

NAME: Kulvir Singh
Reg. No:19BCE2074

E-Record 2

Experiment:2A Properties of Eigen values and Eigen vectors, Cayley Hamilton theorem

Aim:

1. Let $A = \begin{pmatrix} 3 & 1 & -1 \\ 1 & 3 & -1 \\ -1 & -1 & 5 \end{pmatrix}$

- (a) Find characteristic equation of A (without using **poly** command).
- (b) Find eigen values by finding the roots of characteristic equation.
- (c) Find eigen vector X of A by solving the equation $AX = \lambda X$.
- (d) Verify the properties of Eigen values.
- (e) Verify Cayley-Hamilton theorem and find inverse

Mathematical Background:

Eigenvalues and Eigenvectors

We study the problem

$$AX = \lambda X$$

where A is given $n \times n$ square matrix, X is an unknown $n \times 1$ column vector, and λ is a scalar.
given an $n \times n$ matrix A , find the value of λ such that $[A - \lambda I]X = 0$ admits non-trivial solution, and find those non-trivial solution.

This is called the **Eigenvalue Problem**

Solving **characteristic equation** $|A - \lambda I| = 0$, we get n values of λ . These values are known as **eigenvalues**.
The vectors corresponding to each of these n values of λ are known as **eigenvectors**.

Properties of Eigenvalues

- 1) Any square matrix A and its transpose A^T have the same eigen values.
- 2) The eigenvalues of triangular matrix are just the diagonal elements of the matrix.
- 3) The eigenvalues of an idempotent matrix are either 0 or 1.
- 4) The sum of the eigen values of a matrix is the sum of the elements of the principal diagonal.
- 5) The product of the eigenvalues of a matrix A is equal to its determinant.
- 6) If λ is an eigenvalues of a matrix A , then $\frac{1}{\lambda}$ is the eigenvalue of A^{-1} .
- 7) If λ is an eigenvalue of an orthogonal matrix, then $\frac{1}{\lambda}$ is also its eigenvalue.
- 8) If $\lambda_1, \lambda_2, \dots, \lambda_n$ are the eigenvalues of a matrix, then A^m has the eigenvalues $\lambda_1^m, \lambda_2^m, \dots, \lambda_n^m$ (m being a positive integer).

Cayley-Hamilton Theorem

Every square matrix satisfies its own characteristic equation.

