

Department of Mathematics
School of Advanced Sciences
MAT 2002 – Applications of Differential and Difference Equations(MATLAB)
Experiment 1-A
Fourier Series

Fourier Series: The **Fourier Series** of a periodic function $f(x)$ of period $2l$ defined on the interval $(\alpha, \alpha + 2l)$ is given by $f(x) \approx \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi x}{l}\right) + b_n \sin\left(\frac{n\pi x}{l}\right)$, where the

Fourier coefficients $a_0, a_n, b_n, n = 0, 1, 2, \dots$ can be evaluated by the following formulae

$$a_0 = \frac{1}{l} \int_{\alpha}^{\alpha+2l} f(x) dx,$$

$$a_n = \frac{1}{l} \int_{\alpha}^{\alpha+2l} f(x) \cos\left(\frac{n\pi x}{l}\right) dx,$$

$$b_n = \frac{1}{l} \int_{\alpha}^{\alpha+2l} f(x) \sin\left(\frac{n\pi x}{l}\right) dx.$$

In the Fourier series the term $(a_1 \cos x + b_1 \sin x)$ is called the Fundamental Harmonic/First Harmonic, $(a_2 \cos 2x + b_2 \sin 2x)$ is called the Second Harmonic and so on.

MATLAB Syntax Used:

<code>syms var1 var2</code>	Creates symbolic variables <code>var1</code> and <code>var2</code>
<code>disp(x)</code>	Displays the contents of <code>x</code> without printing the variable name
<code>int(expr, var, a, b)</code>	Evaluates the definite integral of <code>expr</code> with respect to <code>var</code> from <code>a</code> to <code>b</code> .
<code>ezplot(fun, [xmin, xmax])</code>	Plot the function <code>fun</code> over the domain <code>(xmin, xmax)</code> .

MATLAB Code:

```

clear all
close all
clc
syms x
f =input('Enter the function of x: ');
I=input('Enter the interval of [a,b]: ');
m=input('Enter the number of Harmonics required: ');
a=I(1);b=I(2);
L=(b-a)/2;
a0=(1/L)*int(f,a,b);
Fx=a0/2;
for n=1:m
figure;
an(n)=(1/L)*int(f*cos(n*pi*x/L),a,b);
bn(n)=(1/L)*int(f*sin(n*pi*x/L),a,b);
Fx=Fx+an(n)*cos(n*pi*x/L)+bn(n)*sin(n*pi*x/L);
Fx=vpa(Fx,4);
ezplot(Fx,[a,b]);

```

```

hold on
ezplot(f,[a,b]);
title(['Fourier Series with ',num2str( n ),'harmonics']);
legend('Fourier Series', 'Function Plot');
hold off
end
disp(strcat('Fourier series with', num2str(n), 'harmonics
is:',char(Fx)))

```

Example: To find the Fourier series expansion of the function $f(x) = x - x^2$, $-\pi < x < \pi$ upto 3 harmonics.

Input:

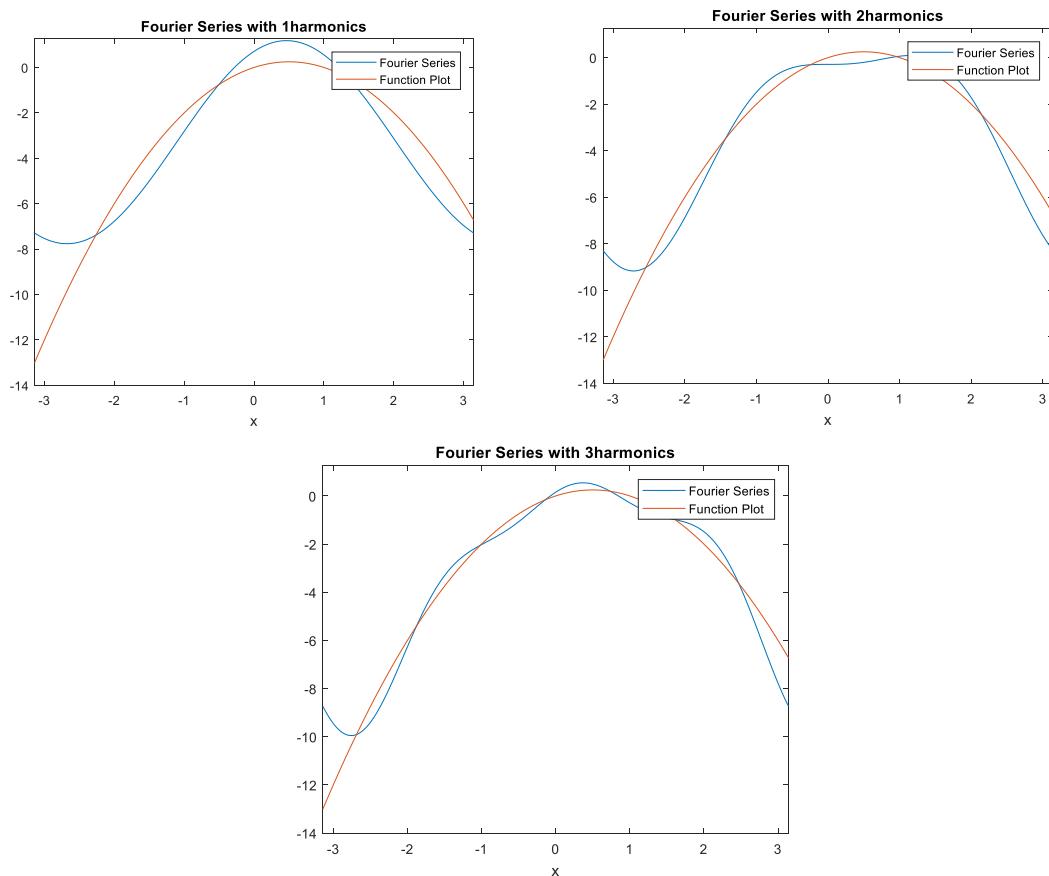
Enter the function of x: x-x^2

Enter the interval of [a,b]: [-pi,pi]

Enter the number of Harmonics required: 3

Output:

Fourier series with 3 harmonics is: $0.4444\cos(3.0*x) - 1.0*\sin(2.0*x)$
 $- 1.0*\cos(2.0*x) + 0.6667*\sin(3.0*x)$



Example: To find the Fourier series expansion up to 5 harmonics for the function

$$f(x) = \begin{cases} 1 + \frac{2x}{\pi}; & -\pi < x \leq 0 \\ 1 - \frac{2x}{\pi}; & 0 \leq x < \pi \end{cases}$$

Input:

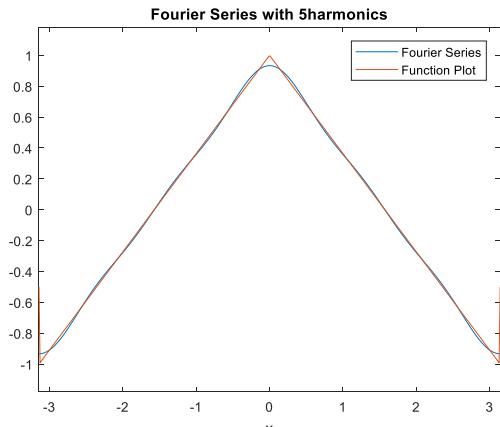
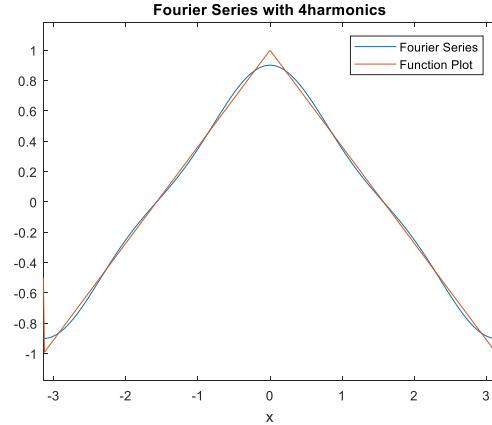
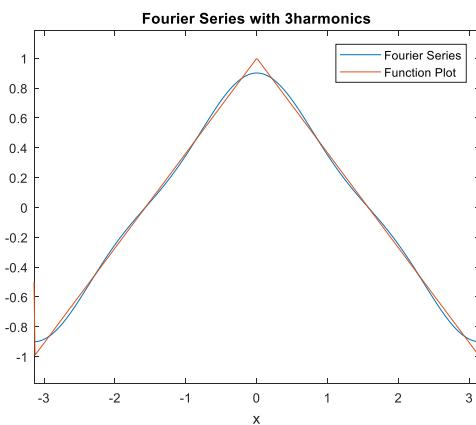
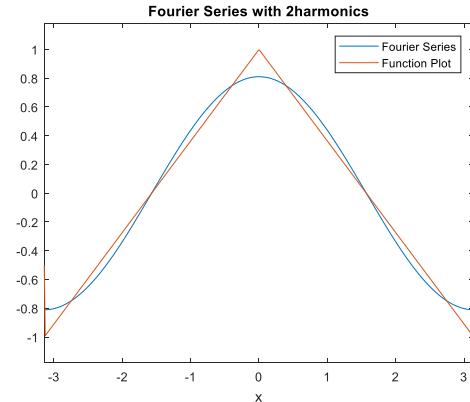
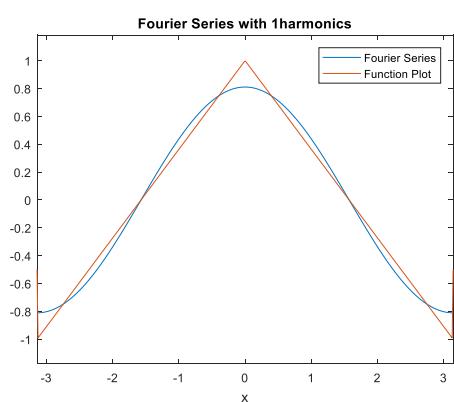
Enter the function of x: $(1+2*x/pi) * (\text{heaviside}(x+\pi) - \text{heaviside}(x)) + (1-2*x/pi) * (\text{heaviside}(x) - \text{heaviside}(x-\pi))$

Enter the interval of [a,b]: [-pi,pi]

Enter the number of Harmonics required: 5

Output:

Fourier series with 5 harmonics is: $0.03242\cos(5.0*x) + 0.09006\cos(3.0*x) + 0.8106\cos(x)$



Exercise:

1. Find the Fourier series expansion of the following functions:
 - a) $f(x) = \begin{cases} -1; & -2 < x < 0 \\ 1; & 0 < x < 2 \end{cases}, f(x+4) = f(x)$
 - b) $f(x) = \begin{cases} 0; & -2 < x < 0 \\ 4; & 0 < x < 2 \end{cases}, f(x+4) = f(x)$
 - c) $f(x) = e^{-x}$ in the interval $0 < x < 2\pi$, given that $f(x+2\pi) = f(x)$.
2. A sinusoidal voltage $E \sin \omega t$, where t is time, is passed through a half-wave rectifier that clips the negative portion of the wave. Find the Fourier series of the resulting periodic function $f(t) = \begin{cases} 0; & -\pi/\omega < t < 0 \\ E \sin \omega t; & 0 < t < \pi/\omega \end{cases}, f(t+2\pi/\omega) = f(t)$, with $E = 5$, $\omega = 2\pi$.

MAT2002 – Applications of Differential & Difference Equations

Fourier Series

(Exp – 1A 1B)

Faculty Name -
Monica C

Presented By -
Kumar Sparsh
19BCB0025

1. Find the Fourier series expansion of the following functions:

a) $f(x) = \begin{cases} -1; & -2 < x < 0 \\ 1; & 0 < x < 2 \end{cases}, f(x+4) = f(x)$

Sol.-

1 (a)

The screenshot shows the MATLAB desktop interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. The EDITOR tab is active, showing a script named 'exp_1a.m'. The code in the editor is as follows:

```

1 %EXP_1A
2 clear all
3 close all
4 clc
5 syms x
6 f =input('Enter the function of x: ');
7 I=input('Enter the interval of [a,b]: ');
8 m=input('Enter the number of Harmonics required: ');
9 a=I(1);b=I(2);
10 L=(b-a)/2;
11 a0=(1/L)*int(f,a,b);
12 Fx=a0/2;
13 for n=1:m
14 figure;
15 an(n)=(1/L)*int(f*cos(n*pi*x/L),a,b);
16 bn(n)=(1/L)*int(f*sin(n*pi*x/L),a,b);
17 Fx=Fx+an(n)*cos(n*pi*x/L)+bn(n)*sin(n*pi*x/L);
18 Fx=vpa(Fx,4);
19 ezplot(Fx,[a,b]);
20 hold on
21 ezplot(f,[a,b]);
22 title(['Fourier Series with ',num2str(n),'harmonics']);
23 legend('Fourier Series', 'Function Plot');
24 hold off
25 end
26 disp(strcat('Fourier series with', num2str(n),'harmonics is:',char(Fx)))

```

The bottom status bar indicates 'COMMAND WINDOW'.

Figure 1: MATLAB Code

The screenshot shows the MATLAB Command Window. The top menu bar includes HOME, PLOTS, APPS, FIGURE, FILE, ANNOTATIONS, TOOLS, and EDIT. The workspace contains several files: rough.m, untitled.m, Untitled3.m, Untitled4.m, exp_2a_4cm, untitled2.m, exp_2b_5m, exp_1a.m, Figure 1, Figure 2, Figure 3, and Figure 4. The command window displays the following text:

```

Enter the function of x:
-1*(heaviside(x+2)-heaviside(x)) + 1*(heaviside(x)-heaviside(x-2))
Enter the interval of [a,b]:
[-2,2]
Enter the number of Harmonics required:
4
Fourier series with4harmonics is:0.4244*sin(4.712*x) + 1.273*sin(1.571*x)
>>

