Department of Computer Science and Engineering

Scilab

LINEAR ALGEBRA AND ITS APPLICATIONS -UE19MA251

Session: Jan-May 2021

NAME OF THE STUDENT : RAMYA NARASIMHA PRABHU

SRN : PES1UG19CS380

BRANCH : COMPUTER SCIENCE AND ENGINEERING

SEMESTER & SECTION : IV F 2

Problem Number - 1: Gaussian Elimination

Procedure: Given a system of n equations in n unknowns we use the method of pivoting to solve for the unknowns. The method starts by subtracting multiples of the first equation from the other equations. The aim is to eliminate the first unknown from the second equation onwards. We use the coefficient of the first unknown in the first equation as the first pivot to achieve this. At the end of the first stage of elimination, there will be a column of zeros below the first pivot. Next, the pivot for the second stage of elimination is located in the second row second column of the system. A multiple of the second equation will now be subtracted from the remaining equations using the second pivot to create zeros just below it in that column. The process is continued until the system is reduced to an upper triangular one. The system can now be solved backward bottom to top.

Example : Solve the following system of equations by Gaussian Elimination. Identify the pivots

$$x + 2y - z = 6$$

$$2x + y + z = 3$$

$$x - y + z = -2$$

Solution:

```
//x+ 2y- z= 6

//2x+ y+ z= 3

//x- y+ z= −2

clc; clear; close;

A = [1, 2, -1; 2, 1, 1; 1, -1, 1], b = [6; 3;-2]

A_aug = [A b]

a = A_aug

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('The values of x, y, z are ', x(1),x(2),x(3) )
disp('The pivots are ', a(1,1),a(2,2),a(3,3))
```

```
//x + 2y − z = 6
//2x + y + z = 3
//x − y + z = −2
clc; clear; close;
A = [1, 2, -1; 2, 1, 1; 1, -1, 1], b = [6; 3;-2]
A_aug = [A b]
a = A_aug
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('The values of x, y, z are ', x(1),x(2),x(3) )
disp('The pivots are ', a(1,1),a(2,2),a(3,3))
```

Output:

```
"The values of x, y, z are "

 1.

 2.

-1.

"The pivots are "

 1.

-3.

-1.

-->
```

Variable Browser

	Name	Value	Type	Visibility	Memory
	A	3x3	Double	local	280 B
	A_aug	3x4	Double	local	304 B
	a	3x4	Double	local	304 B
	b	[6; 3; -2]	Double	local	232 B
	i	1	Double	local	216 B
	j	4	Double	local	216 B
	k	3	Double	local	216 B
	n	3	Double	local	216 B
	sumk	5	Double	local	216 B
	x	[1; 2; -1]	Double	local	232 B

Example: Solve the following system of equations by Gaussian Elimination. Identify the pivots.

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

Solution:

```
clc; clear; close;
A = [2, 5, 1; 4, 8, 1; 0, 1, -1], b = [0; 2; 3]
A_aug = [A b]
a = A_aug
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('The values of x, y, z are ', x(1),x(2),x(3) )
disp('The pivots are ', a(1,1),a(2,2),a(3,3))
```

```
//2x + 3y + z = 8 , 4x + 7y + 5z = 20, -2y + 2z = 0
clear;close();clc;
format('v', 5);
A=[2,3,1;4,7,5;0,-2,2];
for l=1:3
    L(l,l)=1;
end
for i=1:3
    for j=1:3
        s=0;
        if j>=i
            for k=1:i-1
                s=s+L(i,k)*U(k,j);
            end
            U(i,j)=A(i,j)-s;
        else
            for k=1:j-1
                s=s+L(i,k)*U(k,j);
            end
            L(i,j)=(A(i,j)-s)/U(j,j);
        end
    end
    
end
b=[8;20;0];
c=L\b;
x=U\c;
disp(x,'Solution of the given equation is:')

```

Output

	Name	Value	Type	Visibility	Memory
	A	3x3	Double	local	280 B
	A_aug	3x4	Double	local	304 B
	a	3x4	Double	local	304 B
	b	[0; 2; 3]	Double	local	232 B
	i	1	Double	local	216 B
	j	4	Double	local	216 B
	k	3	Double	local	216 B
	n	3	Double	local	216 B
	sumk	-1	Double	local	216 B
	x	[0.5; 0.333;...	Double	local	232 B

```
 "The values of x, y, z are "

  0.5000000

  0.3333333

 -2.6666667

 "The pivots are "

  2.

 -2.

 -1.5

-->
```

<u>Problem Number - 2: LU decomposition of a matrix</u>

Procedure: Given a square matrix A, we can factorize A as a product of two matrices L and U. The matrix U is upper triangular with pivots on the main diagonal and the matrix L is lower triangular with 1's on the main diagonal and the multipliers in their respective positions. The factorization is carried out using the method of Gaussian Elimination.