Matlab Code:

```
clc
clear all
close all
syms L real
A=input("Enter the square matrix:");
ch = det(A-L*eye(length(A)));
disp(ch);
r=solve(ch);
p=poly(A);
r1=roots([p]);
disp(r1);
for i=1:length(A)
    x=null(A-r(i)*eye(length(A)))
end
[p,d]=eig(A)
```

```

%%det of A = product of eigen values
det_A = det(A)%determinant of A
prod_A= prod(eig(A))%product of eigen values

%%trace of A = sum of the eigen values
trace_A = trace(A)
sum_A = sum(eig(A))

%%A and A^T have the same eigen values
eig_A=eig(A) %eigen values of A
eig_AT=eig(transpose(A)) %eigen values of A transpose

%%A-lambda and A^-1 (1/lambda)
inv_A=eig(inv(A))

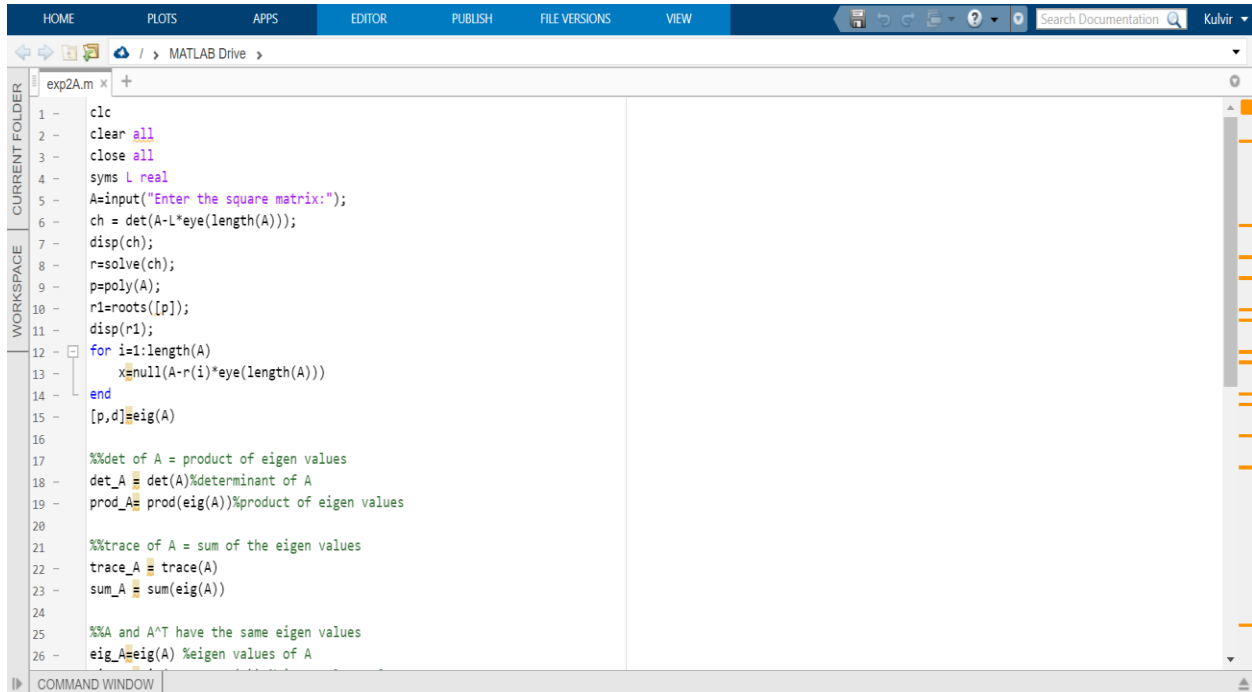
%%A-lambda and A^n -lambda^n
power_A=eig(A^2)

n=length(p);
cht= p(1)*A^(n-1);
for i=2:n
    cht=cht+p(i)*A^(n-i);
end
cht=round(cht);
disp(cht)

%inverse using C H theorem
inv_A=p(1)*A^(n-2);
for i=2:n-1
    inv_A=inv_A+p(i)*A^(n-i-1);
end
inv_A=inv_A/(-p(n))

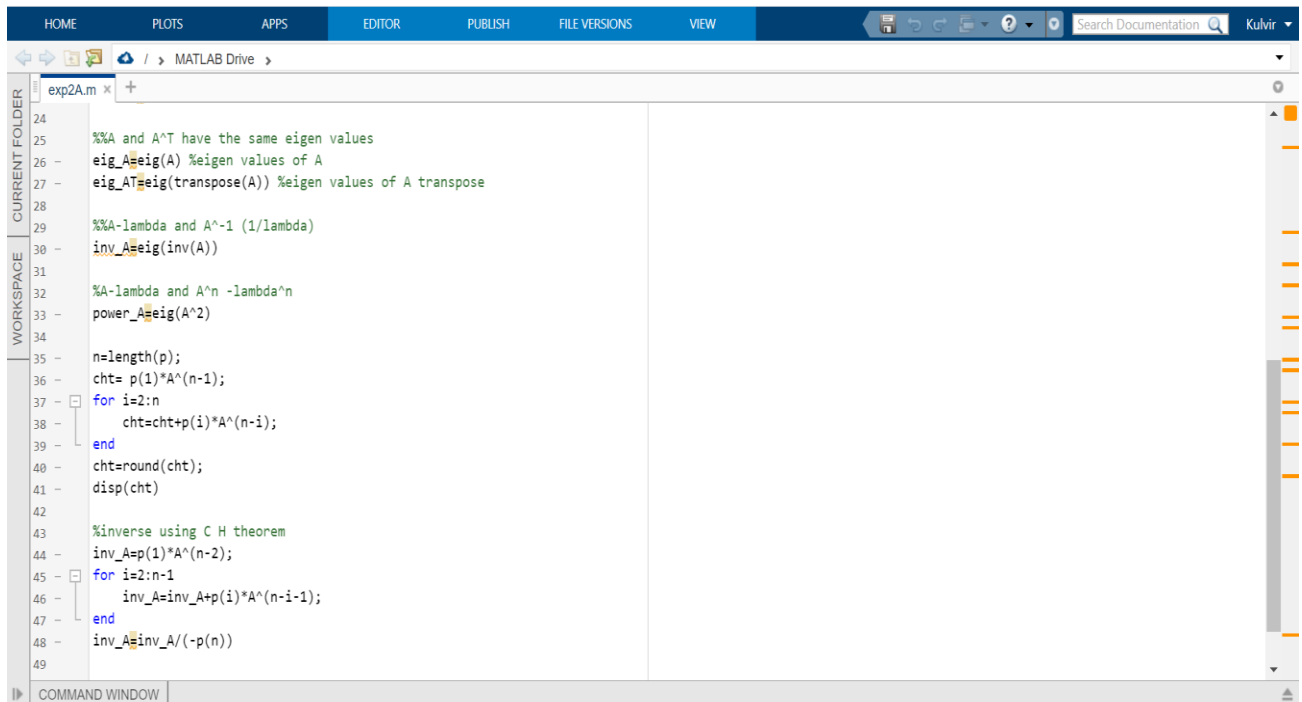
```