```

Figure 2 : 1(a) Command Window output

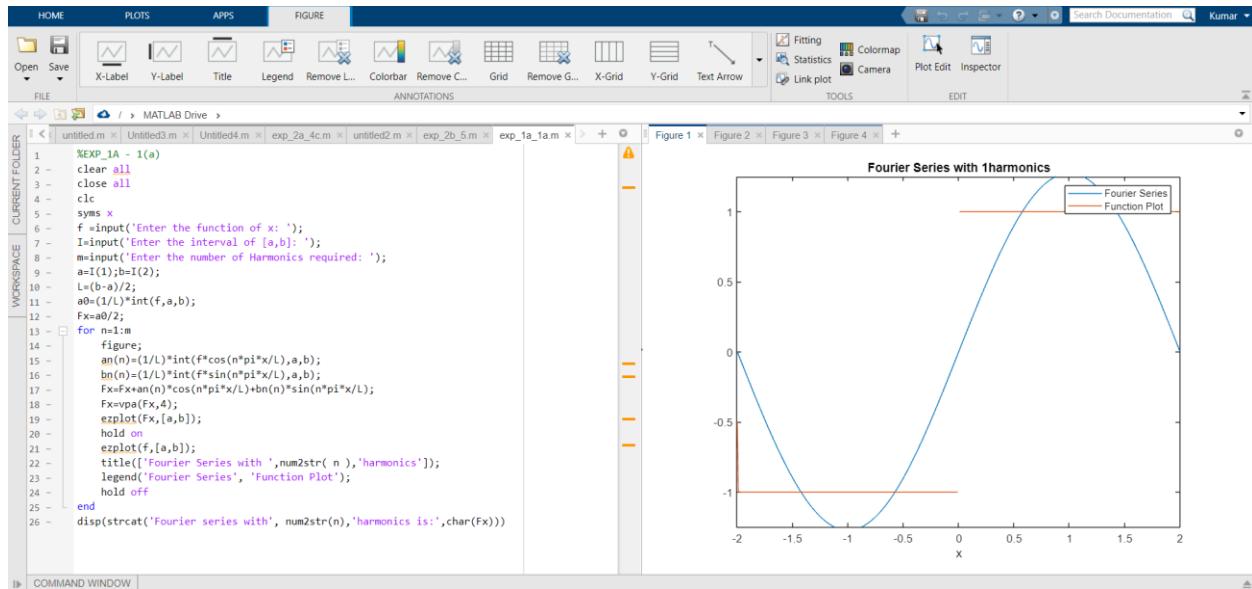


Figure 3 : 1(a) Fourier Series with 1 harmonic

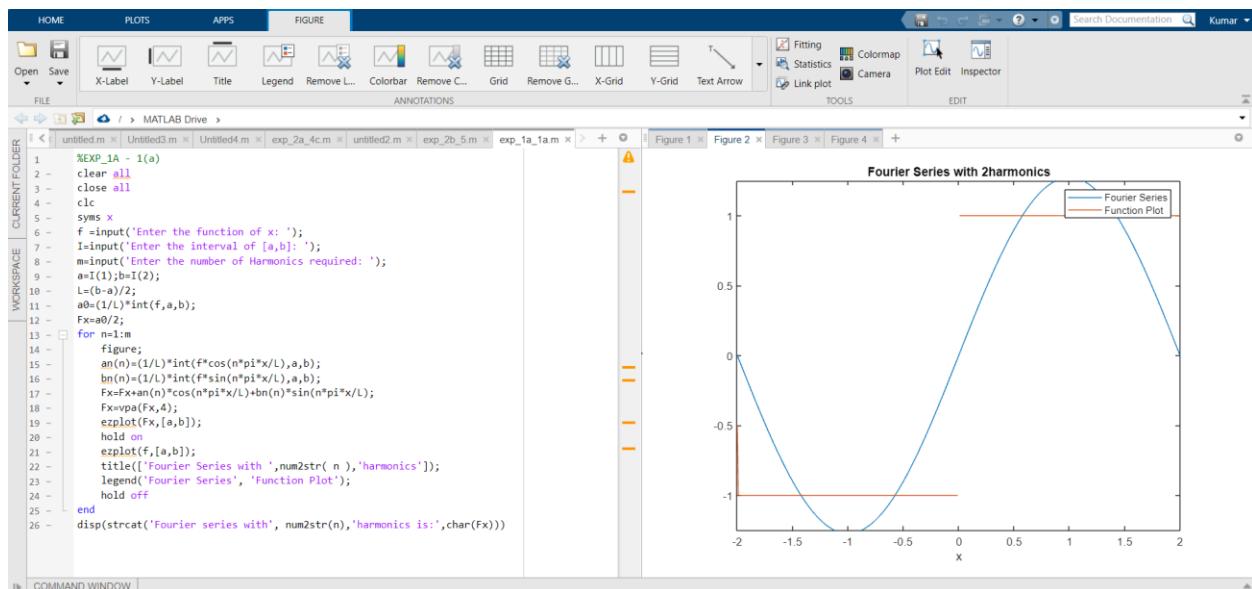


Figure 4 : 1(a) Fourier Series with 2 harmonics

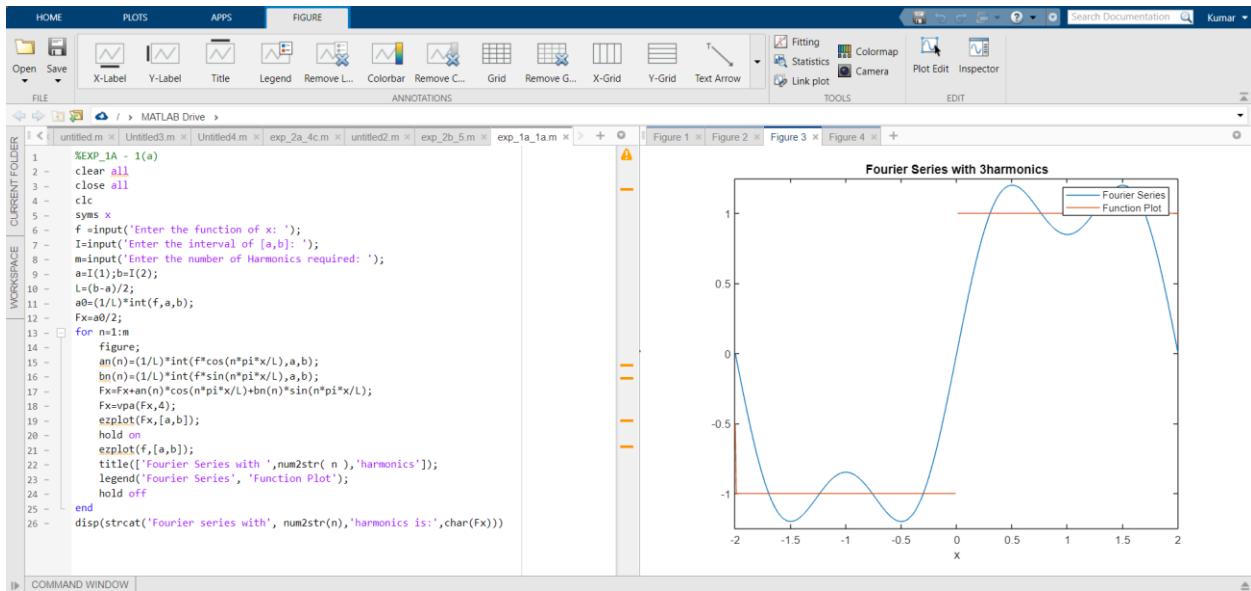


Figure 5 : 1(a) Fourier Series with 3 harmonics

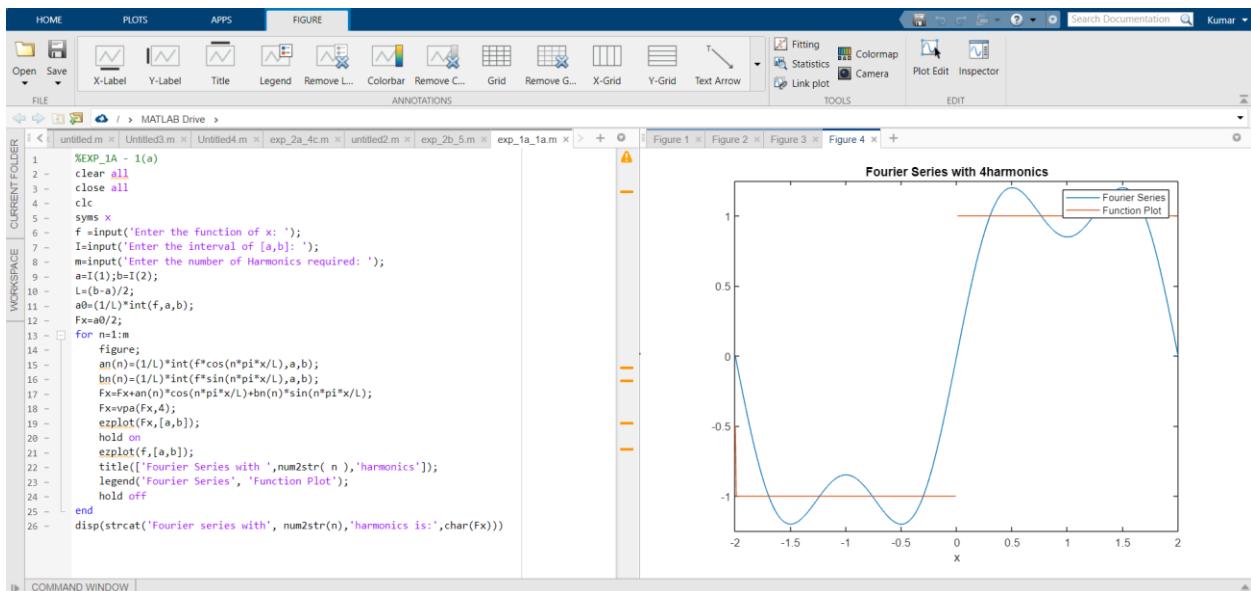


Figure 6 : 1(a) Fourier Series with 4 harmonics

1(b)

$$\text{b) } f(x) = \begin{cases} 0; & -2 < x < 0 \\ 4; & 0 < x < 2 \end{cases}, \quad f(x+4) = f(x)$$

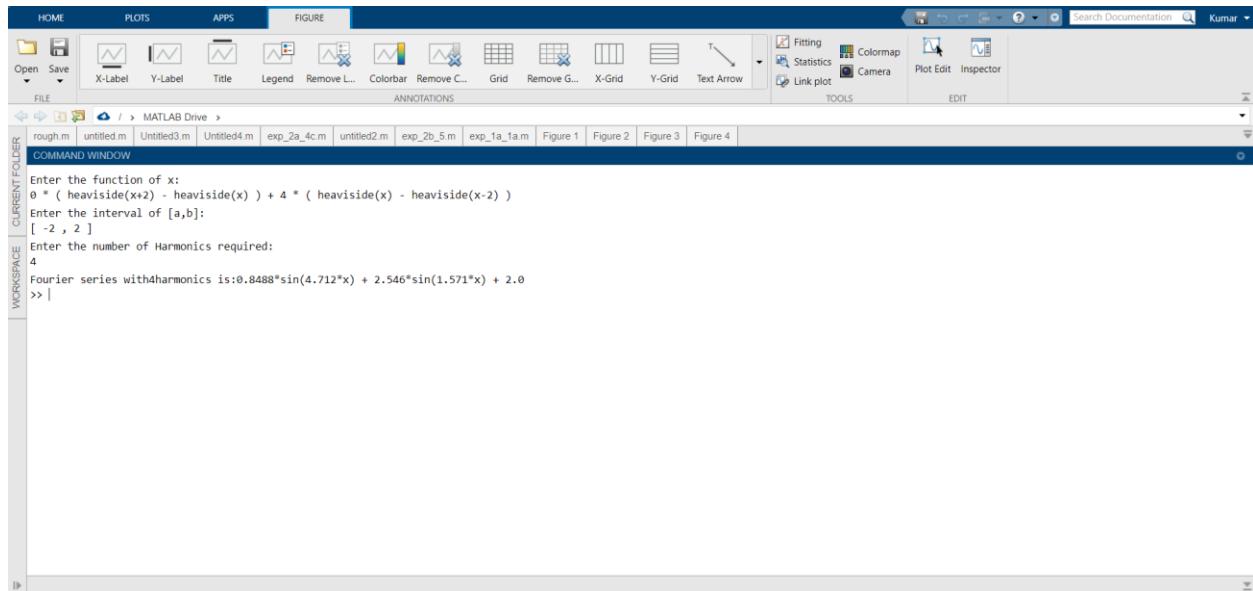


Figure 7 : 1(b) Command window output

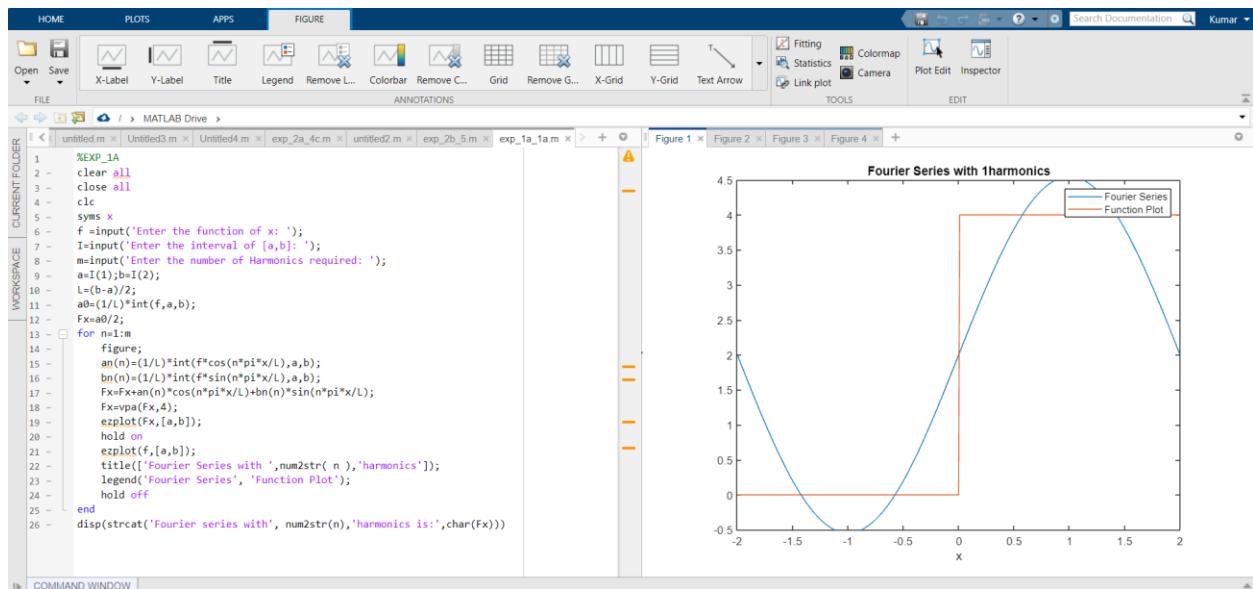


Figure 8 : 1(b) Fourier Series with 1 harmonic

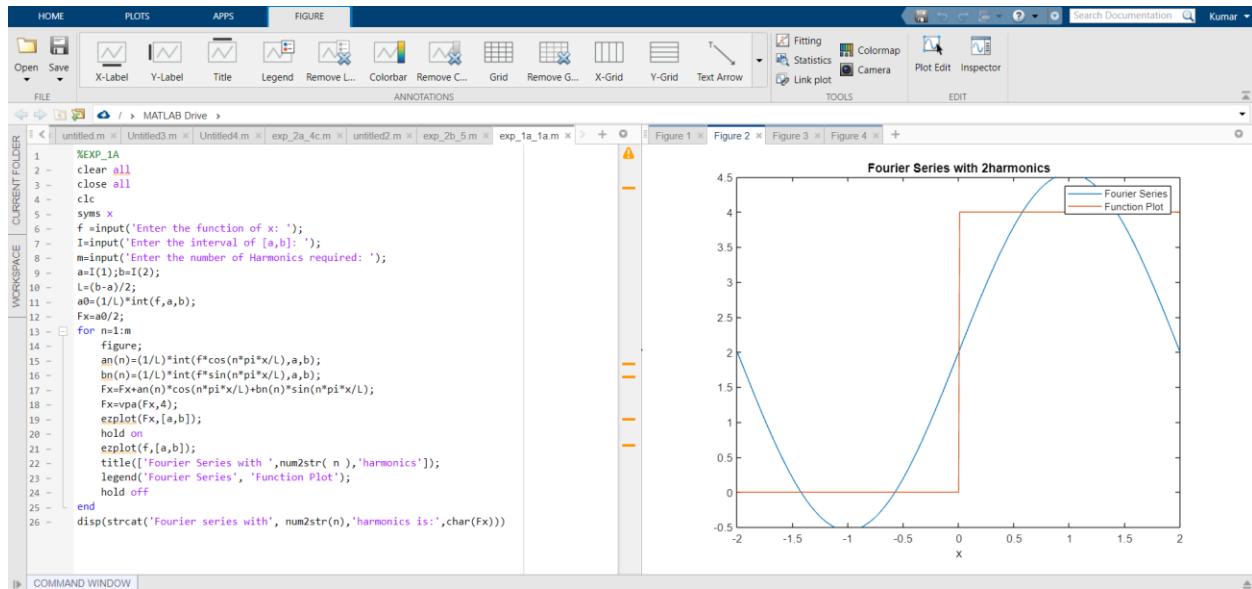


Figure 9 : 1(b) Fourier Series with 2 harmonics

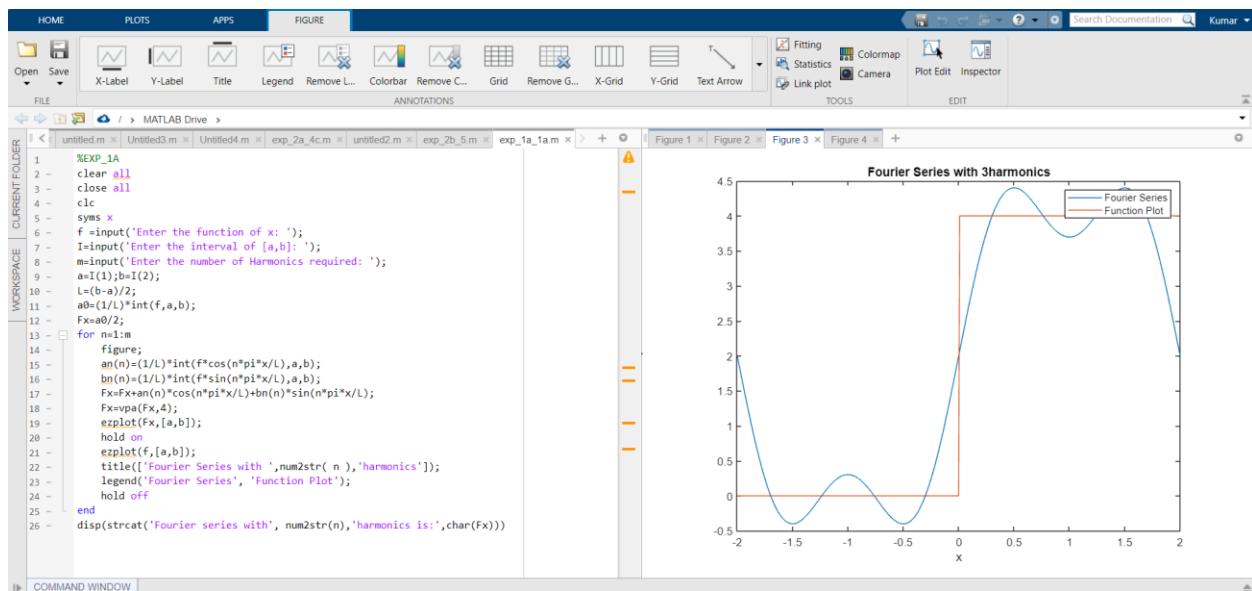


Figure 10 : 1(b) Fourier Series with 3 harmonics

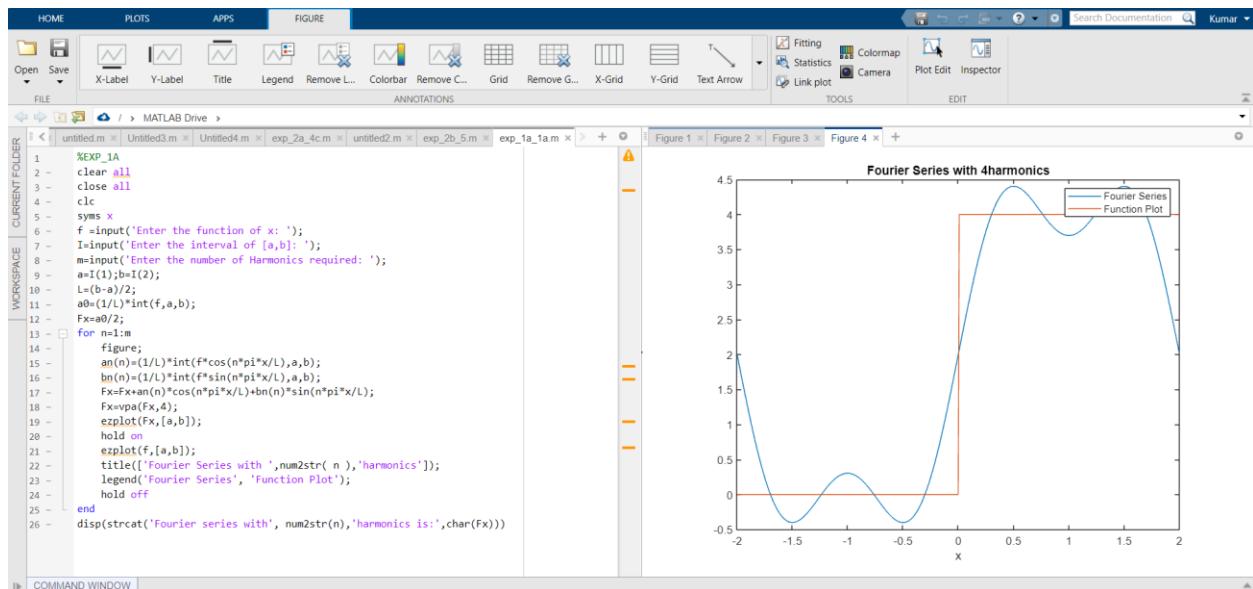


Figure 11 : 1(b) Fourier Series with 4 harmonics

1(c)

c) $f(x) = e^{-x}$ in the interval $0 < x < 2\pi$, given that $f(x+2\pi) = f(x)$.