Example: Find the triangular factors L and U for the matrix A =

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

Soulution:

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

U = A;

disp(A, "The given matrix 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 (2,:) / (m/n) ;

m = det ( U(2,2) );

n = det ( U(3,2) );

c = n/m;

L = [1, 0, 0; a, 1, 0; b, c, 1]
```

```
disp(U, "The Upper Traingular Matrix U=")

disp(L, "The Lower Traingular Matrix L=")
```

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

U = A;
disp(A, "The given matrix 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 (2,:) / (m/n) ;

m = det ( U(2,2) );
n = det ( U(3,2) );
c = n/m;

L = [1, 0, 0; a, 1, 0; b, c, 1]

disp(U, "The Upper Traingular Matrix U=")

disp(L, "The Lower Traingular Matrix L=")
```

Output:

```
 3.    1.    2.
 2.   -3.   -1.
 1.    2.    1.

"The given matrix A="

 3.    1.           2.
 0.   -3.6666667   -2.3333333
 1.    3.2222222    1.7777778

"The Upper Traingular Matrix U="

 1.          0.          0.
 0.6666667   1.          0.
 0.3333333  -0.8787879   1.

"The Lower Traingular Matrix L="
```

Name	Value	Type	Visibility	Memory
A	3x3	Double	local	280 B
A_aug	3x4	Double	local	304 B
L	3x3	Double	local	280 B
U	3x3	Double	local	280 B
a	0.667	Double	local	216 B
b	0.333	Double	local	216 B
c	-0.879	Double	local	216 B
i	1	Double	local	216 B
j	4	Double	local	216 B
k	3	Double	local	216 B
m	-3.67	Double	local	216 B
n	3.22	Double	local	216 B
sumk	5	Double	local	216 B
x	[1; 2; -1]	Double	local	232 B

Example: Solve the system of equations by decomposing A as a product A = LU
$2x + 3y + z = 8$, $4x + 7y + 5z = 20$, $-2y + 2z = 0$

Solution:

```
//2x + 3y + z = 8 , 4x + 7y + 5z = 20, -2y + 2z = 0
clear;close();clc;
format('v', 5);
A=[2,3,1;4,7,5;0,-2,2];
for l=1:3
    L(l,l)=1;
end
for i=1:3
    for j=1:3
        s=0;
        if j>=i
            for k=1:i-1
                s=s+L(i,k)*U(k,j);
            end
            U(i,j)=A(i,j)-s;
        else
            for k=1:j-1
                s=s+L(i,k)*U(k,j);
            end
            L(i,j)=(A(i,j)-s)/U(j,j);
        end
    end
end
b=[8;20;0];
```

c=L\b;

x=U\c;

disp(x,'Solution of the given equation is:')

```
//2x + 3y + z = 8 , 4x + 7y + 5z = 20, -2y + 2z
clear;close();clc;
format('v', 5);
A=[2,3,1;4,7,5;0,-2,2];
for l=1:3
    L(l,l)=1;
end
for i=1:3
    for j=1:3
        s=0;
        if j>=i
             for k=1:i-1
                 s=s+L(i,k)*U(k,j);
             end
             U(i,j)=A(i,j)-s;
        else
            for k=1:j-1
                s=s+L(i,k)*U(k,j);
            end
            L(i,j)=(A(i,j)-s)/U(j,j);
        end
    end

end
b=[8;20;0];
c=L\b;
x=U\c;
disp(x,'Solution of the given equation is:')
```

Output:

```
  2.
  1.
  1.

 "Solution of the given equation is:"

-->
```

	Name	Value	Type	Visibility	Memory
	A	3x3	Double	local	280 B
	L	3x3	Double	local	280 B
	U	3x3	Double	local	280 B
	b	[8; 20; 0]	Double	local	232 B
	c	[8; 4; 8]	Double	local	232 B
	i	3	Double	local	216 B
	j	3	Double	local	216 B
	k	2	Double	local	216 B
	l	3	Double	local	216 B
	s	-6	Double	local	216 B
	x	[2; 1; 1]	Double	local	232 B

<u>Problem Number - 3: The Gauss - Jordan method of calculating A^-1</u>

Procedure: To find the inverse of a non-singular matrix A, we begin with an augmented system [A I] where I is an identity matrix of the size same as that of A and reduce A to an upper triangular system by Gaussian Elimination. The system is further reduced to a diagonal one with pivots on the main diagonal and finally to [I A-1].

Example:

Find the inverse of the following matrix :

$$A=\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Solution:

```
clc; clear; close;

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

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

end

for j=1:n
```

Aug(j:1) = Aug(j:1)/Aug(j,j)

end

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

disp(B, "A inverse=")

```
clc; clear; close;
A = [1, 0, 0; 1, 1, 1; 0, 0, 1];
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(i:j-1, :) = Aug(i:j-1, :) - Aug(i:j-1, j)/Aug(j,j)*Aug(j,:);
end
for j=1:n
    Aug(j:1) = Aug(j:1)/Aug(j,j)
end
B = Aug(:, n+1:2*n);
disp(B, "A inverse=");
```

Output:

Scilab 6.1.0 Console

```
  1.   0.   0.
 -1.   1.   0.
  0.   0.   1.

 "A inverse="

-->
```

Variable Browser

	Name	Value	Type	Visibility	Memory
	A	3x3	Double	local	280 B
	Aug	3x6	Double	local	352 B
	B	3x3	Double	local	280 B
	i	3	Double	local	216 B
	j	3	Double	local	216 B
	n	3	Double	local	216 B

Problem Number 4: Span of the Column Space of A

Procedure: Given a matrix A, we reduce it to an upper triangular form using Gaussian Elimination. The columns that contain the pivots span the column space of A.

Example : Identify the columns that are in the column space of A where

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

Solution:

```
clc;clear;close;

disp('The given matrix is');

a=[2,4,6,4;2,5,7,6;2,3,5,2];

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

```
clc;clear;close;
disp('The given matrix is');
a=[2,4,6,4;2,5,7,6;2,3,5,2];
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
```

Output:

```
"The given matrix is"

 2.   4.   6.   4.
 0.   1.   1.   2.
 0.  -1.  -1.  -2.

 2.   4.   6.   4.
 0.   1.   1.   2.
 0.   0.   0.   0.

 1.   2.   3.   2.
 0.   1.   1.   2.
 0.   0.   0.   0.

"is a pivot column"

 1.

"column"

"is a pivot column"

 2.

"column"
```

Variable Browser

Name	Value	Type	Visibility	Memory
a	3x4	Double	local	304 B
i	3	Double	local	216 B
j	4	Double	local	216 B

Problem Number 5: The Four Fundamental Subspaces

Procedure: Given a matrix A, we reduce it to row reduced form and find its rank by identifying the columns that contain the pivots. We then find the four fundamental subspaces viz, the column space C(A), the row space C(AT), the null space N(A) and the left null space N(AT).

Example : Find the four fundamental subspaces of

$$A=\begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix}$$

Solution:

```
clear;

close;

clc;

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

disp('A=',A);

[m,n]=size(A);

disp('m=',m);

disp('n=',n);

[v,pivot]=rref(A);

disp(rref(A));

disp(v)

r = length(pivot);

disp("rank=",r);

cs=A(:,pivot);

disp("Column Space=",cs);

 ns= kernel(A);
```

```
disp("Null Space=",ns);

rs = v(1:r,:)';

disp("Row Space=", rs);

lns = kernel(A');

disp('Left Null Space = ', lns);
```

```
clear;
close;
clc;
A=[1,2,0,1;0,1,1,0;1,2,0,1];
disp('A=',A);
[m,n]=size(A);
disp('m=',m);
disp('n=',n);
[v,pivot]=rref(A);
disp(rref(A));
disp(v)
r = length(pivot);
disp("rank=",r);
cs=A(:,pivot);
disp("Column Space=",cs);
 ns= kernel(A);
 disp("Null Space=",ns);
rs = v(1:r,:)';
disp("Row Space=", rs);
lns = kernel(A');
disp('Left Null Space = ', lns);
```

Output:

```
"A="

 1.   2.   0.   1.
 0.   1.   1.   0.
 1.   2.   0.   1.

"m="

 3.

"n="

 4.

 1.   0.  -2.   1.
 0.   1.   1.   0.
 0.   0.   0.   0.

 1.   0.  -2.   1.
 0.   1.   1.   0.
 0.   0.   0.   0.

"rank="

 2.

"Column Space="

 1.   2.
 0.   1.
 1.   2.

"Null Space="

 3.909D-17  -0.8660254
-0.4082483   0.2886751
 0.4082483  -0.2886751
 0.8164966   0.2886751
```

```
"Row Space="

 1.   0.
 0.   1.
-2.   1.
 1.   0.

"Left Null Space = "

-0.7071068
 1.106D-16
 0.7071068
```

Variable Browser

Name	Value	Type	Visibility	Memory
A	3x4	Double	local	304 B
cs	3x2	Double	local	256 B
lns	[-0.707; 1.11e...	Double	local	232 B
m	3	Double	local	216 B
n	4	Double	local	216 B
ns	4x2	Double	local	272 B
pivot	[1, 2]	Double	local	224 B
r	2	Double	local	216 B
rs	4x2	Double	local	272 B
v	3x4	Double	local	304 B

Problem Number 6: Projections by Least Squares

Procedure: Suppose we do a series of experiments and expect the output b to be a linear function of the input t. We look for a straight line b = C + Dt. If there are experimental errors then we have a system of equations C + Dt1 = b1 C + Dt2 = b2 and so on. That is, we have the system of equations Ax = b. The best solution is obtained by minimizing the error E 2 = b − Ax 2 = (b1 − C − Dt1) 2 + (b2 − C − Dt2) 2 +..... + (bm − C − Dtm) 2

Example: : Find the solution x = (C , D) of the system Ax = b and the line of best fit C + Dt = b given

Solution:

```
clear;close;clc;

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

disp(A, 'A=');

b = [1; 1; 3;];

disp(b, 'b=');

x = (A'*A)\(A'*b);

disp(x,'x=');;

C = x(1, 1);

D = x(2, 1);

disp(C, 'C=');

disp(D, 'D=');

disp('The-line-of-best-fit-is-b=C+Dt');
```

```
clear;close;clc;
A = [1 -1; 1 1; 1 2];
disp(A, 'A=');
b = [1; 1; 3;];
disp(b, 'b=');
x = (A'*A)\(A'*b);
disp(x,'x=');;
C = x(1, 1);
D = x(2, 1);
disp(C, 'C=');
disp(D, 'D=');
disp('The-line-of-best-fit-is-b=C+Dt');
//end
```

Output:

```
 1.  -1.
 1.   1.
 1.   2.

"A="

 1.
 1.
 3.

"b="

 1.2857143
 0.5714286

"x="

 1.2857143

"C="

 0.5714286

"D="

"The-line-of-best-fit-is-b=C+Dt"
```

	Name	Value	Type	Visibility	Memory
	A	3x2	Double	local	256 B
	C	1.29	Double	local	216 B
	D	0.571	Double	local	216 B
	b	[1; 1; 3]	Double	local	232 B
	x	[1.29; 0.571]	Double	local	224 B

Problem Number 7: The Gram- Schmidt Orthogonalization

Procedure: Given a set of mutually independent vectors, we produce a set of orthonormal vectors by applying the Gram – Schmidt process.

Example: Apply the Gram – Schmidt process to the following set of vectors and find the orthogonal matrix: . (1, 1, 0) , (1, 0, 1) , (0, 1, 1)

Solution:

```
clear;close;clc;

A=[1 1 0;1 0 1;0 1 1];

disp('A=',A);

[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('Q=',V);
```

```
clear;close;clc;

A=[1 1 0;1 0 1;0 1 1];
//-- independent vectors stored in columns of A --//

disp('A=',A);
[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('Q=',V);
```

Output:

Scilab 6.1.0 Console

```
"A="

 1.   1.   0.
 1.   0.   1.
 0.   1.   1.

"Q="

 0.7071068   0.4082483  -0.5773503
 0.7071068  -0.4082483   0.5773503
 0.          0.8164966   0.5773503
```

Variable Browser

	Name	Value	Type	Visibility	Memory
	A	3x3	Double	local	280 B
	R	3x3	Double	local	280 B
	V	3x3	Double	local	280 B
	j	2	Double	local	216 B
	k	3	Double	local	216 B
	m	3	Double	local	216 B
	n	3	Double	local	216 B

Problem Number 8: Eigen values and Eigen vectors of a given square matrix

Procedure: Given a square matrix A, we find the characteristic polynomial of A by expanding the matrix equation | A – λ I |. The Eigen values of A are obtained solving the characteristic equation | A – λ I | = 0. The corresponding Eigen vectors are obtained by solving the system of equations Ax = λx.

Example: Find the Eigen values and the corresponding Eigen vectors of

$$A=\begin{pmatrix} 2 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \end{pmatrix}$$

Solution:

```
clc;close;clear;

A=[3,-2,5;-2,3,6;5,6,4];

lam=poly(0,'lam')

lam=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 of A are')

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
get f('eigenvectors')
[x,lam]=eigenvectors(A)
disp(x,'The eigen vector of A are')
```

```
clc;close;clear;
A=[2,2,1;1,3,1;1,2,2];
lam=poly(0,'lam')
lam=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 of A are')
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
//get f('eigenvectors')
[x,lam]=eigenvectors(A)
disp(x,'The eigen vector of A are')
```

Output:

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

"The characteristic Matrix is"

-5 +11lam -7lam² +lam³

"the characteristic polynomial is"

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

"eigen values of A are"

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

"The eigen vector of A are"

-->
```

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 - 0i, 5 - 0i, 1...	Double	local	256 B
	x	3x3	Double	local	280 B