Code Screenshots:



This screenshot shows the first 26 lines of the MATLAB script 'exp2A.m'. The script starts with clearing the workspace and closing all figures. It prompts the user to enter a square matrix 'A'. Subsequent lines calculate the determinant of A using both the built-in 'det' function and a custom function 'det(A-L*eye(length(A)))'. It then solves for the roots of the characteristic polynomial, finds the null space for each root, and computes the eigenvalues and eigenvectors using 'eig(A)'. Finally, it calculates the determinant and trace of A using both direct and eigenvalue-based methods.

```
1 - clc
2 - clear all
3 - close all
4 - syms L real
5 - A=input("Enter the square matrix:");
6 - ch = det(A-L*eye(length(A)));
7 - disp(ch);
8 - r=solve(ch);
9 - p=poly(A);
10 - r1=roots([p]);
11 - disp(r1);
12 - for i=1:length(A)
13 -     x=null(A-r(i)*eye(length(A)))
14 - end
15 - [p,d]=eig(A)
16
17 %%det of A = product of eigen values
18 - det_A = det(A)%determinant of A
19 - prod_A= prod(eig(A))%product of eigen values
20
21 %%trace of A = sum of the eigen values
22 - trace_A = trace(A)
23 - sum_A = sum(eig(A))
24
25 %%A and A^T have the same eigen values
26 - eig_A=eig(A) %eigen values of A
```



This screenshot shows the continuation of the MATLAB script 'exp2A.m' from line 24 to 49. It calculates the eigenvalues of A and its transpose, and the eigenvalues of A inverse and its inverse transpose. It then calculates the inverse of A using the Cayley-Hamilton theorem. The script also calculates the power of A and the Cayley-Hamilton theorem polynomial.

```
24
25 %%A and A^T have the same eigen values
26 - eig_A=eig(A) %eigen values of A
27 - eig_AT=eig(transpose(A)) %eigen values of A transpose
28
29 %%A-lambda and A^-1 (1/lambda)
30 - inv_A=eig(inv(A))
31
32 %%A-lambda and A^n -lambda^n
33 - power_A=eig(A^2)
34
35 - n=length(p);
36 - cht= p(1)*A^(n-1);
37 - for i=2:n
38 -     cht=cht+p(i)*A^(n-i);
39 - end
40 - cht=round(cht);
41 - disp(cht)
42
43 %%inverse using C H theorem
44 - inv_A=p(1)*A^(n-2);
45 - for i=2:n-1
46 -     inv_A=inv_A+p(i)*A^(n-i-1);
47 - end
48 - inv_A=inv_A/(-p(n))
49
```

Output Screenshots:

The image displays two screenshots of the Kulvir software interface, which appears to be a mathematical or engineering tool. The interface includes a top navigation bar with tabs: HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. A search bar labeled 'Search Documentation' and the user name 'Kulvir' are also present.

Top Screenshot:

- CURRENT FOLDER:** Shows the input 'Enter the square matrix:' followed by a 3x3 matrix $\begin{bmatrix} 3 & 1 & -1 \\ 1 & 3 & -1 \\ -1 & -1 & 5 \end{bmatrix}$ and the expression $-L^3 + 11L^2 - 36L + 36$. Below this, the results 6.0000 , 3.0000 , and 2.0000 are displayed.
- WORKSPACE:** Shows two assignments for X . The first is $X = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$. The second is $X = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

Bottom Screenshot:

- CURRENT FOLDER:** Shows the assignment $X = \begin{bmatrix} -1/2 \\ -1/2 \\ 1 \end{bmatrix}$.
- WORKSPACE:** Shows three assignments:
 - $p = \begin{bmatrix} 0.7071 & 0.5774 & -0.4082 \\ -0.7071 & 0.5774 & -0.4082 \\ -0.0000 & 0.5774 & 0.8165 \end{bmatrix}$
 - $d = \begin{bmatrix} 2.0000 & 0 & 0 \\ 0 & 3.0000 & 0 \\ 0 & 0 & 6.0000 \end{bmatrix}$
 - $\det_A = 36$

HOME
PLOTS
APPS
EDITOR
PUBLISH
FILE VERSIONS
VIEW
Search Documentation
Kulvir

CURRENT FOLDER
prod_A =
36.0000
trace_A =
11
sum_A =
11.0000
eig_A =
2.0000
3.0000
6.0000
eig_AT =

COMMAND WINDOW
HOME
PLOTS
APPS
EDITOR
PUBLISH
FILE VERSIONS
VIEW
Search Documentation
Kulvir

CURRENT FOLDER
eig_AT =
2.0000
3.0000
6.0000
inv_A =
0.1667
0.3333
0.5000
power_A =
4.0000
9.0000
36.0000
6 4 -6
4 6 -6
-6 -6 16

HOME
PLOTS
APPS
EDITOR
PUBLISH
FILE VERSIONS
VIEW
Search Documentation
Kulvir

CURRENT FOLDER
0.1667
0.3333
0.5000
power_A =
4.0000
9.0000
36.0000
6 4 -6
4 6 -6
-6 -6 16
inv_A =
1.0e+16 *
0.6005 0.3002 -0.3002
0.3002 0.6005 -0.3002
-0.3002 -0.3002 1.2010

COMMAND WINDOW

Kulvir Singh
19BCE2074

Experiment:2B Diagonalization by similarity transformation, Orthogonal Transformation

Aim:

1. Diagonalize $A = \begin{pmatrix} 1 & 6 & 1 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$ by similarity transformation.
2. Diagonalize $A = \begin{pmatrix} 3 & 2 & 4 \\ 2 & 0 & 2 \\ 4 & 2 & 3 \end{pmatrix}$ by orthogonal transformation.

Mathematical Background:

Similarity Transformation

A is said to be **similar** to B if there exist a non-singular matrix P such that

$$B = P^{-1}AP$$

This transformation of A to B is known as **similarity transformation**.

Let X_1, X_2, \dots, X_n be the n linearly independent eigenvectors of A corresponding to n eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$. $P_{n \times n} = [X_1 \ X_2 \ \dots \ X_n]$ is known as **modal matrix**.

$$\begin{aligned} A P &= A [X_1 \ X_2 \ \dots \ X_n] \\ &= [AX_1 \ AX_2 \ \dots \ AX_n] \\ &= [\lambda_1 X_1 \ \lambda_2 X_2 \ \dots \ \lambda_n X_n] \\ &= [X_1 \ X_2 \ \dots \ X_n] \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \\ &= P D \end{aligned}$$

Multiplying by P^{-1} on both sides,

$$P^{-1}AP = (P^{-1}P)D = D$$

where D is the diagonal matrix with eigen values of A as the principal diagonal elements. D is known as **spectral matrix**.

Orthogonal Transformation

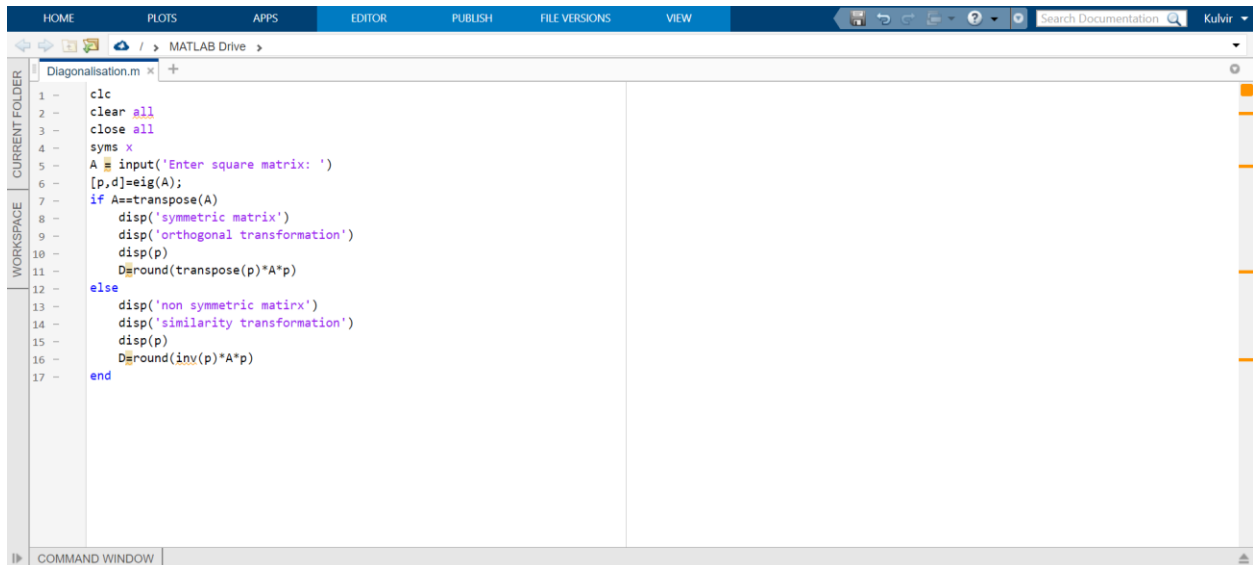
If we normalise each eigen vector and use them to form the normalised modal matrix N then it can be proved that N is an **orthogonal matrix**.

The similarity transformation $P^{-1}AP = D$ takes the form $N^T AN = D$ since $N^{-1} = N^T$ by a property of orthogonal matrix. Transforming A into D by means of the transformation $N^T AN = D$ is called as **orthogonal reduction** or **orthogonal transformation**.

Matlab Code:

```
clc
clear all
close all
syms x
A = input('Enter square matrix: ')
[p,d]=eig(A);
if A==transpose(A)
    disp('symmetric matrix')
    disp('orthogonal transformation')
    disp(p)
    D=round(transpose(p)*A*p)
else
    disp('non symmetric matrix')
    disp('similarity transformation')
    disp(p)
    D=round(inv(p)*A*p)
end
```

Code Screenshot:



Output

1> Similarity Transformation

```
HOME PLOTS APPS EDITOR PUBLISH FILE VERSIONS VIEW Search Documentation Kulvir
Enter square matrix:
[1 6 1;1 2 0;0 0 3]

A =

    1    6    1
    1    2    0
    0    0    3

non symmetric matrix
similarity transformation
-0.9487 -0.8944 -0.2357
 0.3162 -0.4472 -0.2357
         0         0  0.9428

D =

   -1    0    0
    0    4    0
    0    0    3

>> |
```

2> Orthogonal Transformation

```
HOME PLOTS APPS EDITOR PUBLISH FILE VERSIONS VIEW Search Documentation Kulvir
Enter square matrix:
[3 2 4;2 0 2;4 2 3]

A =

    3    2    4
    2    0    2
    4    2    3

symmetric matrix
orthogonal transformation
-0.4862 -0.5649  0.6667
-0.4834  0.8095  0.3333
 0.7279  0.1602  0.6667

D =

   -1    0    0
    0   -1    0
    0    0    8

>> |
```