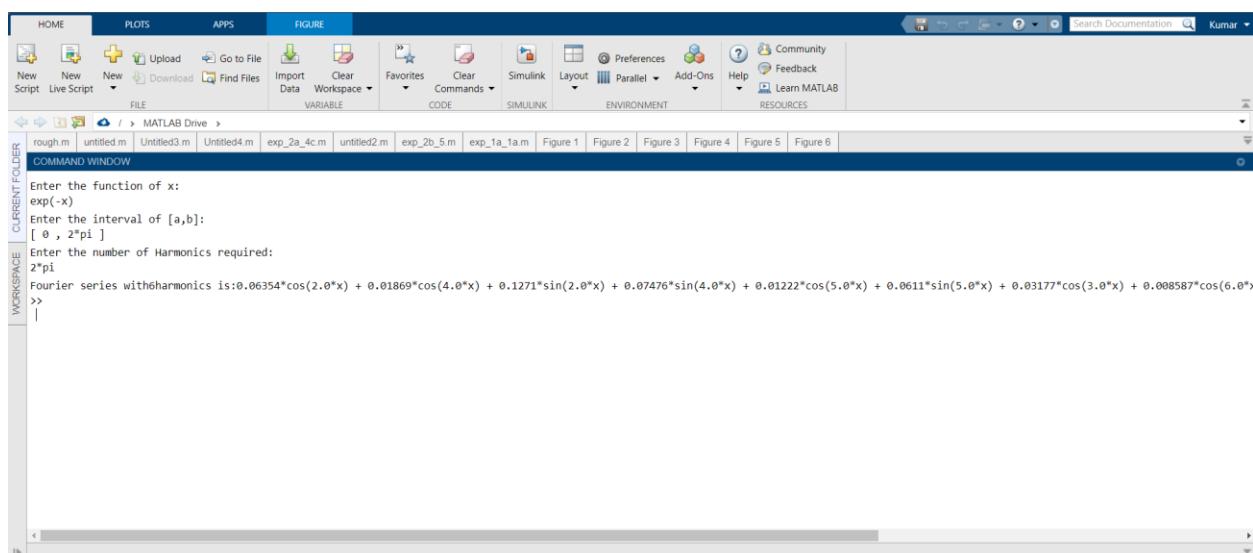


Figure 12 : 1(c) Command window output

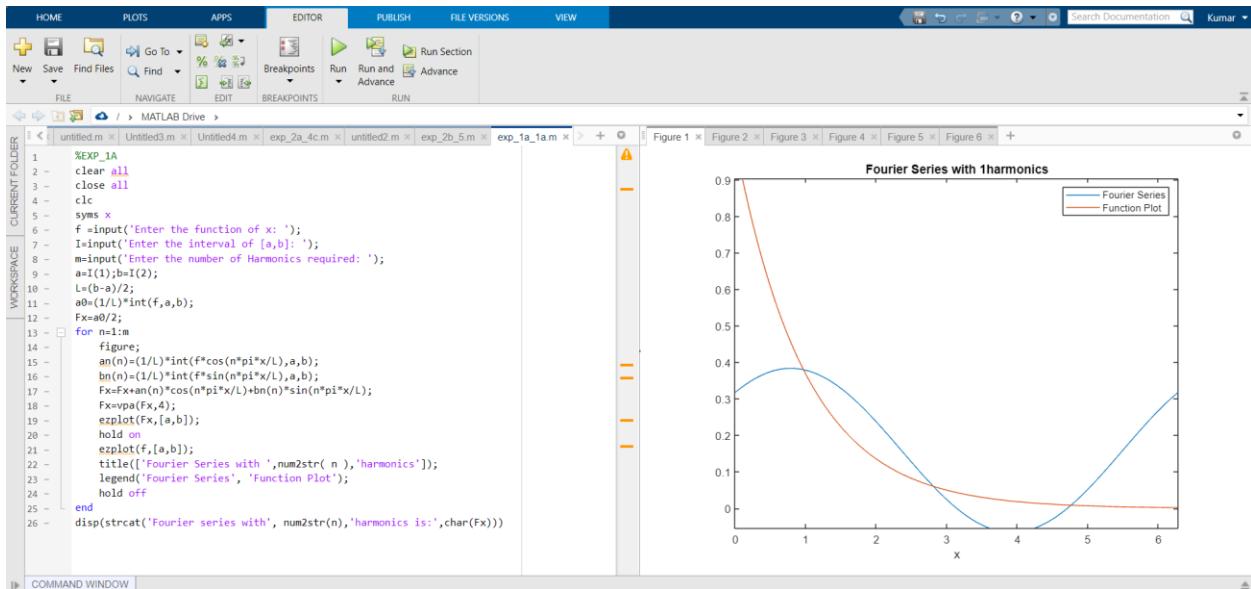


Figure 13 : 1(c) Fourier Series with 1 harmonic

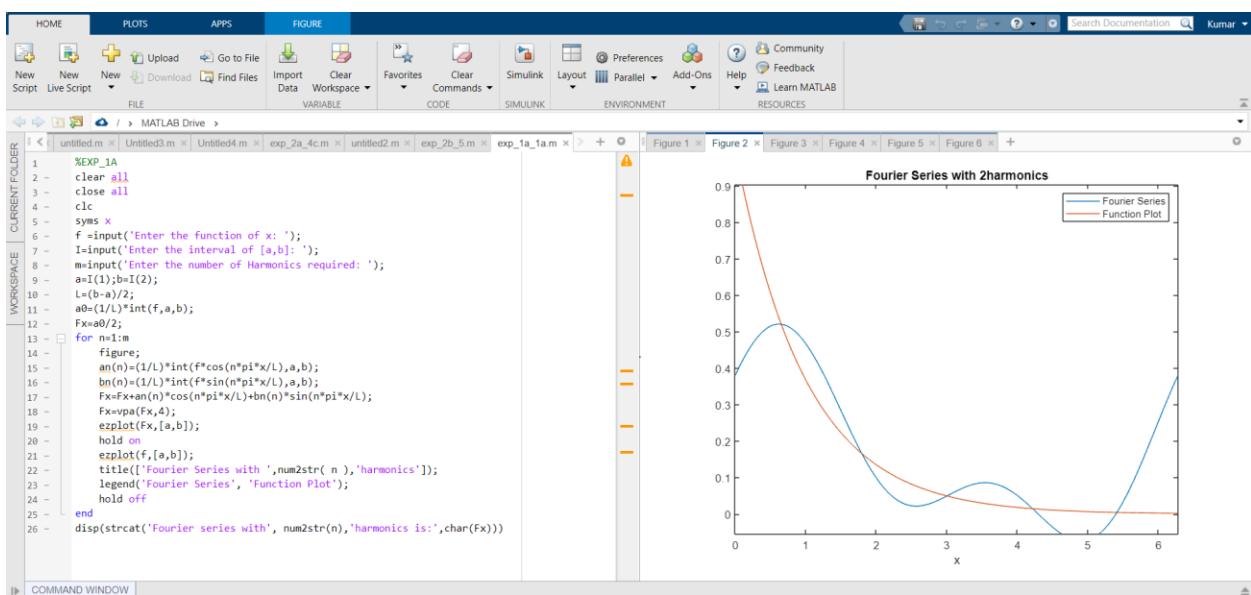


Figure 14 : 1(c) Fourier Series with 2 harmonics

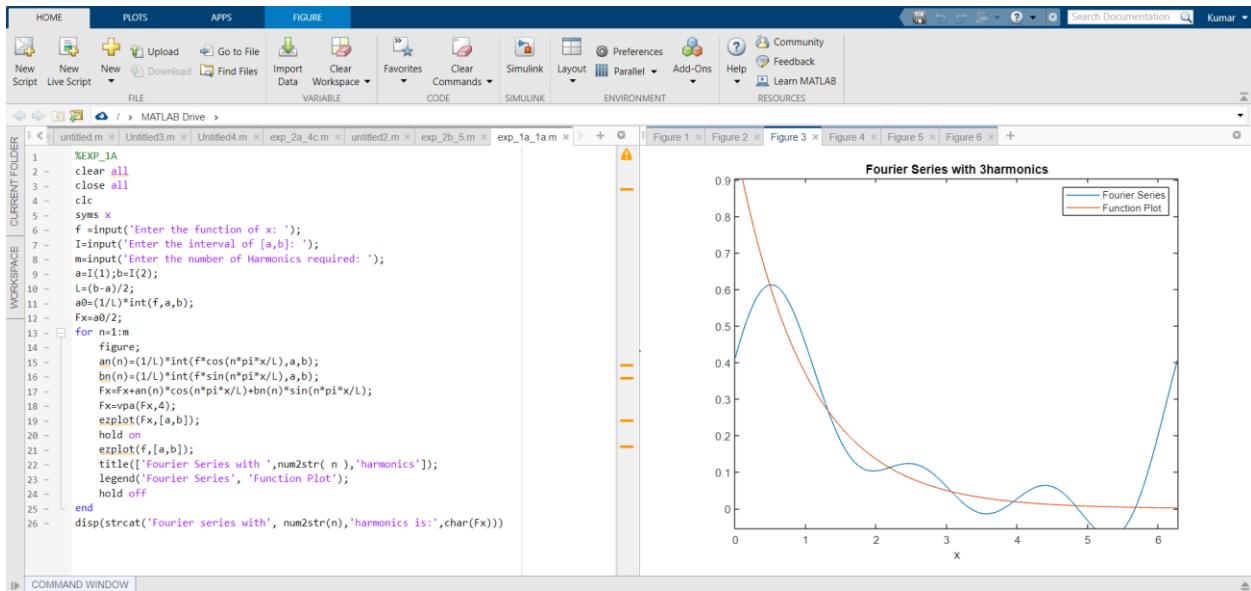


Figure 15 : 1(c) Fourier Series with 3 harmonics

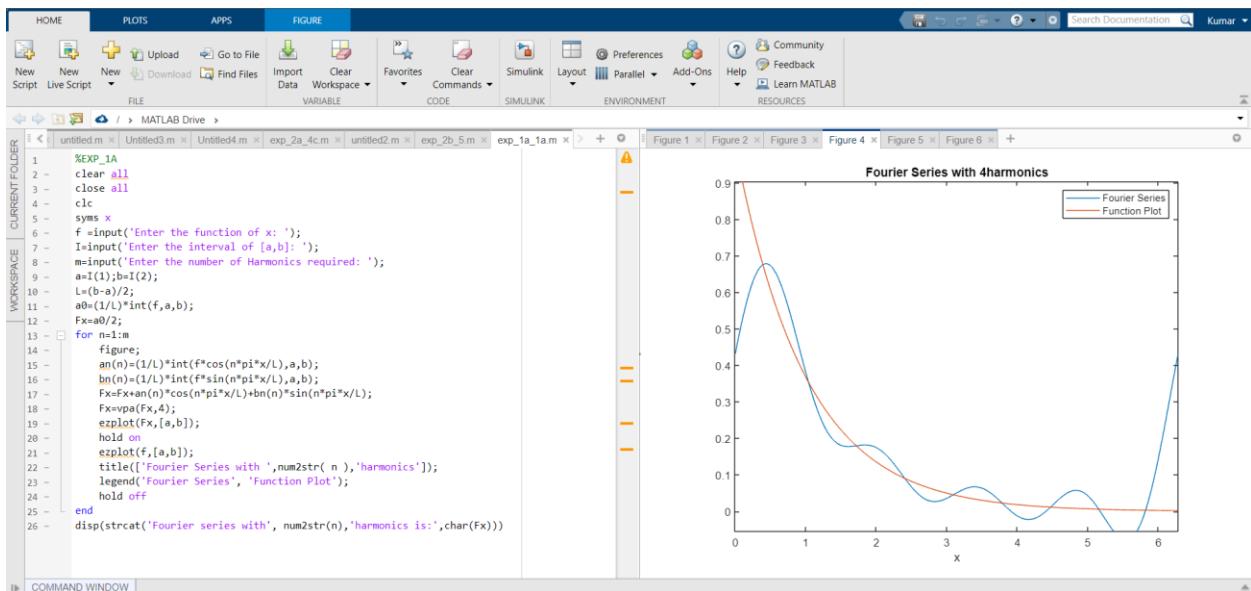


Figure 16 : 1(c) Fourier Series with 4 harmonics

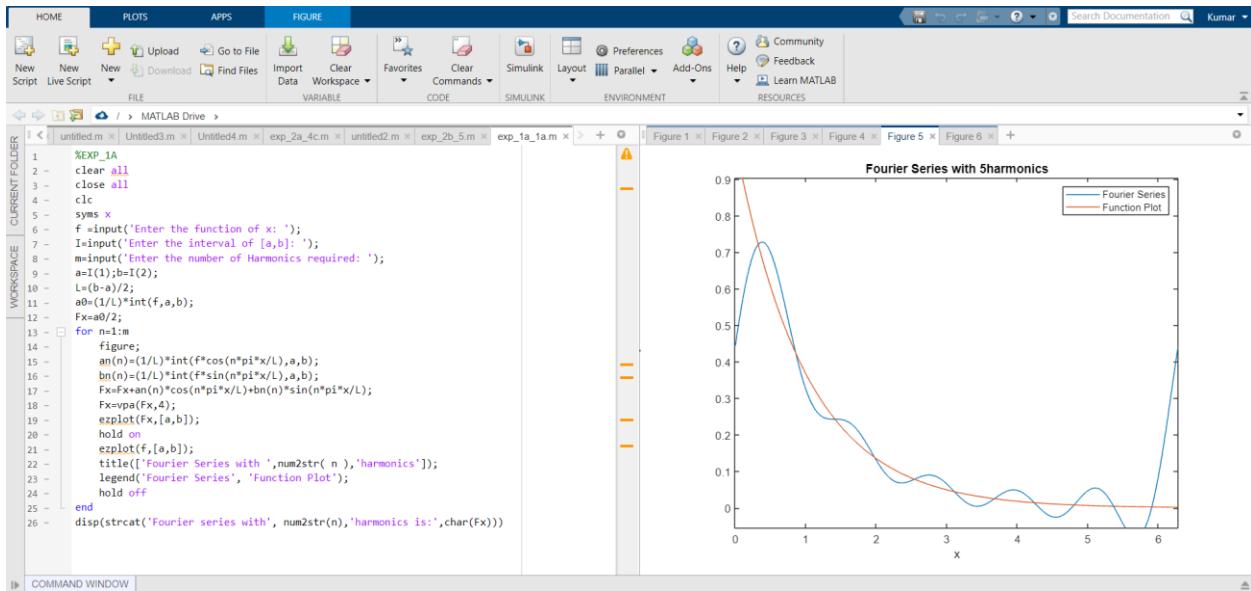


Figure 17 : 1(c) Fourier Series with 5 harmonics

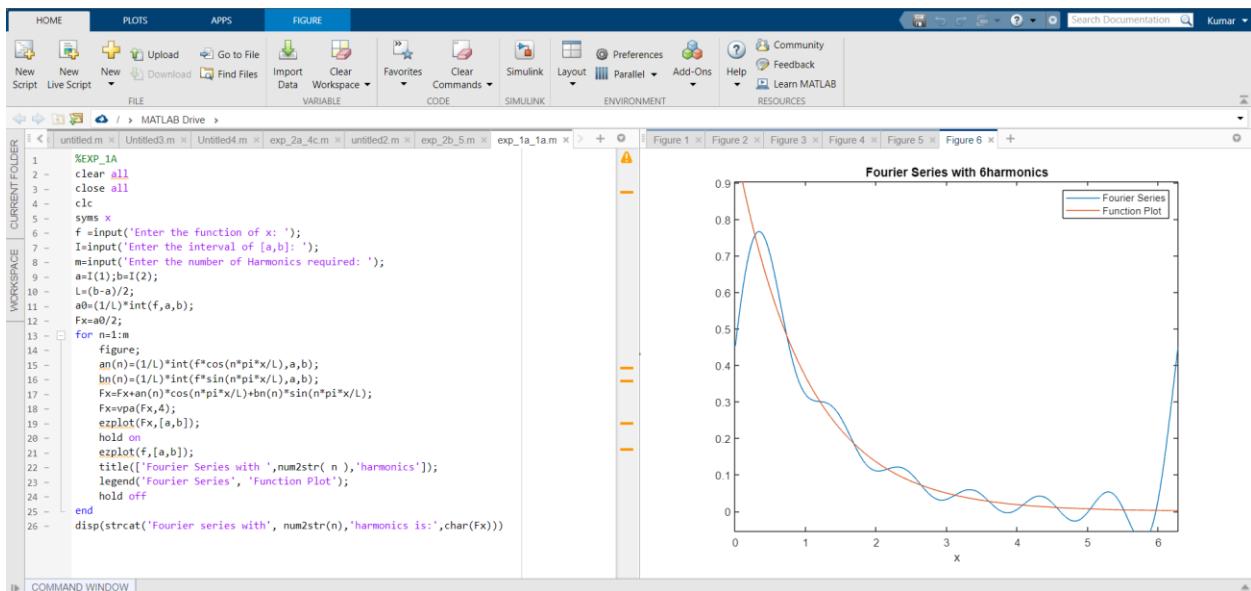


Figure 18 : 1(c) Fourier Series with 6 harmonics

2. A sinusoidal voltage $E \sin \omega t$, where t is time, is passed through a half-wave rectifier that clips the negative portion of the wave. Find the Fourier series of

