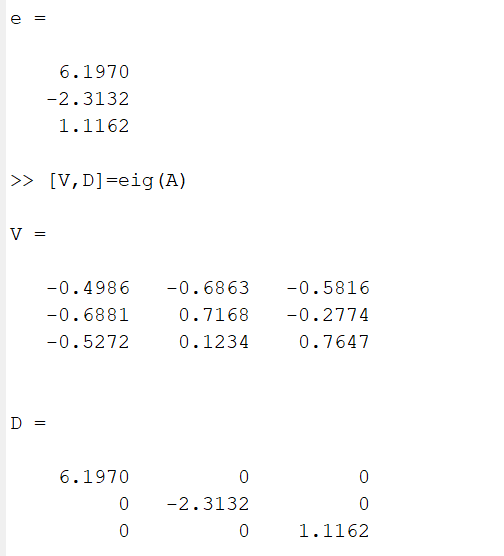
Command:

>> A=[1,3,1;4,1,3;2,1,3]

>> e=eig(A)

>> [V,D]=eig(A)

Output:



**Four Fundamental Subspaces**

1.

Command:

clc;

clear all;

close all;

% Bases of four fundamental vector spaces of matrix A.

A=[1,2,3;2,-1,1];

% Row Reduced Echelon Form

[R, pivot] = rref(A)

% Rank

rank = length(pivot)

% basis of the column space of A

columnsp = A(:,pivot)

% basis of the nullspace of A

nullsp = null(A,'r')

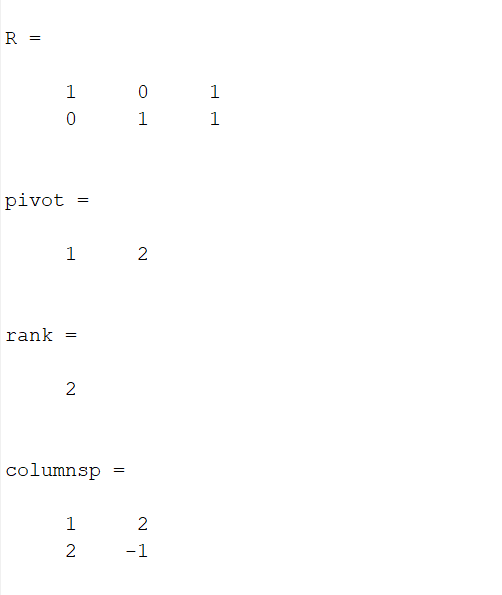
% basis of the row space of A

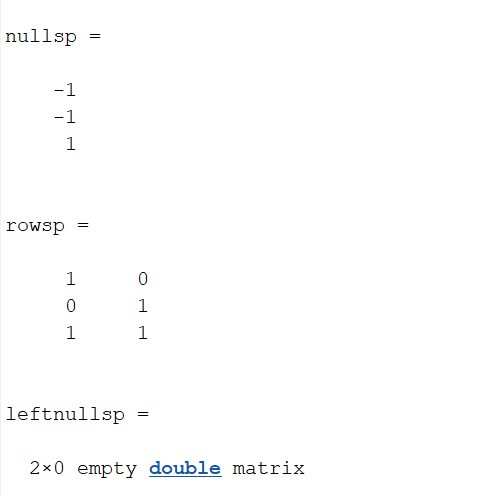
rowsp = R(1:rank,:)'

% basis of the left nullspace of A

leftnullsp = null(A','r')

Output:





2.

Command:

clc;

clear all;

close all;

% Bases of four fundamental vector spaces of matrix A.

A=[1,2,3;4,-2,1];

% Row Reduced Echelon Form

[R, pivot] = rref(A)

% Rank

rank = length(pivot)

% basis of the column space of A

columnsp = A(:,pivot)

% basis of the nullspace of A

nullsp = null(A,'r')

% basis of the row space of A

rowsp = R(1:rank,:)'

% basis of the left nullspace of A

leftnullsp = null(A','r')

Output:

