



Department of Computer Science and Engineering

Scilab

LINEAR ALGEBRA AND ITS APPLICATIONS -UE19MA251

Session: Jan-May 2021

NAME OF THE STUDENT : A Narendiran

SRN : PES1UG19CS001

**BRANCH : COMPUTER SCIENCE AND
ENGINEERING**

SEMESTER & SECTION : IV

Gaussian Elimination

1. Solve the system of equations by Gaussian Elimination. Identify the pivots

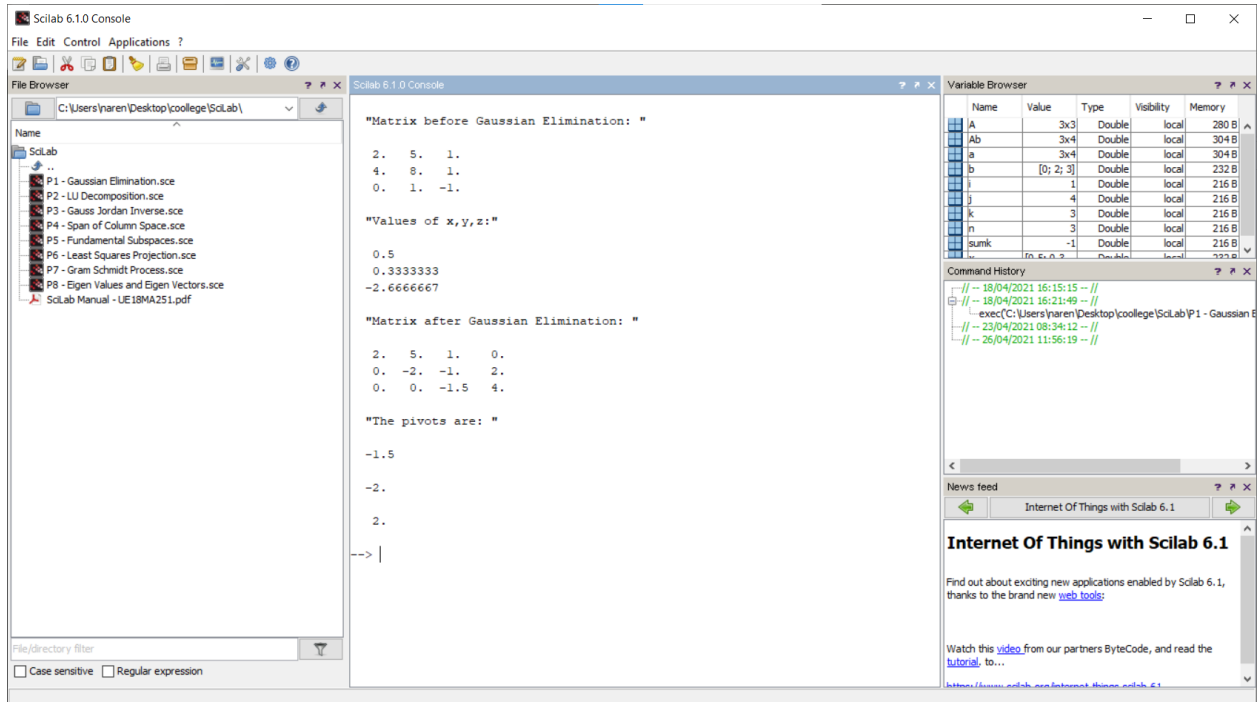
$$2x + 5y + z = 0, 4x + 8y + z = 2, y - z = 3$$

```
clc;clear;
A = [2,5,1;4,8,1;0,1,-1], b = [0;2;3];
disp("Matrix before Gaussian Elimination: ")
disp(A);
Ab = [A b];
a = Ab;
n = 3;
for i = 2:n
    for j=2:n+1
        a(i,j) = a(i,j) - a(1,j)*a(i,1)/a(1,1);
    end
    a(i,1) = 0;
end
for i=3:n
    for j=3:n+1
        a(i,j) = a(i,j)-a(2,j)*a(i,2)/a(2,2);
    end
    a(i,2) = 0;
end

x(n) = a(n,n+1)/a(n,n);
for i=n-1:-1:1
    sumk = 0;
    for k=i+1:n
        sumk = sumk+a(i,k)*x(k);
    end
    x(i) = (a(i,n+1) - sumk)/a(i,i);
end

disp("Values of x,y,z:")
disp(x);
disp("Matrix after Gaussian Elimination: ")
```

```
disp(a);
disp("The pivots are: ");
disp(a(3,3),a(2,2),a(1,1));
```



LU decomposition of a matrix

1. Factorize the Matrix as $A = LU$

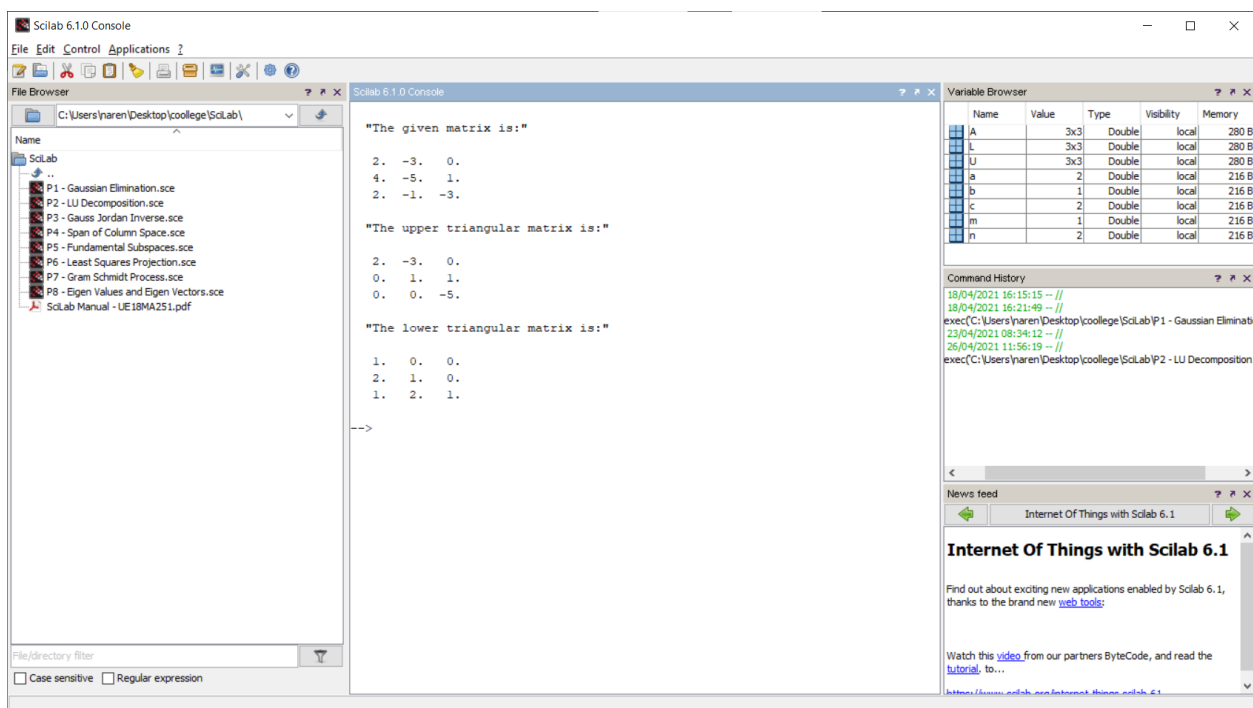
$$A = \begin{pmatrix} 2 & -3 & 0 \\ 4 & -5 & 1 \\ 2 & -1 & -3 \end{pmatrix}$$

```
clear;clc;
A = [2 -3 0;4 -5 1; 2 -1 -3];
U = A;
disp("The given matrix is:",A);
m = det(U(1,1));
n = det(U(2,1));
a = n/m;
U(2,:) = U(2,:) - U(1,:)/(m/n);
```

```

n = det(U(3,1));
b = n/m;
U(3,:) = U(3,:) - U(1,:)/(m/n);
m = det(U(2,2));
n = det(U(3,2));
c = n/m;
U(3,:) = U(3,:) - U(2,:)/(m/n);
disp("The upper triangular matrix is:",U);
L = [1,0,0;a,1,0;b,c,1];
disp("The lower triangular matrix is:",L);

```



The Gauss - Jordan method of calculating A^{-1}

1. Find the inverse of the matrix A:

```

(1 0 0
 1 1 1
 0 0 1)

```

```

clc;clear;
A = [1 0 0;1 1 1;0 0 1];
n = length(A(1,:));
Aug = [A,eye(n,n)];

```

//Forward Elimination

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

//Backward Elimination

for j = n:-1:2

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

end

//Diagonal Normalization

for j=1:n

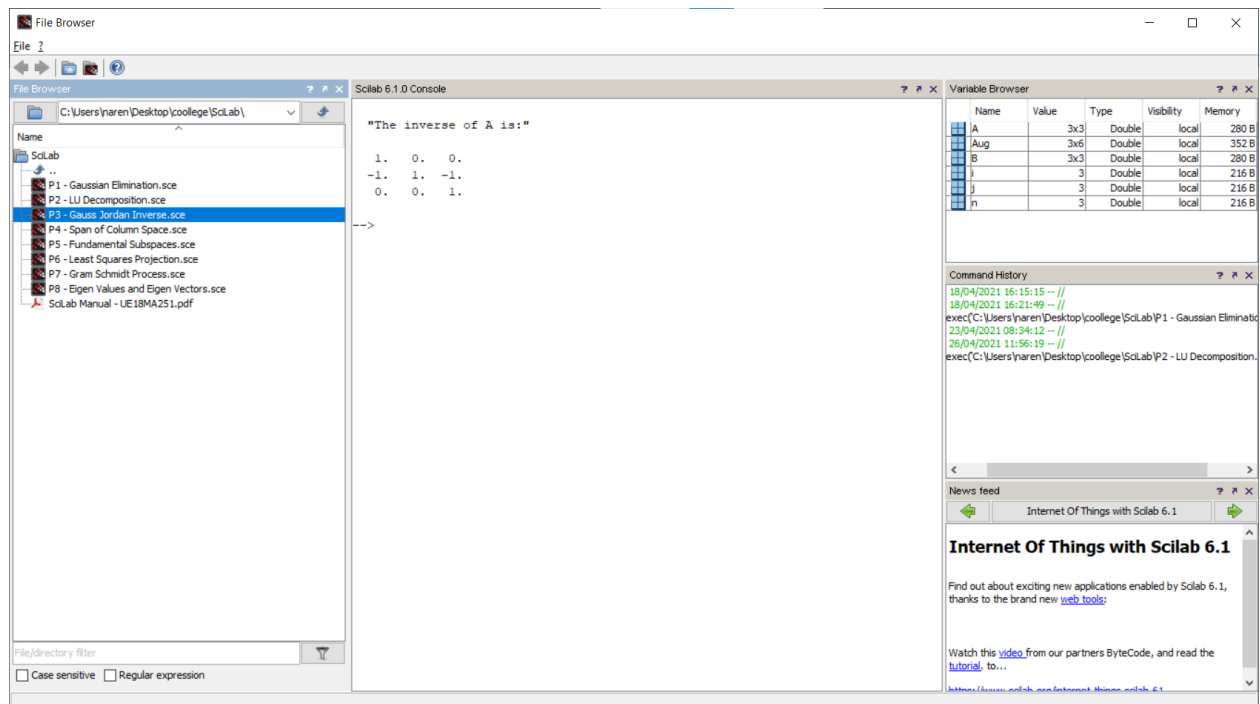
Aug(j,:) = Aug(j,:)/Aug(j,j);

end

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

disp("The inverse of A is:");

disp(B);

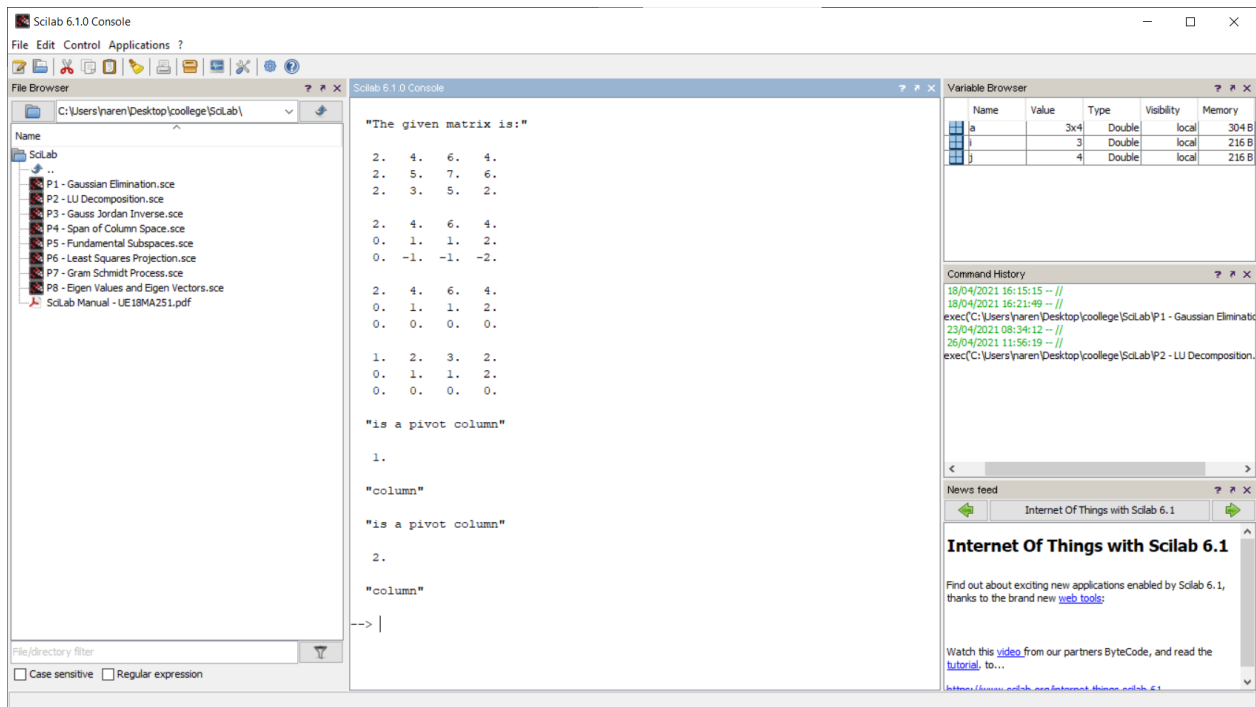


Span of the Column Space of A

Identify the columns that span the column space of A

$$A = \begin{pmatrix} 2 & 4 & 6 & 4 \\ 2 & 5 & 7 & 6 \\ 2 & 3 & 5 & 2 \end{pmatrix}$$

```
clc;clear;
a = [2 4 6 4;2 5 7 6;2 3 5 2];
disp("The given matrix is:");
disp(a);
a(2,:) = a(2,:)-(a(2,1)/a(1,1))*a(1,:);
a(3,:) = a(3,:)-(a(3,1)/a(1,1))*a(1,:);
disp(a);
a(3,:) = a(3,:)-(a(3,2)/a(2,2))*a(2,:);
disp(a);
a(1,:) = a(1,:)/a(1,1);
a(2,:) = a(2,:)/a(2,2);
disp(a);
for i=1:3
    for j=i:4
        if(a(i,j)<>0)
            disp("is a pivot column",j,"column");
            break;
        end
    end
end
end
```



The Four Fundamental Subspaces

Find the four fundamental subspaces of

$$A = \begin{bmatrix} 1 & 3 & 3 & 2 \\ 2 & 6 & 9 & 7 \\ -1 & -3 & 3 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 & 3 & 2 \\ 2 & 6 & 9 & 7 \\ -1 & -3 & 3 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 & 3 & 2 \\ 2 & 6 & 9 & 7 \\ -1 & -3 & 3 & 4 \end{bmatrix}$$

clc;clear;

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

disp("The given matrix is:");

disp(A);

[m,n] = size(A);

disp(m,"m = ");

disp(n,"n = ");

[v,pivot] = rref(A);

disp(rref(A),"Row Reduced Echelon Form: ");

r = length(pivot);

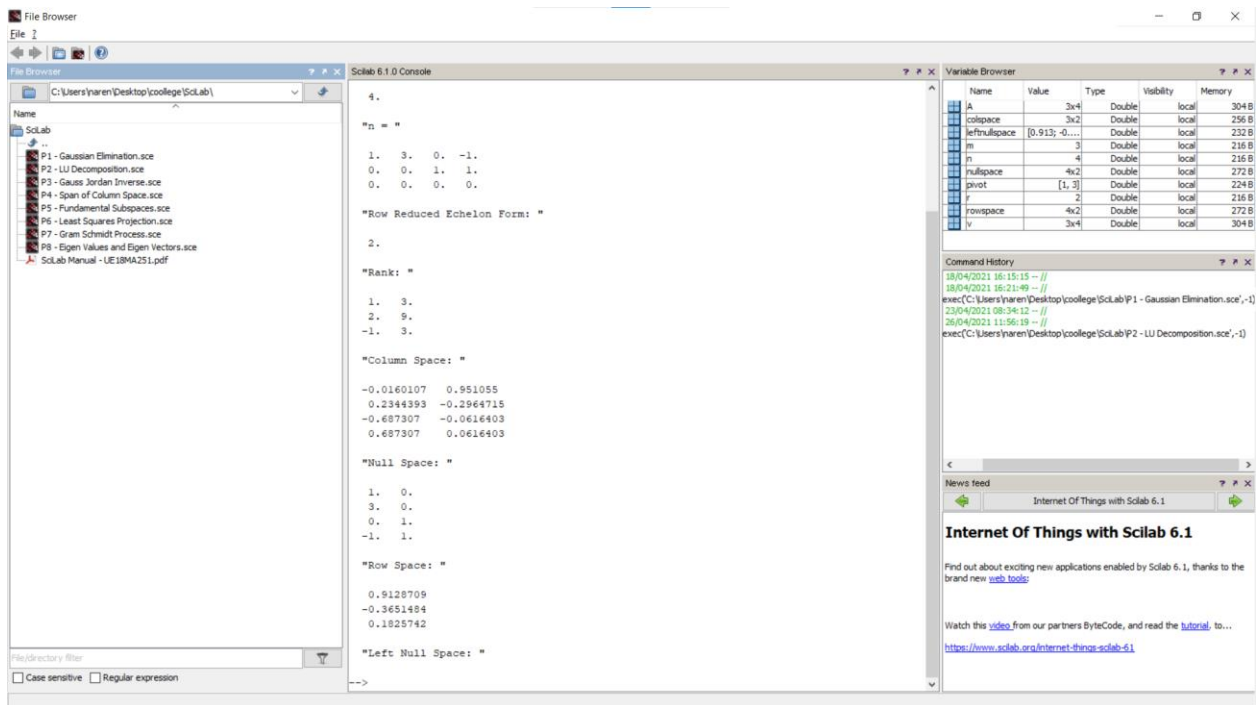
disp(r,"Rank: ");

colspace = A(:,pivot);

```

disp(colspace,"Column Space: ");
nullspace = kernel(A);
disp(nullspace,"Null Space: ");
rowspace = v(1:r,:);
disp(rowspace,"Row Space: ");
leftnullspace = kernel(A');
disp(leftnullspace,"Left Null Space: ");

```



Projections by Least Squares

1. Solve $Ax = b$ by least squares where

$A =$

$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$

$b =$

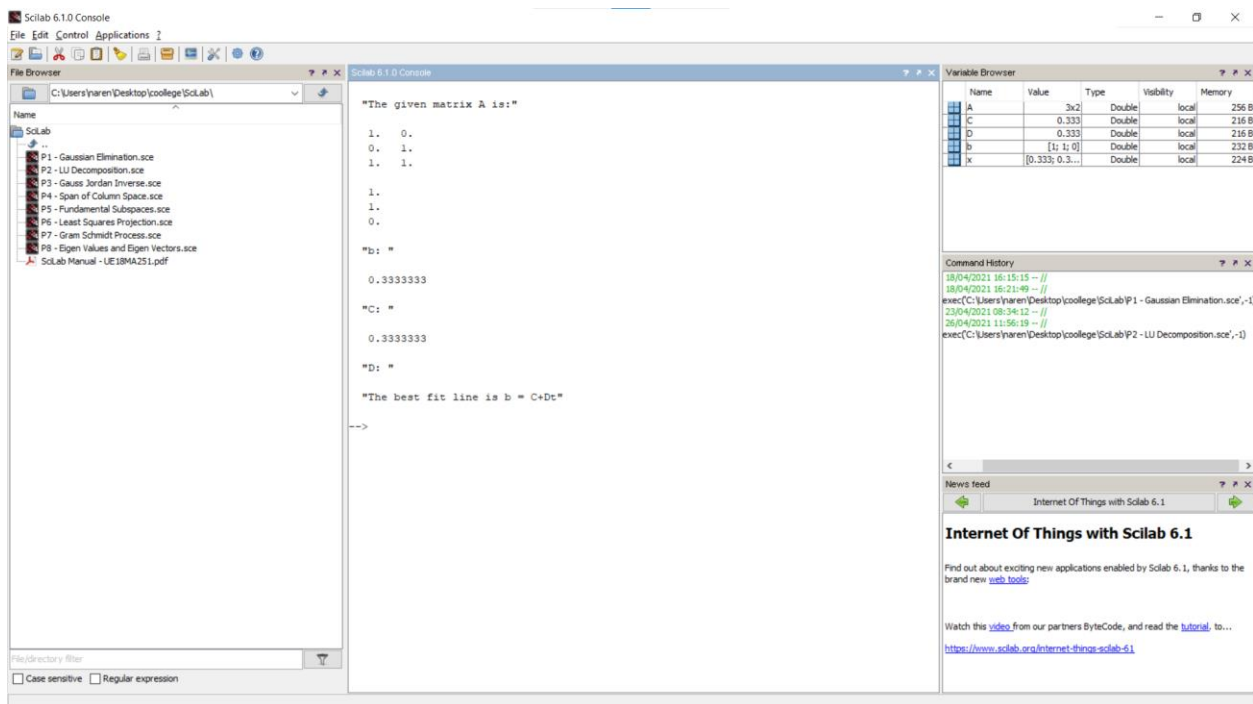
$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

1

0)

```
clc;clear;
A = [1 0;0 1;1 1];
b = [1;1;0];
disp("The given matrix A is:")
disp(A);
disp(b, "b: ");
x = (A'*A)\(A'*b)
C = x(1,1);
D = x(2,1);
disp(C,"C: ");
disp(D,"D: ");
disp("The best fit line is b = C+Dt")
```



The Gram- Schmidt Orthogonalization

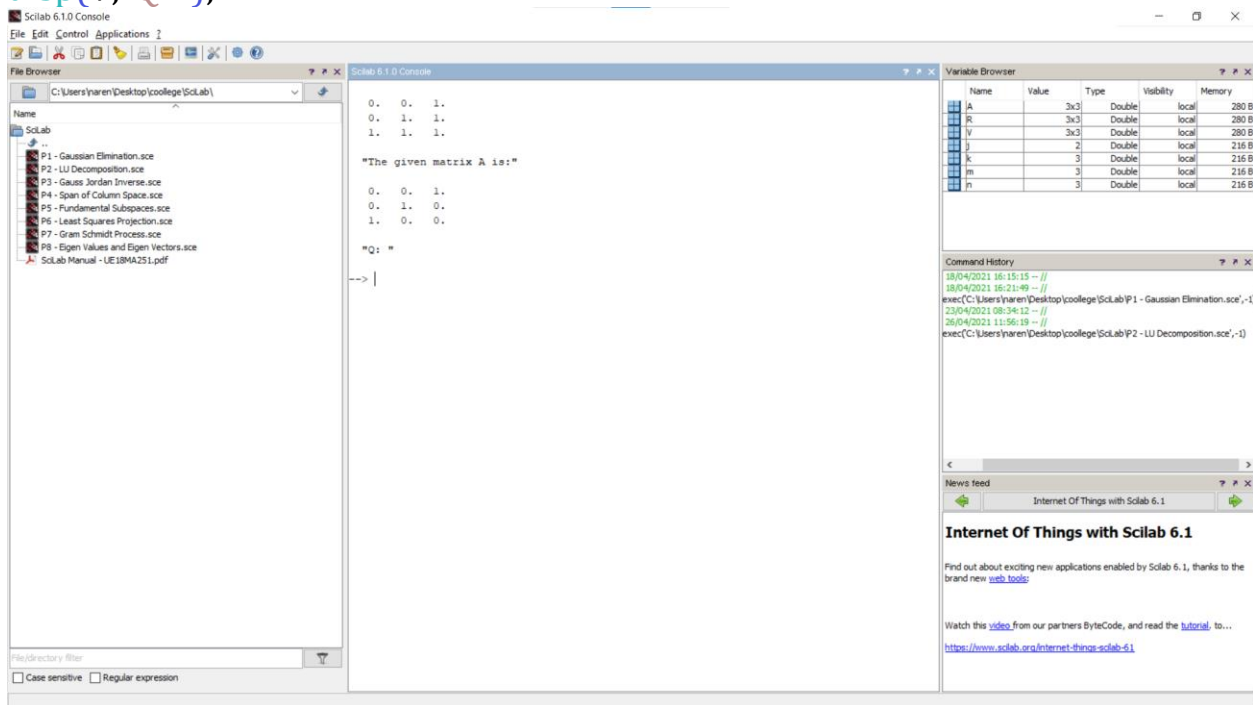
Apply the Gram – Schmidt process to the set of vectors and find the

orthogonal matrix: $(0, 0, 1), (0, 1, 1), (1, 1, 1)$

```

clc;clear;
A= [0 0 1;0 1 1;1 1 1];
disp(A, "The given matrix A is:");
[m,n] = size(A);
for k=1:n
    V(:,k) = A(:,k);
    for j=1:k-1
        R(j,k) = V(:,j)'*A(:,k);
        V(:,k) = V(:,k)-R(j,k)*V(:,j);
    end
    R(k,k) = norm(V(:,k));
    V(:,k) = V(:,k)/R(k,k);
end
disp(V,"Q: ");

```



Eigen values and Eigen vectors of a given square matrix

Find the Eigen values and the corresponding Eigen vectors of the matrix A

A= (

2 2 1

1 3 1

1 2 2)

```
clc;clear;
A = [2 2 1;1 3 1;1 2 2];
disp(A,"The given matrix A is: ")
lam = poly(0,"lam");
charMat = A-lam*eye(3,3);
disp(charMat,"The Characteristic Matrix is: ");
charPoly = poly(A,"lam");
disp(charPoly,"The Characteristic Polynomial is:");
lam = spec(A);
disp(lam,"Eigen Values: ");
function [x, lam]=eigenvectors(A)
    [n,m] = size(A);
    lam = spec(A)';
    x = [];
    for k=1:3
        B = A-lam(k)*eye(3,3);
        C = B(1:n-1,1:n-1);
        b = -B(1:n-1,n);
        y = C\b;
        y = [y;1];
        y = y/norm(y);
        x = [x y];
    end
endfunction
[x,lam] = eigenvectors(A);
disp(x,"Eigen Vectors of A: ");
```

Scilab 6.1.0 Console

File Edit Control Applications ?

File Browser

C:\Users\haren\Desktop\college\Scilab\

Name

Scilab

..

P1 - Gaussian Elimination.sce

P2 - LU Decomposition.sce

P3 - Gauss Jordan Inverse.sce

P4 - Span of Column Space.sce

P5 - Fundamental Subspaces.sce

P6 - Least Squares Projection.sce

P7 - Gram Schmidt Process.sce

P8 - Eigen Values and Eigen Vectors.sce

Scilab Manual - UE18MA251.pdf

File/directory filter

☐ Case sensitive
 ☐ Regular expression

Scilab 6.1.0 Console

```

2. 2. 1.
1. 3. 1.
1. 2. 2.

"The given matrix A is: "

2 -lam 2 1
1 3 -lam 1
1 2 2 -lam

"The Characteristic Matrix is: "

-5 +11lam -7lam* +lam*

"The Characteristic Polynomial is:"

1. + 0.i
5. + 0.i
1. + 0.i

"Eigen Values: "

0. 0.5773503 0.
-0.4472136 0.5773503 -0.4472136
0.8944272 0.5773503 0.8944272

"Eigen Vectors of A: "

-->

```

Variable Browser

Name	Value	Type	Visibility	Memory
A	3x3	Double	local	280 B
charMat	3x3	Polynomial	local	408 B
charPoly	1x1	Polynomial	local	280 B
lam	[1 - 0, 5 - ...]	Double	local	256 B
x	3x3	Double	local	280 B

Command History

```

18/04/2021 16:15:15 -- //
18/04/2021 16:21:49 -- //
exec(C:\Users\haren\Desktop\college\Scilab\P1 - Gaussian Elimination.sce',-1)
22/04/2021 09:34:12 -- //
26/04/2021 11:56:19 -- //
exec(C:\Users\haren\Desktop\college\Scilab\P2 - LU Decomposition.sce',-1)

```

News feed

Internet Of Things with Scilab 6.1

Find out about exciting new applications enabled by Scilab 6.1, thanks to the brand new [web tools](#).

Watch this [video](#) from our partners ByteCode, and read the [tutorial](#), to...

<https://www.scilab.org/internet-things-scilab-6.1>