$$\text{the resulting periodic function } f(t) = \begin{cases} 0; -\pi/\omega < t < 0 \\ E \sin \omega t; 0 < t < \pi/\omega \end{cases},$$

$$f(t+2\pi/\omega) = f(t), \text{ with } E = 5, \omega = 2\pi.$$

Sol.-

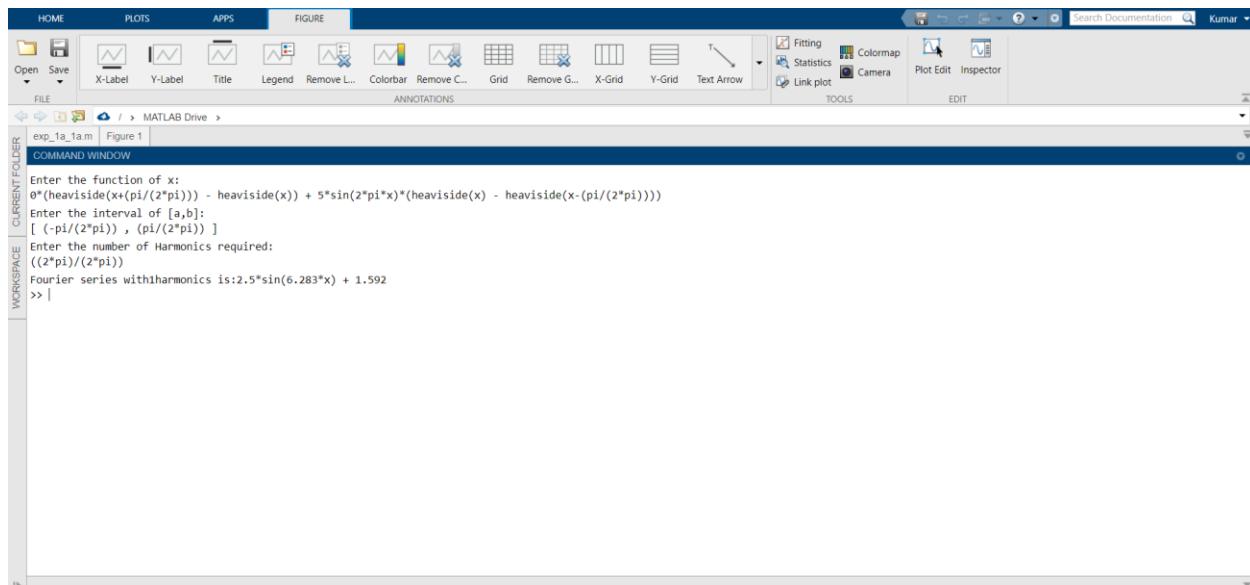


Figure 19 : 2 - Command window output

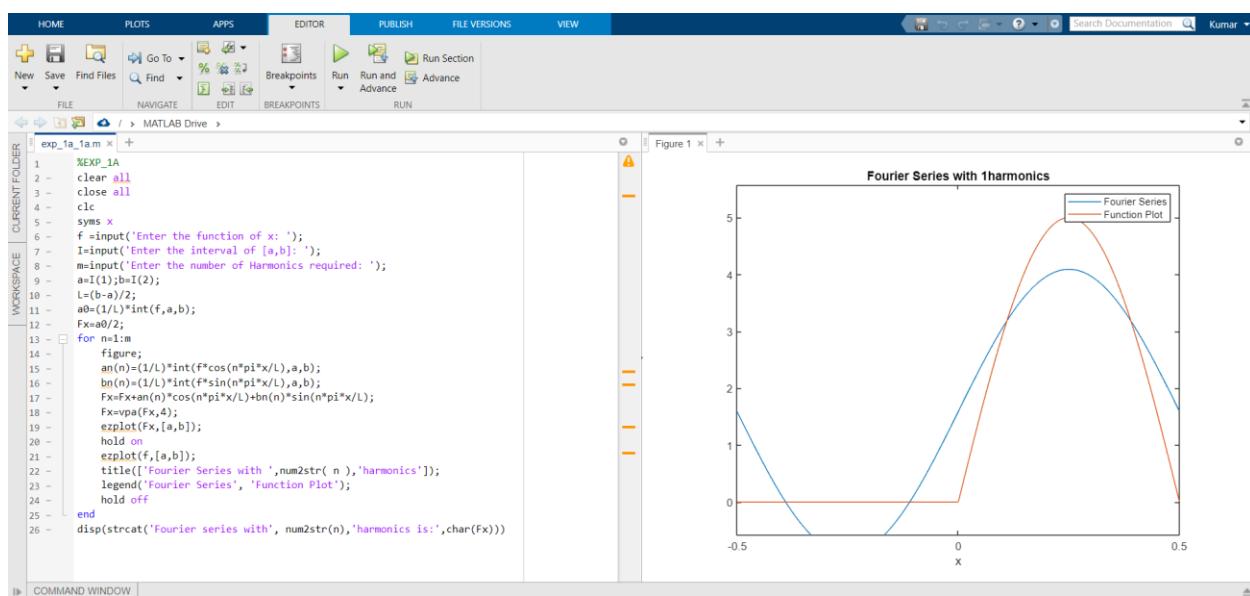


Figure 20 : 2 - Code & figure output

Department of Mathematics
School of Advanced Sciences
MAT 2002 – Applications of Differential and Difference Equations (MATLAB)
Experiment 1-B
Harmonic Analysis

Introduction:

In practice, the function is often not given by a formula, but a table of corresponding values. In such cases, the integrals in the Fourier series cannot be evaluated, instead the following forms can be used.

Since the mean value of the function $y = f(x)$ over the interval (a, b) is $\frac{1}{b-a} \int_a^b f(x)dx$, the

Fourier coefficients in the interval $(\alpha, \alpha + 2l)$ can be written as

$$a_0 = 2 \times \frac{1}{2l} \int_{\alpha}^{\alpha+2l} f(x)dx = 2 \times \text{Mean of } f(x) \text{ in } (\alpha, \alpha + 2l)$$

$$a_n = 2 \times \frac{1}{2l} \int_{\alpha}^{\alpha+2l} f(x) \cos\left(\frac{n\pi x}{l}\right) dx = 2 \times \text{Mean of } f(x) \cos\left(\frac{n\pi x}{l}\right) \text{ in } (\alpha, \alpha + 2l)$$

$$b_n = 2 \times \frac{1}{2l} \int_{\alpha}^{\alpha+2l} f(x) \sin\left(\frac{n\pi x}{l}\right) dx = 2 \times \text{Mean of } f(x) \sin\left(\frac{n\pi x}{l}\right) \text{ in } (\alpha, \alpha + 2l)$$

Hence the Fourier series for the tabulated data can be written as

$$f(x) = \frac{a_0}{2} + a_1 \cos \theta + b_1 \sin \theta + a_2 \cos 2\theta + b_2 \sin 2\theta + \dots \text{ where } \theta = \frac{\pi x}{l}.$$

The term $(a_1 \cos x + b_1 \sin x)$ is called the Fundamental Harmonic/First Harmonic, $(a_2 \cos 2x + b_2 \sin 2x)$ is called the Second Harmonic and so on.

Given m data points of function $y = f(x)$ of period 2π or $2l$, we find the Fourier coefficients

$$a_0 = 2 \frac{\sum f(x_i)}{m}, \quad a_n = 2 \frac{\sum_{i=1}^m f(x_i) \cos n\theta_i}{m}, \quad b_n = 2 \frac{\sum_{i=1}^m f(x_i) \sin n\theta_i}{m}, \quad \theta_i = \frac{\pi x_i}{l}, \quad n = 1, 2, \dots$$

MATLAB Syntax Used:

<code>syms var1 var2</code>	Creates symbolic variables var1 and var2
<code>disp(x)</code>	Displays the contents of x without printing the variable name
<code>length(X)</code>	returns the length of vector X
<code>plot(fun)</code>	Plots the discrete function fun whose domain and range are given.

MATLAB Code:

```
clear all
clc
syms t
x=input('Enter the equally spaced values of x: ');
y=input('Enter the values of y=f(x): ');
m=input('Enter the number of harmonics required: ');
n=length(x);a=x(1);b=x(n);
h=x(2)-x(1);
L=(b-a+h)/2;
theta=pi*x/L;
a0=(2/n)*sum(y);
Fx=a0/2; x1=linspace(a,b,100);
for i=1:m
figure
an=(2/n)*sum(y.*cos(i*theta));
bn=(2/n)*sum(y.*sin(i*theta));
Fx=Fx+an*cos(i*pi*t/L)+bn*sin(i*pi*t/L) ;
Fx=vpa(Fx,4);
Fx1=subs(Fx,t,x1);
plot(x1,Fx1);
hold on
plot(x,y);
title(['Fourier Series with ',num2str(i),'harmonics'])
legend('Fourier Series', 'Function Plot')
hold off;
end
disp(strcat('Fourier series with', num2str(i), 'harmonics
is:',char(Fx)));
```

Example 1: Compute the first four harmonics of the Fourier series given by the following table:

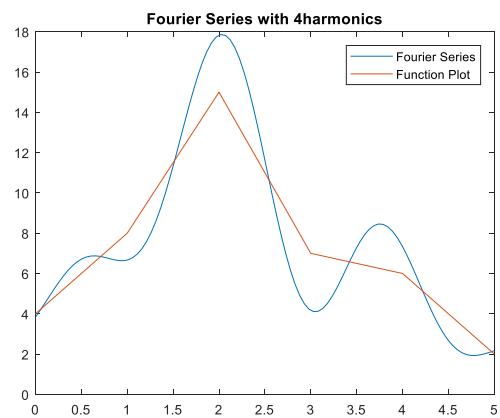
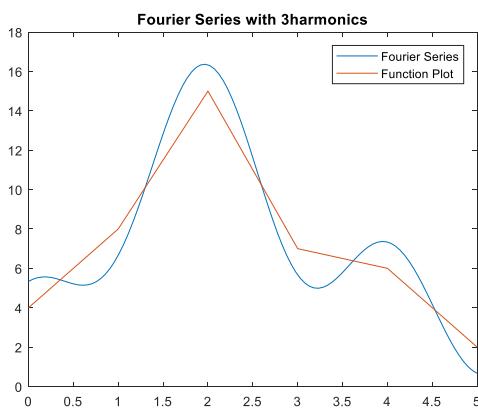
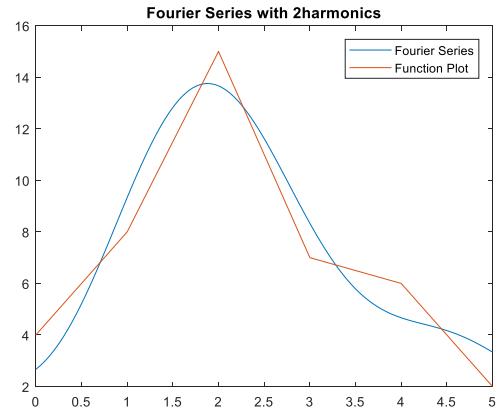
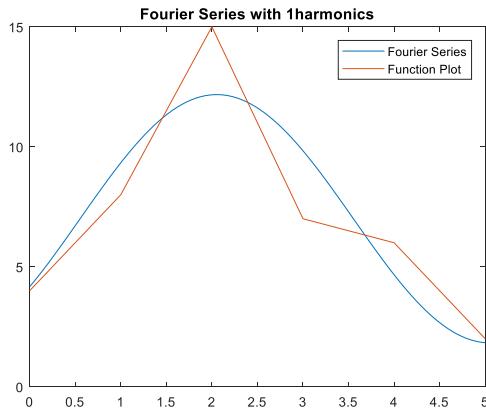
x	0	1	2	3	4	5
y	4	8	15	7	6	2

Input

```
Enter the equally spaced values of x: 0:5
Enter the values of y=f(x): [4 8 15 7 6 2]
Enter the number of harmonics required: 4
```

Output

```
Fourier series with4harmonics is:4.33*sin(1.047*t) -
1.5*cos(2.094*t) - 1.5*cos(4.189*t) - 2.833*cos(1.047*t) -
0.866*sin(2.094*t) + 0.866*sin(4.189*t) + 2.667*cos(3.142*t) -
6.123e-16*sin(3.142*t) + 7.0
```



Exercise:

1. The following table gives the variations of periodic current over a period T_0

T_0 sec	0	$T/6$	$T/3$	$T/2$	$2T/3$	$5T/6$	T
A amp	1.98	1.30	1.05	1.30	-0.88	-0.25	1.98

Show that there is a direct current part of 0.75 amp in the variable current and obtain the amplitude of the first three harmonics (Take $T = 1$).

2. Find the constant, the first sine and cosine terms in the Fourier series expansion of the function $y = f(x)$ tabulated below:

x	0	1	2	3	4	5
$y = f(x)$	6	15	18	22	17	12

1. The following table gives the variations of periodic current over a period T_0

T_0 sec	0	$T/6$	$T/3$	$T/2$	$2T/3$	$5T/6$	T
A amp	1.98	1.30	1.05	1.30	-0.88	-0.25	1.98

Show that there is a direct current part of 0.75 amp in the variable current and obtain the amplitude of the first three harmonics (Take $T = 1$).

Sol: -

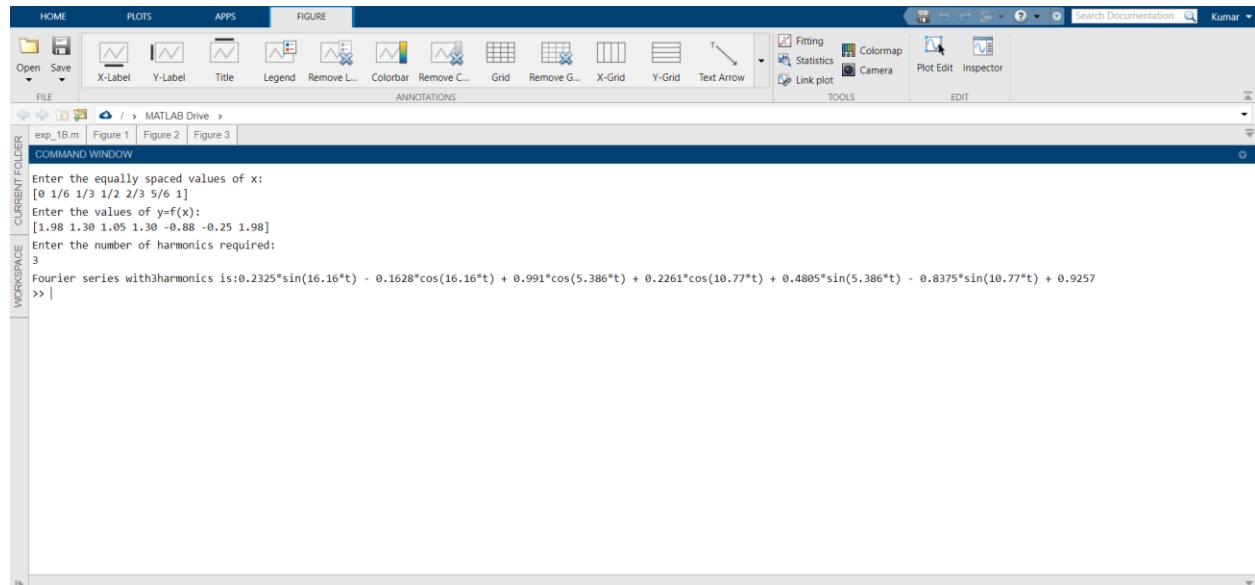


Figure 1:1 - Command window output

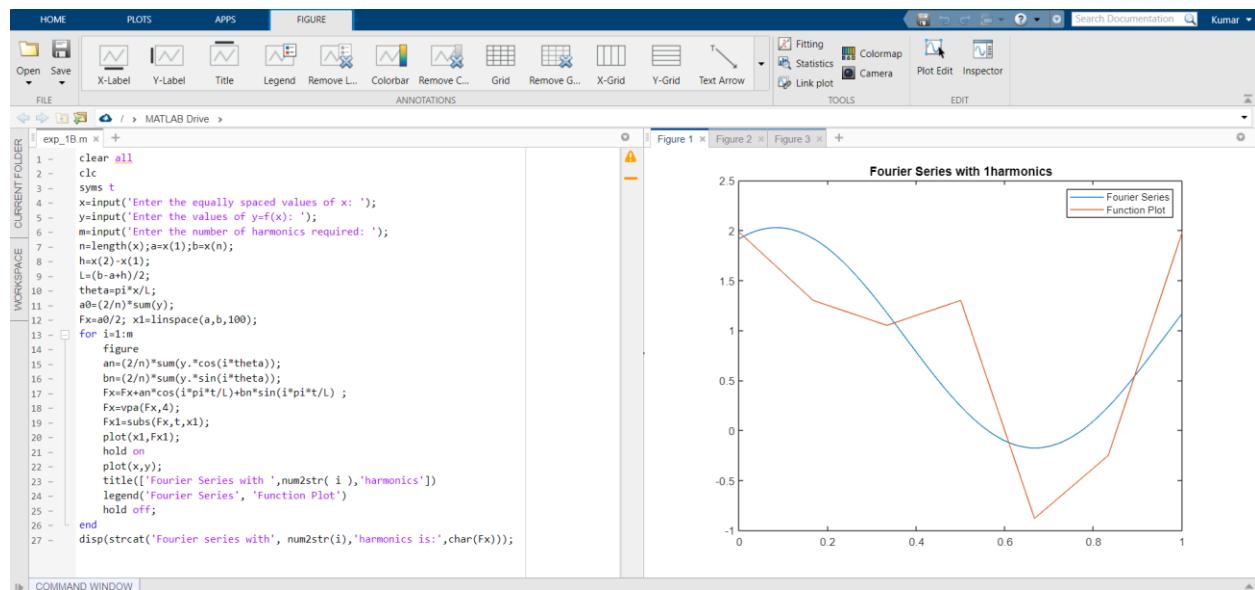


Figure 2 : 1 - Fourier series with 1 harmonic

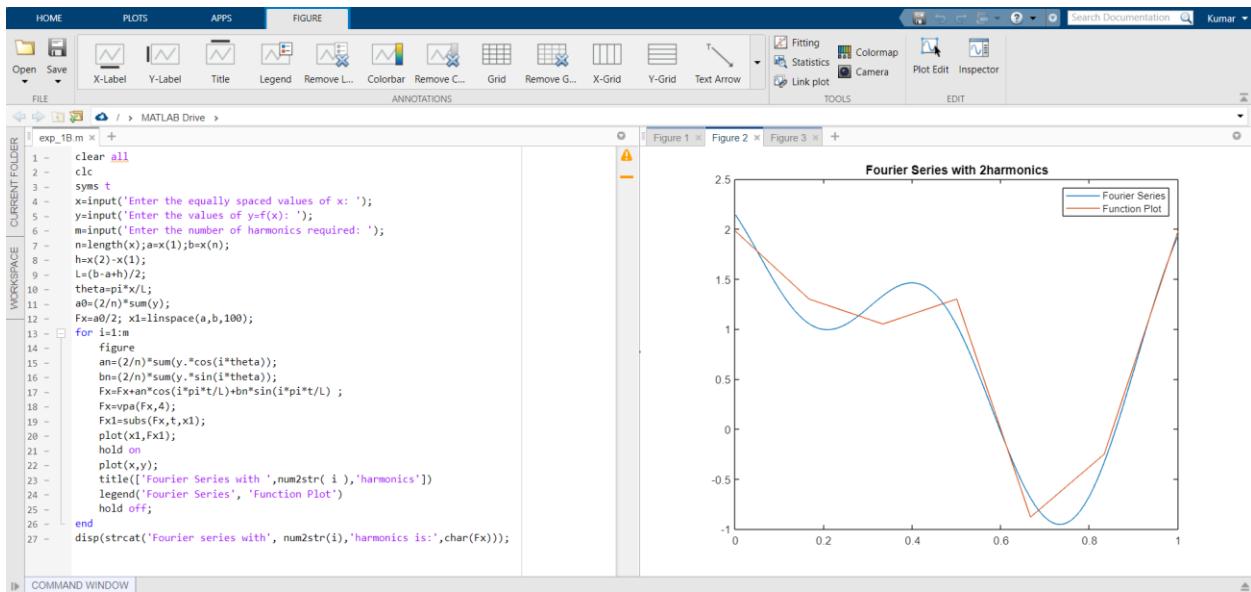


Figure 3 : 1 - Fourier Series with 2 harmonics

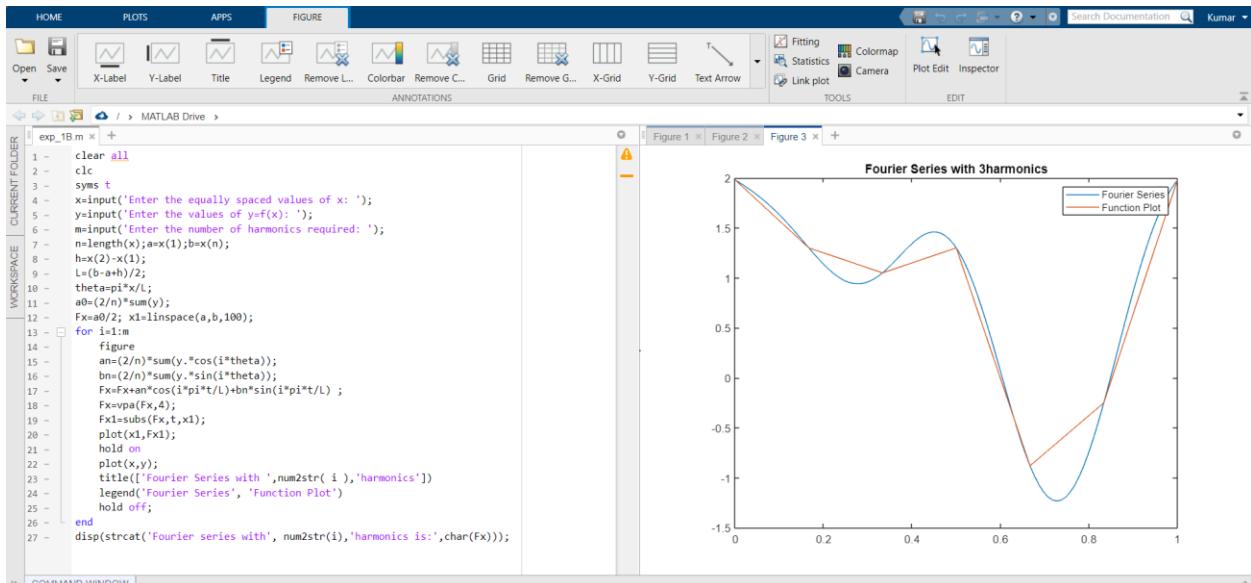


Figure 4 : 1 - Fourier Series with 3 harmonics

2. Find the constant, the first sine and cosine terms in the Fourier series expansion of the function $y = f(x)$ tabulated below:

x	0	1	2	3	4	5
$y = f(x)$	6	15	18	22	17	12

Sol: -

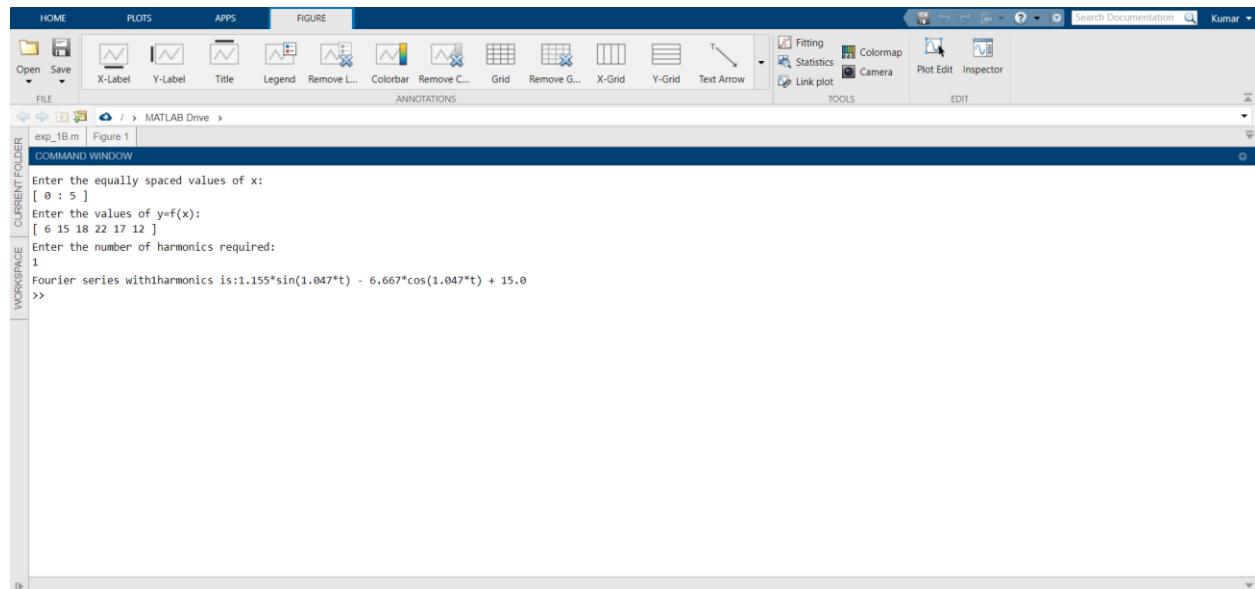


Figure 5 : 2 - Command window output

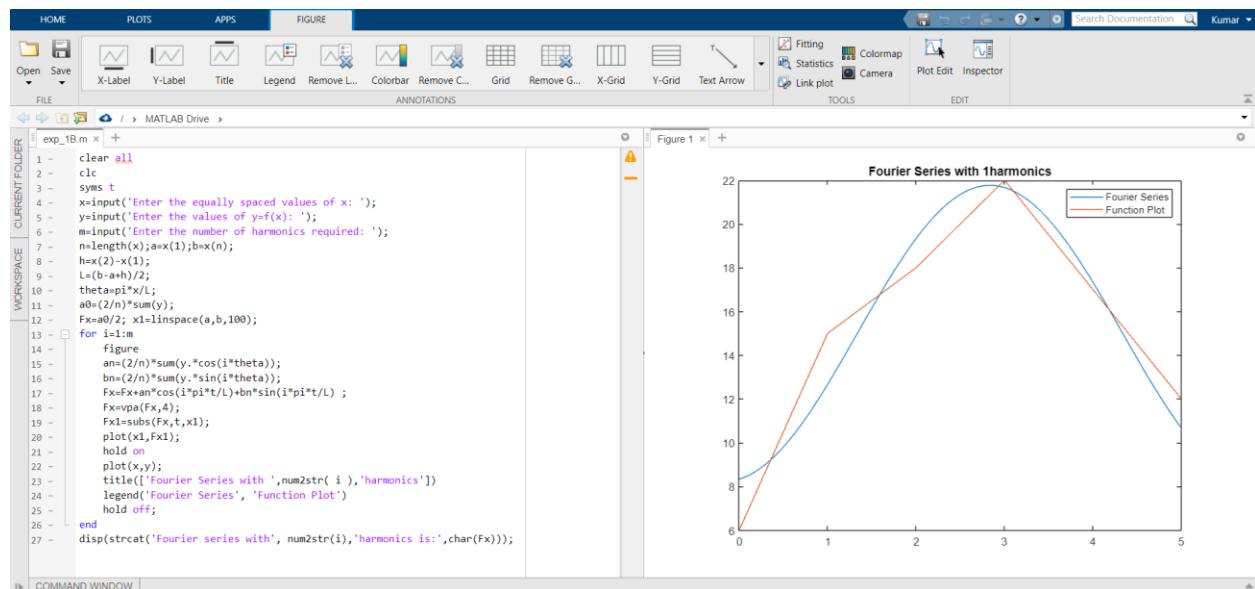


Figure 6 : 2 - Code & Fourier Series with 1 harmonic



DEPARTMENT OF MATHEMATICS SCHOOL OF ADVANCED SCIENCES

Properties of Eigen values and Eigen vectors, Cayley Hamilton theorem.

Course Code: MAT2002

Course Name: Application of Differential and Difference Equations

Experiment: 2-A

Duration: 90 Minutes

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 an scalar.

given an $n \times n$ matrix A , find the value of λ such that $[A - \lambda I]X = 0$ admits non-trival solution, and find those non-trival solution.

This is called the **Eigenvalue Problem**

Solving **characterstic 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.

1. Find the eigenvalues and eigenvectors of the following matrices:

(a) $\begin{bmatrix} 3 & 4 \\ 4 & -3 \end{bmatrix}$

(b) $\begin{bmatrix} 7 & -2 & 2 \\ -2 & 1 & 4 \\ -2 & 4 & 1 \end{bmatrix}$

MATLAB CODE

```
clc  
clear
```

```
A=input('Enter the Matrix: ');
```

```
%Characteristic Equation  
cf=poly(A);  
disp('Characteristic Equations')  
disp(cf)
```

```
%Eigenvalues  
EV=eig(A);  
disp('Eigenvalues')  
disp(EV)
```

```
%Eigenvectors  
[P D]=eig(A);  
disp('Eigenvectors')  
disp(P)
```

INPUT

```
Enter the Matrix: [3 4;4 -3]
```

OUTPUT

Eigenvalues

-5

5

Eigenvectors

0.4472 -0.8944

-0.8944 -0.4472

INPUT

Enter the Matrix: [7 -2 2; -2 1 4; -2 4 1]

OUTPUT

Characteristic Equations

1.0000 -9.0000 -1.0000 105.0000

Eigenvalues

7.0000

5.0000

-3.0000

Eigenvectors

0.5774 0.0000 -0.2709

-0.5774 -0.7071 -0.7450

-0.5774 -0.7071 0.6096

2. Prove the following statement by MATLAB

The product of the eigenvalues of a matrix A is equal to its determinant.

MATLAB CODE

clc

clear

```
A=input('Enter the Matrix: ');
```

```
%Determinant
```

```
detA=det(A);
```

```
disp('Determinant of A:')
```

```
disp(detA)
```

```
%Eigenvalues
```

```
EV=eig(A);
```

```
disp('Eigenvalues:')
```

```
disp(EV)
```

```
%Product of eigenvalues
```

```
prev=prod(EV);
```

```
disp('Product of Eigenvalues:')
```

```
disp(prev)
```

INPUT

Enter the Matrix: [7 -2 2; -2 1 4; -2 4 1]

OUTPUT

Determinant of A:

-105

Eigenvalues:

7.0000

5.0000

-3.0000

Product of Eigenvalues:

-105.0000

3. Verify Cayley-Hamilton theorem for the matrix $A = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$ and find its inverse.

MATLAB CODE

```
clc  
clear
```

```
A=input('Enter the Matrix: ');\n\n%Verification of Cayley-Hamilton theorem\ncf=poly(A);\nn=length(cf);\nCHT=cf(1)*A^(n-1);\nfor i=2:n\n    CHT=CHT+cf(i)*A^(n-i);\nend\ndisp('R.H.S of C-H Theorem: ')\ndisp(round(CHT))\n\n%To find the inverse\nINV=cf(1)*A^(n-2);\nfor i=2:n-1\n    INV=INV+cf(i)*A^(n-i-1);\nend\nINV=INV/(-cf(n));\ndisp('Inverse of A: ')\ndisp(INV)
```

INPUT

Enter the Matrix: [1 4;2 3]

OUTPUT

R.H.S of C-H Theorem:

0	0
0	0

Inverse of A:

$$\begin{matrix} -0.6000 & 0.8000 \\ 0.4000 & -0.2000 \end{matrix}$$

Exercise

4. Prove the following statements:

- (a) The sum of the eigen values of a matrix is the sum of the elements of the principal diagonal.
- (b) If λ is an eigenvalues of a matrix A , then $\frac{1}{\lambda}$ is the eigenvalue of A^{-1} .
- (c) 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).

5. Using Cayley-Hamilton theorem,

(a) find the inverse of $A = \begin{bmatrix} 1 & 1 & 3 \\ 1 & 3 & -3 \\ -2 & -4 & -4 \end{bmatrix}$.

(b) find A^8 , if $A = \begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}$.

MAT2002 – Applications of Differential & Difference Equations

Properties of Eigen values and Eigen vectors, Cayley Hamilton theorem.

(Exp – 2A)

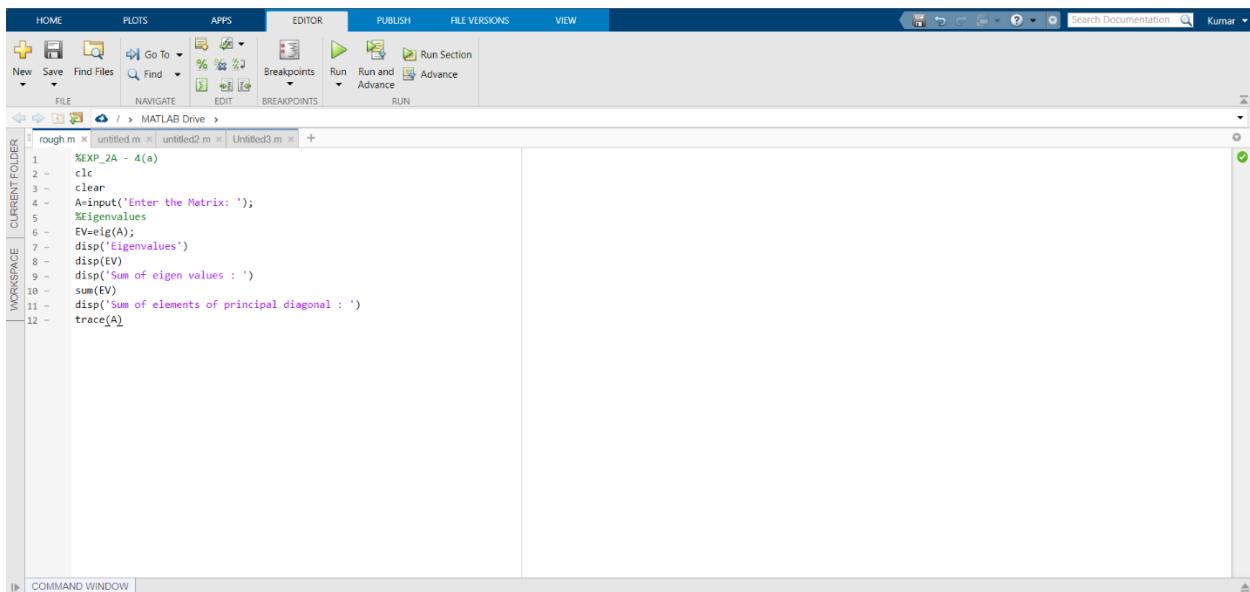
Faculty Name -
Monica C

Presented By -
Kumar Sparsh
19BCB0025

4. Prove the following statements:

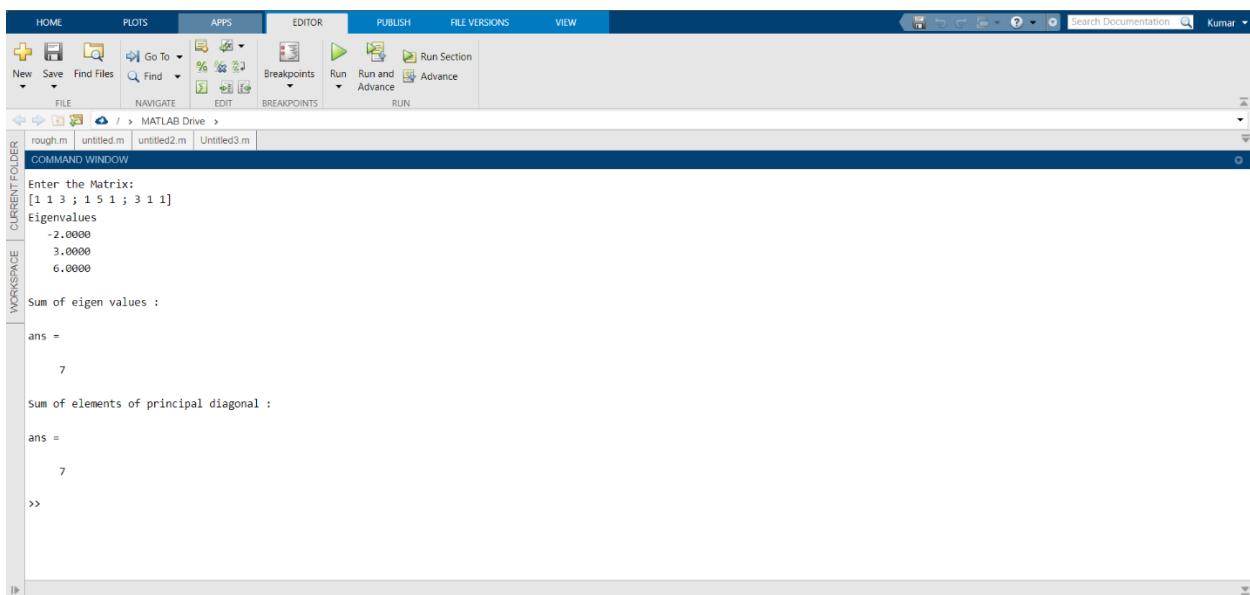
- (a) The sum of the eigen values of a matrix is the sum of the elements of the principal diagonal.

Sol: -



```
%EXP_2A ~ 4(a)
clc
clear
A=input('Enter the Matrix: ');
%Eigenvalues
EV=eig(A);
disp('Eigenvalues')
disp(EV)
disp('Sum of eigen values : ')
sum(EV)
disp('Sum of elements of principal diagonal : ')
trace(A)
```

Figure 1 : 4(a) Code



```
Enter the Matrix:
[1 1 3 ; 1 5 1 ; 3 1 1]
Eigenvalues
-2.0000
3.0000
6.0000

Sum of eigen values :

ans =
7

Sum of elements of principal diagonal :

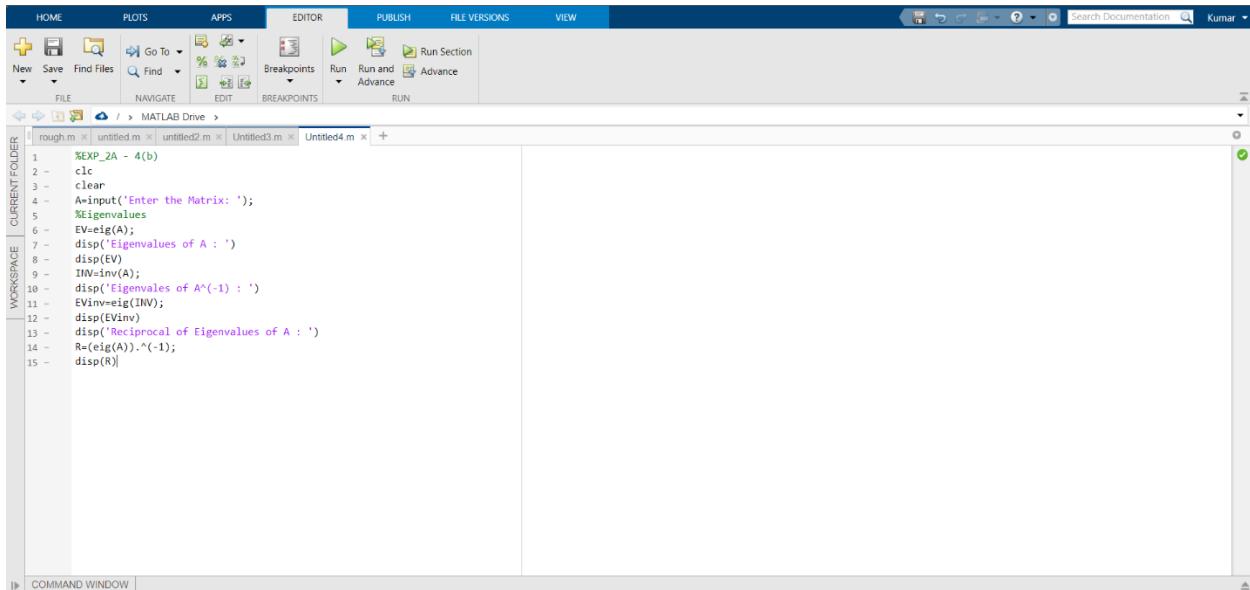
ans =
7

>>
```

Figure 2 : 4(a) Command Window output

(b) If λ is an eigenvalues of a matrix A , then $\frac{1}{\lambda}$ is the eigenvalue of A^{-1} .

Sol: -



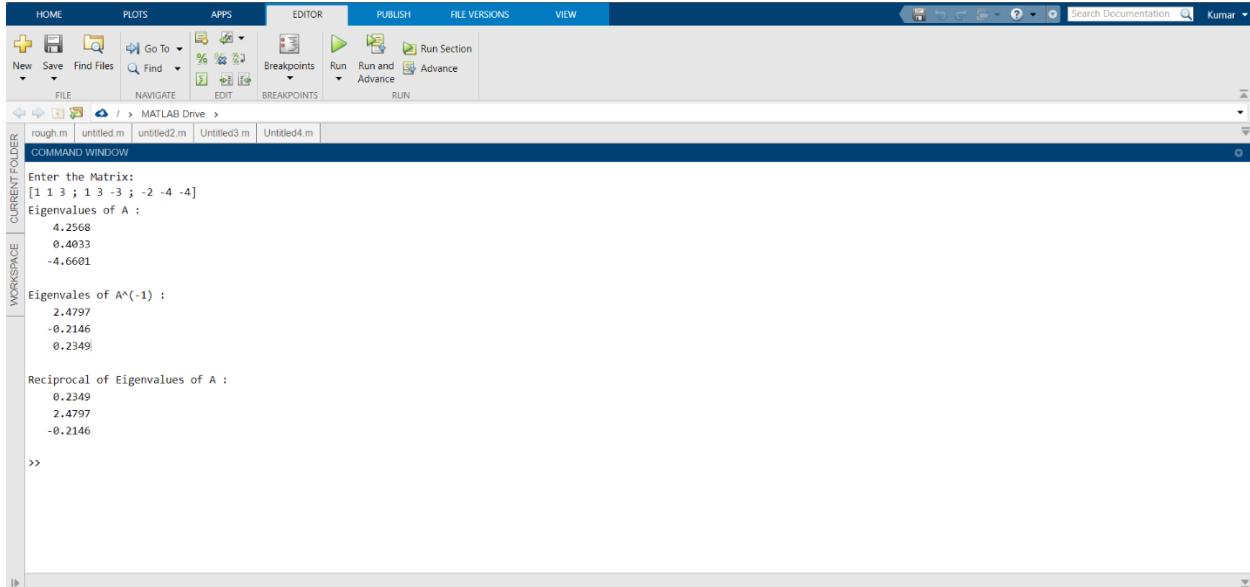
The screenshot shows the MATLAB Editor window. The code in the editor is as follows:

```

1 %EXP_2A - 4(b)
2 -
3 clc
4 A=input('Enter the Matrix: ');
5 %Eigenvalues
6 EV=eig(A);
7 disp('Eigenvalues of A : ')
8 disp(EV)
9 INV=inv(A);
10 disp('Eigenvalues of A^(-1) : ')
11 EVinv=eig(INV);
12 disp(EVinv);
13 disp('Reciprocal of Eigenvalues of A : ')
14 R=(eig(A)).^(-1);
15 disp(R)

```

Figure 3 : 4(b) Code



The screenshot shows the MATLAB Command Window. The output is as follows:

```

Enter the Matrix:
[1 1 3 ; 1 3 -3 ; -2 -4 -4]
Eigenvalues of A :
    4.2568
    0.4033
   -4.6601

Eigenvalues of A^(-1) :
    2.4797
   -0.2146
    0.2349

Reciprocal of Eigenvalues of A :
    0.2349
    2.4797
   -0.2146

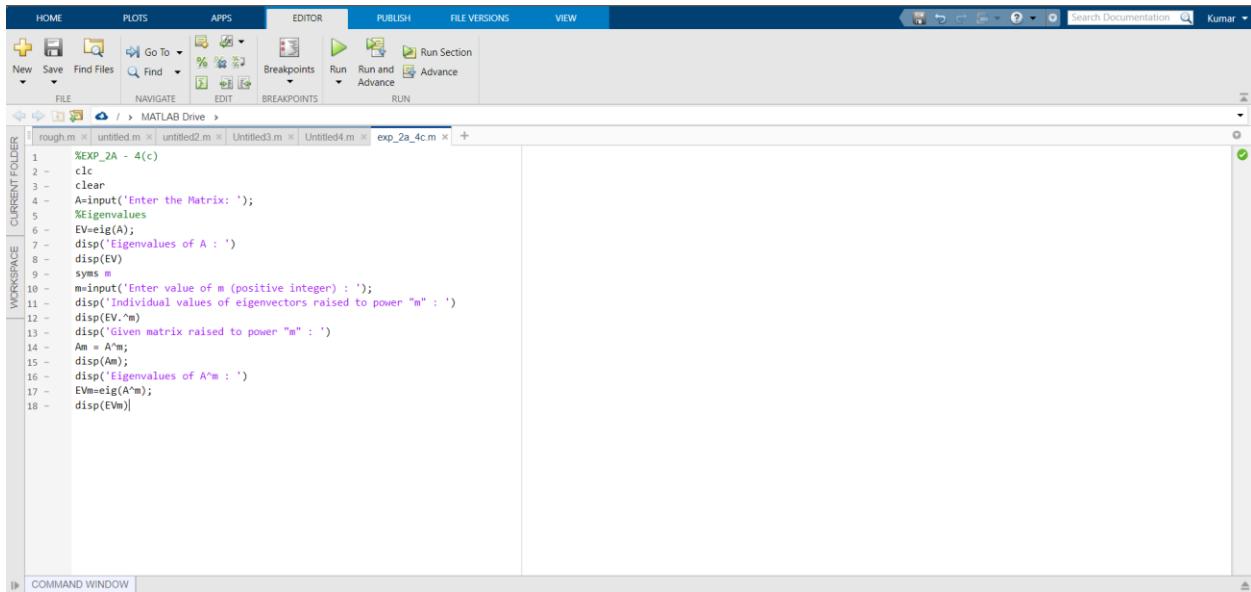
>>

```

Figure 4 : 4(b) Command Window output

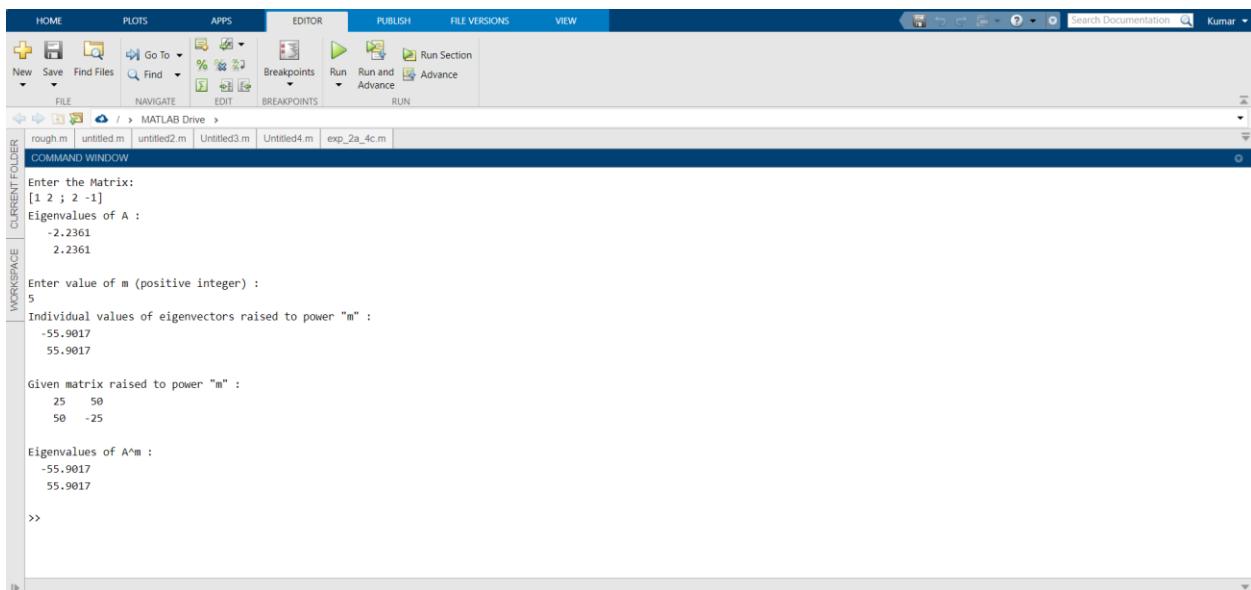
(c) If $\lambda_1, \lambda_2, \dots, \lambda_n$ are the eigenvalues of a matrix, then A^m has the eigenvalues $\lambda_1^m, \lambda_2^m, \dots, \lambda_n^m$ (m being a positive integer).

Sol: -



```
%EXP_2A - 4(c)
clc
clear
A=input('Enter the Matrix: ');
%Eigenvalues
EV=eig(A);
disp('Eigenvalues of A : ')
disp(EV)
syms m
m=input('Enter value of m (positive integer) : ');
disp('Individual values of eigenvectors raised to power "m" : ')
disp(EV.^m)
disp('Given matrix raised to power "m" : ')
Am = A^m;
disp(Am);
disp('Eigenvalues of A^m : ')
EVmeig=eig(A^m);
disp(EVmeig)
```

Figure 5 : 4(c) MATLAB Code



```
Enter the Matrix:
[1 2 ; 2 -1]
Eigenvalues of A :
 -2.2361
 2.2361

Enter value of m (positive integer) :
5
Individual values of eigenvectors raised to power "m" :
 -55.9017
 55.9017

Given matrix raised to power "m" :
 25   50
 50  -25

Eigenvalues of A^m :
 -55.9017
 55.9017

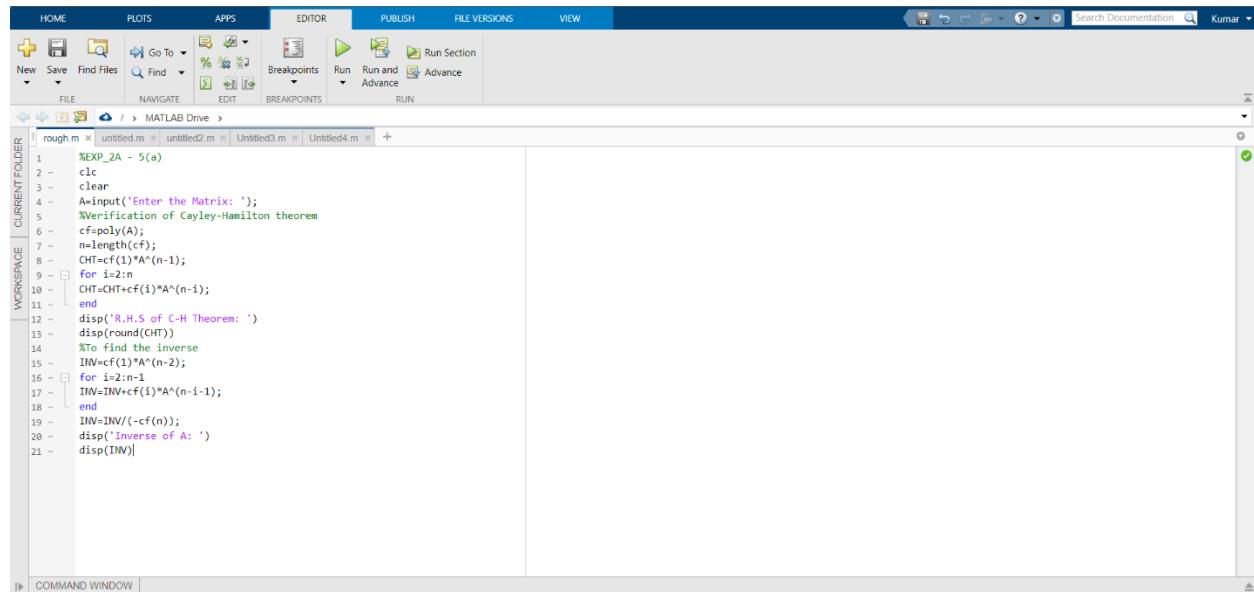
>>
```

Figure 6 : 4(c) Command Window output

5. Using Cayley-Hamilton theorem,

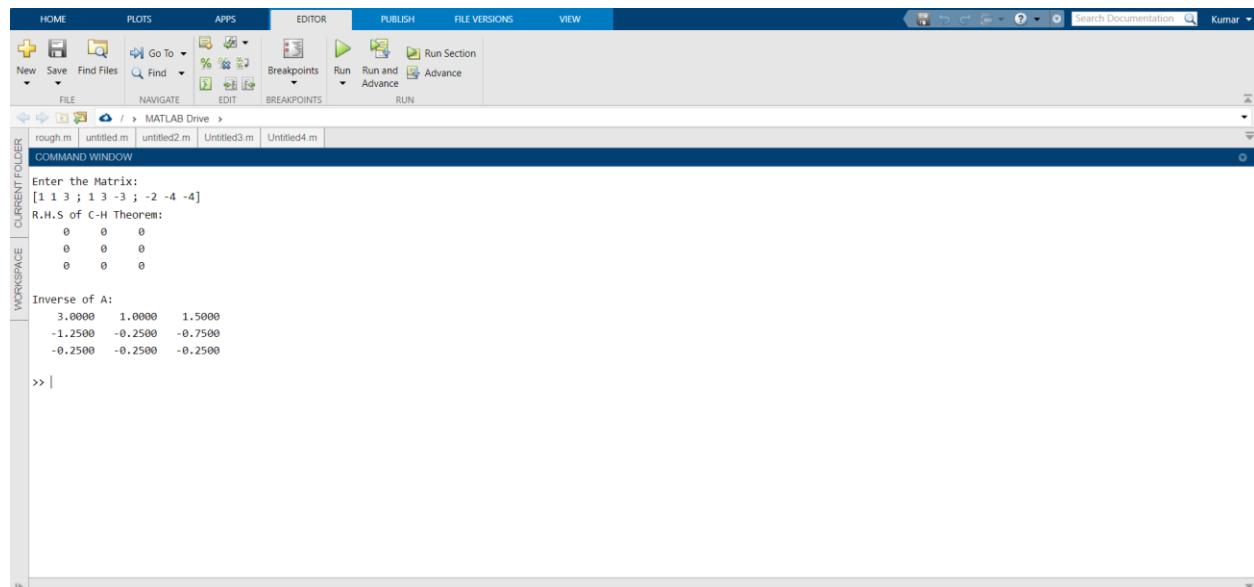
(a) find the inverse of $A = \begin{bmatrix} 1 & 1 & 3 \\ 1 & 3 & -3 \\ -2 & -4 & -4 \end{bmatrix}$.

Sol: -



```
%EXP_2A - 5(a)
1 %Enter the Matrix: '
2 clc
3 clear
4 A=input('
5 %Verification of Cayley-Hamilton theorem
6 cf=poly(A);
7 n=length(cf);
8 CHT=cf(1)*A^(n-1);
9 for i=2:n
10 CHT=CHT+cf(i)*A^(n-i);
11 end
12 disp('R.H.S of C-H Theorem: ')
13 disp(round(CHT))
14 %To find the inverse
15 INV=cf(1)*A^(n-2);
16 for i=2:n-1
17 INV=INV+cf(i)*A^(n-i-1);
18 end
19 INV=INV/(-cf(n));
20 disp('Inverse of A: ')
21 disp(INV)|
```

Figure 7 : 5(a) MATLAB Code



```
Enter the Matrix:
[1 1 3 ; 1 3 -3 ; -2 -4 -4]
R.H.S of C-H Theorem:
0 0 0
0 0 0
0 0 0

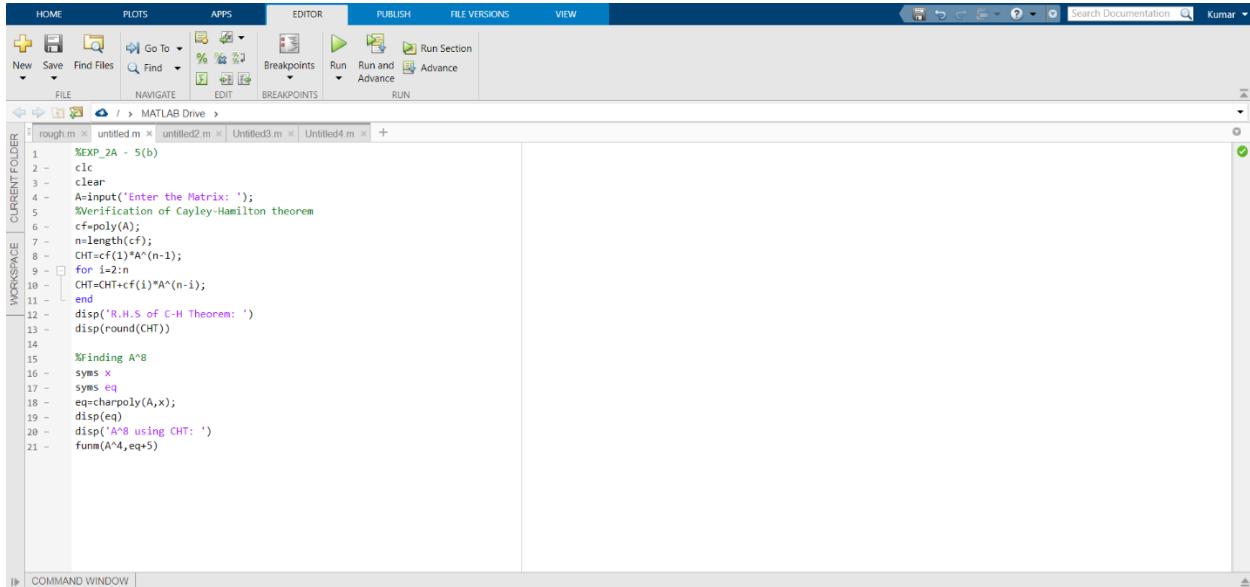
Inverse of A:
3.0000 1.0000 1.5000
-1.2500 -0.2500 -0.7500
-0.2500 -0.2500 -0.2500

>> |
```

Figure 8 : 5(a) Command Window output

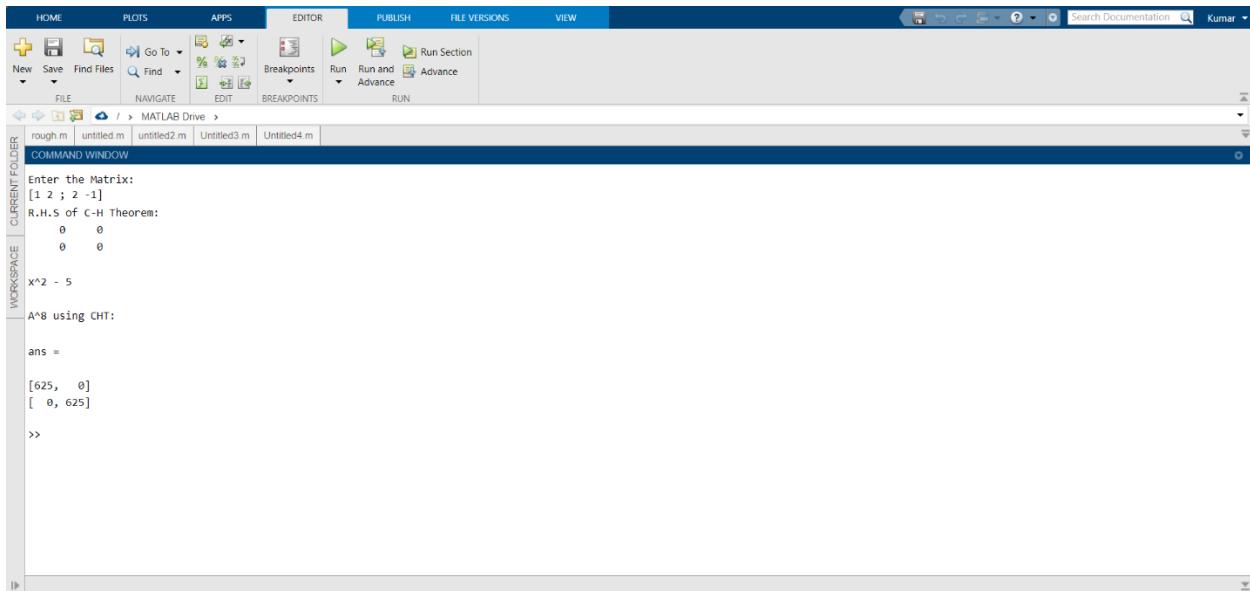
(b) find A^8 , if $A = \begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}$.

Sol: -



```
%EXP_2A - 5(b)
1 %Enter the Matrix: ')
2 clc
3 clear
4 A=input('
5 %Verification of Cayley-Hamilton theorem
6 cf=poly(A);
7 n=length(cf);
8 CHT=cf(1)*A^(n-1);
9 for i=2:i
10 CHT=CHT+cf(i)*A^(n-i);
11 end
12 disp('R.H.S of C-H Theorem: ')
13 disp(round(CHT))
14
15 %Finding A^8
16 syms x
17 syms eq
18 eq=charpoly(A,x);
19 disp(eq)
20 disp('A^8 using CHT: ')
21 funm(A^4,eq+5)
```

Figure 9 : 5(b) MATLAB Code



```
Enter the Matrix:
[1 2 ; 2 -1]
R.H.S of C-H Theorem:
0 0
0 0

X^2 - 5

A^8 using CHT:

ans =
[ 625, 0]
[ 0, 625]

>>
```

Figure 10 : 5(b) Command Window output



DEPARTMENT OF MATHEMATICS SCHOOL OF ADVANCED SCIENCES

Diagonalization by Similarity transformation, Orthogonal transformation.

Course Code: MAT2002

Course Name: Application of Differential and Difference Equations

Experiment: 2-B

Duration: 90 Minutes

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**.

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

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 $N^TAN = D$ since $N^{-1} = N^T$ by a property of orthogonal matrix. Transforming A into D by means of the transformation $N^TAN = D$ is called as **orthogonal reduction or orthogonal transformation**.

1. Reduce the following matrices to the diagonal form by similarity transformation:

$$(a) A = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}$$

$$(b) A = \begin{bmatrix} 2 & 2 & -7 \\ 2 & 1 & 2 \\ 0 & 1 & -3 \end{bmatrix}$$

MATLAB CODE

```
clc  
clear
```

```
A=input('Enter the matrix for diagonalization :');
```

```
[P D]=eig(A);  
disp('Given Matrix (A) :')  
disp(A)  
disp('Modal Matrix (P):')  
disp(P)  
disp('Inverse of P :')  
PI=inv(P);  
disp(PI)  
disp('Diagonal Matrix (D=P^(-1)*A*P):')  
DM=round(inv(P)*A*P, 2);  
disp(DM)
```

INPUT

```
Enter the matrix for diagonalization :[4 1;2 3]
```

OUTPUT

Given Matrix (A) :

$$\begin{matrix} 4 & 1 \\ 2 & 3 \end{matrix}$$

Modal Matrix (P):

$$\begin{matrix} 0.7071 & -0.4472 \\ 0.7071 & 0.8944 \end{matrix}$$

Inverse of P :

$$\begin{matrix} 0.9428 & 0.4714 \\ -0.7454 & 0.7454 \end{matrix}$$

Diagonal Matrix (D=P^(-1)*A*P):

$$\begin{matrix} 5 & 0 \\ 0 & 2 \end{matrix}$$

INPUT

Enter the matrix for diagonalization :[2 2 -7;2 1 2;0 1 -3]

OUTPUT

Given Matrix (A) :

$$\begin{matrix} 2 & 2 & -7 \\ 2 & 1 & 2 \\ 0 & 1 & -3 \end{matrix}$$

Modal Matrix (P):

-0.6350	0.2357	0.7276
-0.7620	-0.9428	-0.4851
-0.1270	-0.2357	0.4851

Inverse of P :

-1.1249	-0.5624	1.1249
0.8485	-0.4243	-1.6971
0.1178	-0.3534	1.5314

Diagonal Matrix (D=P^(-1)*A*P):

3	0	0
0	1	0
0	0	-4

2. Diagonalise the matrix $A = \begin{bmatrix} 2 & 0 & 4 \\ 0 & 6 & 0 \\ 4 & 0 & 2 \end{bmatrix}$ by means of an orthogonal transformation.

MATLAB CODE

```
clc  
clear
```

```
A=input('Enter the symetric matrix for diagonalization :');
```

```
[P D]=eig(A);  
disp('Given Matrix (A) :')  
disp(A)  
disp('Modal Matrix (P):')  
disp(P)  
NP=normc(P);  
disp('Normalized Modal Matrix (N):')  
disp(NP)  
disp('Diagonal Matrix (D=N^ T*A*N) :')  
DM=round(NP'*A*NP,2);  
disp(DM)
```

INPUT

Enter the symmetric matrix for diagonalization :[2 0 4;0 6 0;4 0 2]

OUTPUT

Given Matrix (A) :

$$\begin{matrix} 2 & 0 & 4 \\ 0 & 6 & 0 \\ 4 & 0 & 2 \end{matrix}$$

Modal Matrix (P):

$$\begin{matrix} 0.7071 & 0.7071 & 0 \\ 0 & 0 & -1.0000 \\ -0.7071 & 0.7071 & 0 \end{matrix}$$

Normalized Modal Matrix (N):

$$\begin{matrix} 0.7071 & 0.7071 & 0 \\ 0 & 0 & -1.0000 \\ -0.7071 & 0.7071 & 0 \end{matrix}$$

Diagonal Matrix (D=N^T * A * N) :

$$\begin{matrix} -2 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 6 \end{matrix}$$

Exercise

3. Diagonalize $A = \begin{bmatrix} 1 & 6 & 1 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$ by similarity transformation and hence find A^8 .
 4. Transform the quadratic form $13x^2 - 10xy + 13y^2$ to canonical form and specify the matrix of transformation.
 5. Transform the quadratic form $3x_1^2 + 5x_2^2 + 3x_3^2 - 2x_2x_3 + 2x_3x_1 - 2x_1x_2$ to canonical form and specify the matrix of transformation.
-

MAT2002 – Applications of Differential & Difference Equations

Diagonalization by Similarity transformation, Orthogonal transformation

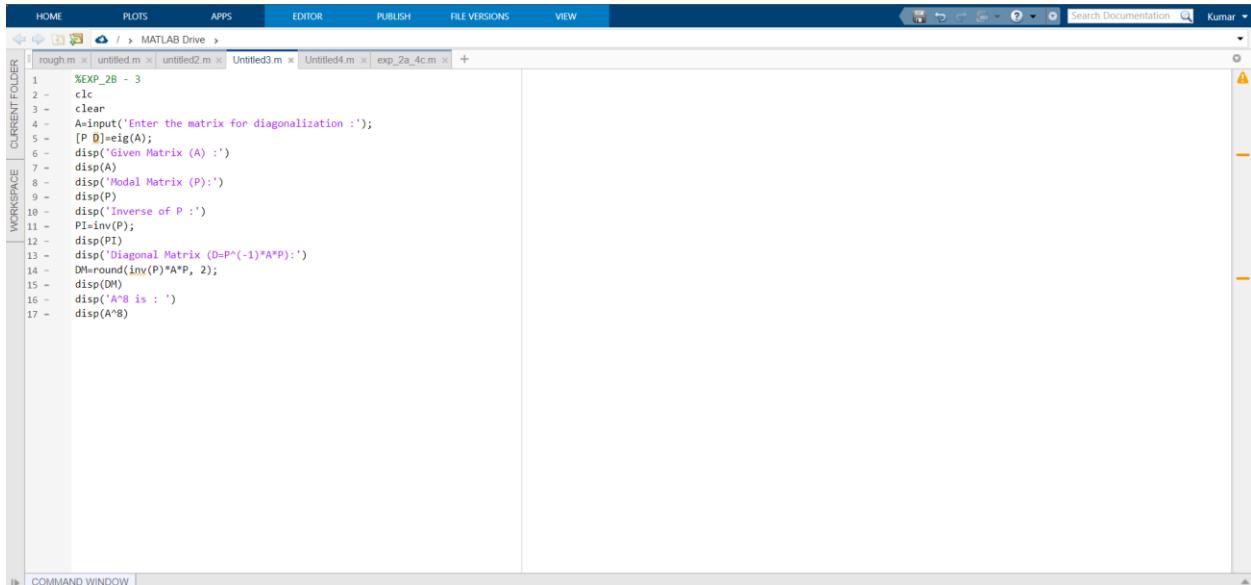
(Exp – 2B)

Faculty Name -
Monica C

Presented By -
Kumar Sparsh
19BCB0025

3. Diagonalize $A = \begin{bmatrix} 1 & 6 & 1 \\ 1 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$ by similarity transformation and hence find A^8 .

Sol:



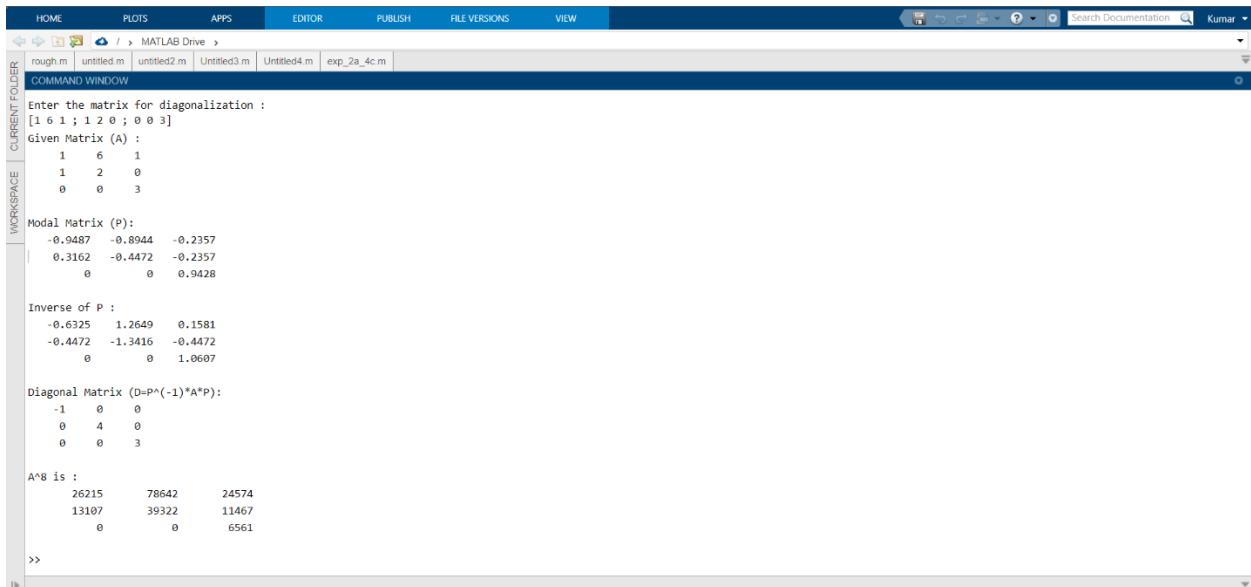
The screenshot shows the MATLAB interface with the code for diagonalization in the editor window. The code is as follows:

```

1 %EXP_2B - 3
2 clc
3 clear
4 A=input('Enter the matrix for diagonalization :');
5 [P,D]=eig(A);
6 disp('Given Matrix (A) :')
7 disp(A)
8 disp('Modal Matrix (P):')
9 disp(P)
10 disp('Inverse of P :')
11 P=inv(P);
12 disp(P);
13 disp('Diagonal Matrix (D=P^(-1)*A*p):')
14 D=round(inv(P)*A*p, 2);
15 disp(D)
16 disp('A^8 is : ')
17 disp(A^8)

```

Figure 1 : 3 - MATLAB Code



The screenshot shows the MATLAB Command Window output. The user enters the matrix A, and MATLAB displays the given matrix, the modal matrix P, the inverse of P, and the resulting diagonal matrix D. Finally, it calculates and displays A^8 .

```

Enter the matrix for diagonalization :
[1 6 1 ; 1 2 0 ; 0 0 3]
Given Matrix (A) :
1   6   1
1   2   0
0   0   3

Modal Matrix (P):
-0.0487   -0.8944   -0.2357
 0.3162   -0.4472   -0.2357
 0         0         0.9428

Inverse of P :
-0.6325   1.2649   0.1581
-0.4472   -1.3416   -0.4472
 0         0         1.0607

Diagonal Matrix (D=P^(-1)*A*p):
-1   0   0
 0   4   0
 0   0   3

A^8 is :
26215    78642    24574
13107    39322    11467
 0         0         6561
>>

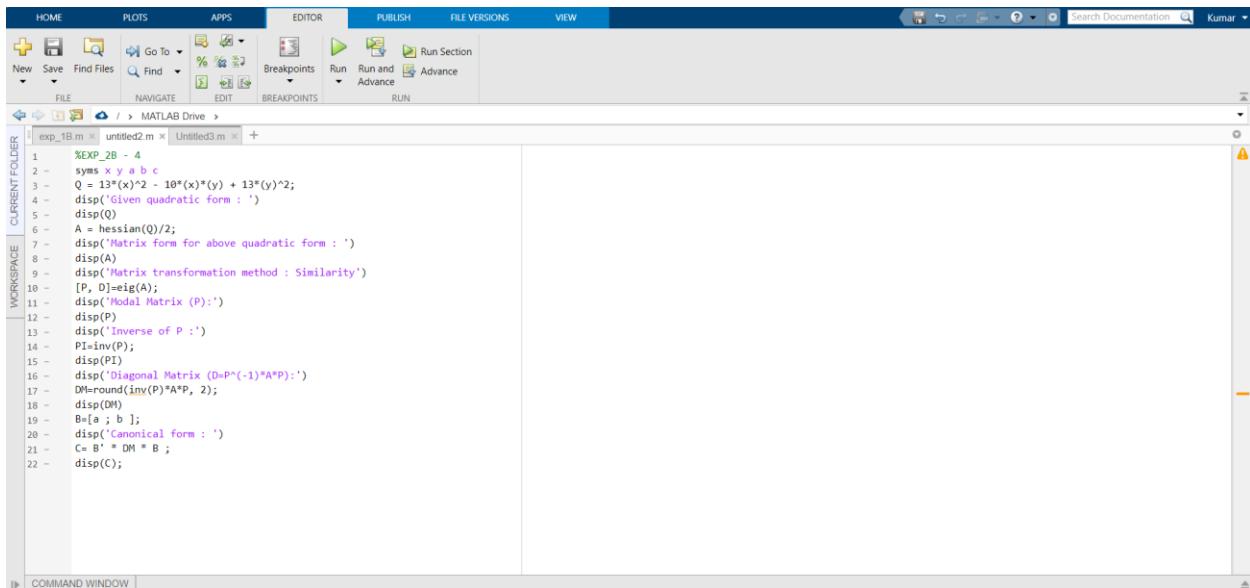
```

Figure 2 : 3 - Command Window output

4. Transform the quadratic form $13x^2 - 10xy + 13y^2$ to canonical form and specify the matrix of transformation.

Sol:

Similarity Transformation



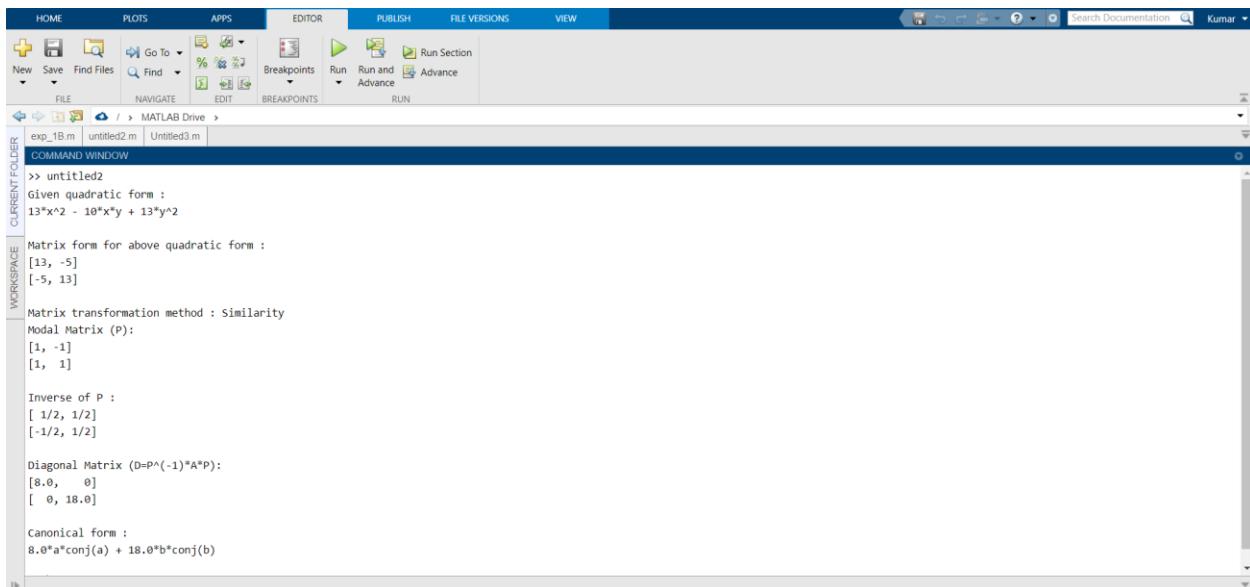
The screenshot shows the MATLAB Editor window. The code in the editor is as follows:

```

1 %EXP_2B - 4
2 - syms x y a b
3 - Q = 13*(x)^2 - 10*(x)*(y) + 13*(y)^2;
4 - disp('Given quadratic form : ')
5 - disp(Q)
6 - A = hessian(Q)/2;
7 - disp('Matrix form for above quadratic form : ')
8 - disp(A)
9 - disp('Matrix transformation method : Similarity')
10 - [P, D]=eig(A);
11 - disp('Modal Matrix (P):')
12 - disp(P)
13 - disp('Inverse of P :')
14 - PI=inv(P);
15 - disp(PI)
16 - disp('Diagonal Matrix (D=P^(-1)*A*P):')
17 - DM=round(inv(P)*A*P, 2);
18 - disp(DM)
19 - B=[a ; b];
20 - disp('Canonical form : ')
21 - C= B' * DM * B ;
22 - disp(C);

```

Figure 3 : 4 - MATLAB Code for Similarity Transformation



The screenshot shows the MATLAB Command Window. The output is as follows:

```

>> untitled2
Given quadratic form :
13*x^2 - 10*x*y + 13*y^2

Matrix form for above quadratic form :
[13, -5]
[-5, 13]

Matrix transformation method : Similarity
Modal Matrix (P):
[1, -1]
[1, 1]

Inverse of P :
[ 1/2, 1/2]
[-1/2, 1/2]

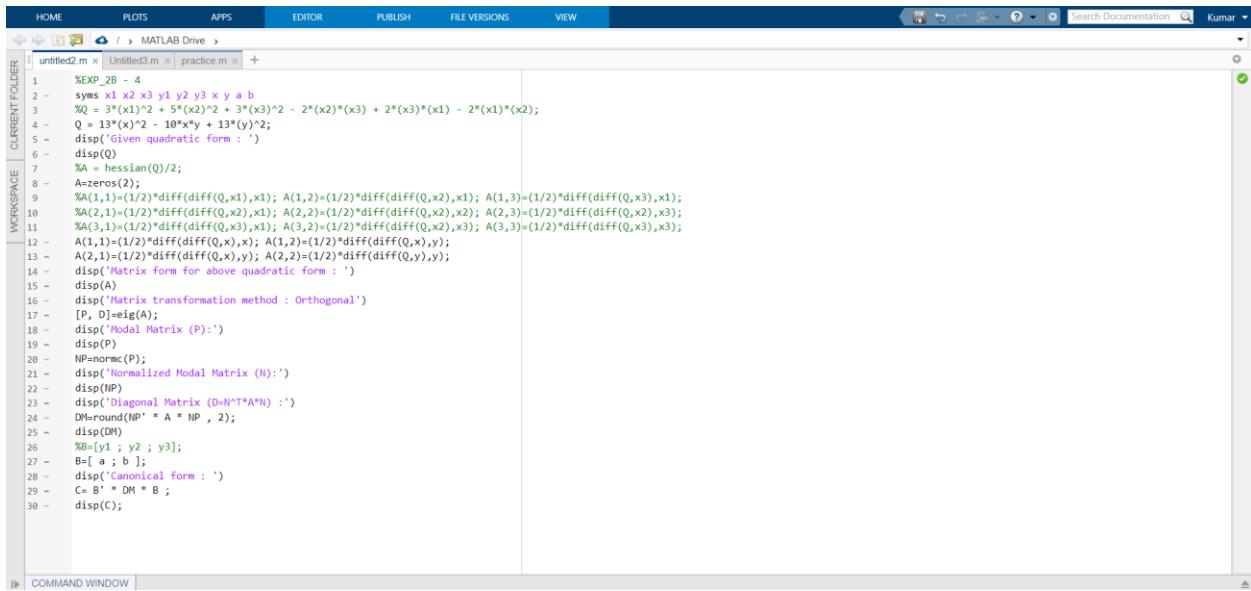
Diagonal Matrix (D=P^(-1)*A*P):
[8.0, 0]
[0, 18.0]

Canonical form :
8.0*a*conj(a) + 18.0*b*conj(b)

```

Figure 4 : 4 - Command Window output for Similarity Transformation

Orthogonal Transformation

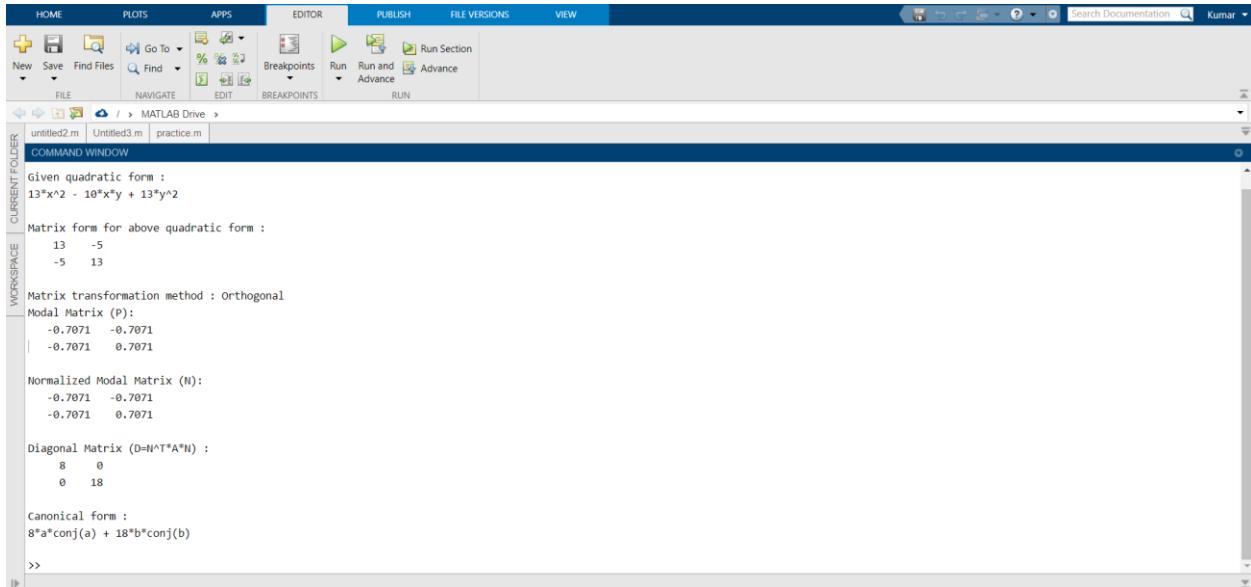


```

1 %EXP_2B - 4
2 - syms x1 x2 x3 y1 y2 y3 x y a b
3 - %Q = 3*(x1)^2 + 5*(x2)^2 + 3*(x3)^2 - 2*(x2)*(x3) + 2*(x3)*(x1) - 2*(x1)*(x2);
4 - Q = 13*x^2 - 10*x*y + 13*y^2;
5 - disp('Given quadratic form : ')
6 - disp(Q)
7 - A=zeros(2);
8 - A(1,1)=(1/2)*diff(diff(Q,x1),x1); A(1,2)=(1/2)*diff(diff(Q,x2),x1); A(1,3)=(1/2)*diff(diff(Q,x3),x1);
9 - A(2,1)=(1/2)*diff(diff(Q,x2),x1); A(2,2)=(1/2)*diff(diff(Q,x2),x2); A(2,3)=(1/2)*diff(diff(Q,x2),x3);
10 - A(3,1)=(1/2)*diff(diff(Q,x3),x1); A(3,2)=(1/2)*diff(diff(Q,x2),x3); A(3,3)=(1/2)*diff(diff(Q,x3),x3);
11 - A(1,1)=(1/2)*diff(diff(Q,x),x); A(1,2)=(1/2)*diff(diff(Q,y),x);
12 - A(2,1)=(1/2)*diff(diff(Q,x),y); A(2,2)=(1/2)*diff(diff(Q,y),y);
13 - disp('Matrix form for above quadratic form : ')
14 - disp(A)
15 - disp('Matrix transformation method : Orthogonal')
16 - [P, D]=eig(A);
17 - disp('Modal Matrix (P):')
18 - disp(P)
19 - NP=normc(P);
20 - disp('Normalized Modal Matrix (N):')
21 - disp(NP)
22 - disp('Diagonal Matrix (D=N*T*A*N) :')
23 - DM=round(NP'* A * NP , 2);
24 - disp(DM)
25 - %B=[y1 ; y2 ; y3];
26 - Be=[ a ; b ];
27 - disp('Canonical form : ')
28 - C= B' * DM * B ;
29 - disp(C);
30 -

```

Figure 5 : 4 - MATLAB Code for Orthogonal Transformation



```

Given quadratic form :
13*x^2 - 10*x*y + 13*y^2

Matrix form for above quadratic form :
13   -5
     -5   13

Matrix transformation method : Orthogonal
Modal Matrix (P):
-0.7071   -0.7071
|  -0.7071    0.7071

Normalized Modal Matrix (N):
-0.7071   -0.7071
|  -0.7071    0.7071

Diagonal Matrix (D=N*T*A*N) :
  8    0
    0   18

Canonical form :
8*a*conj(a) + 18*b*conj(b)

>>

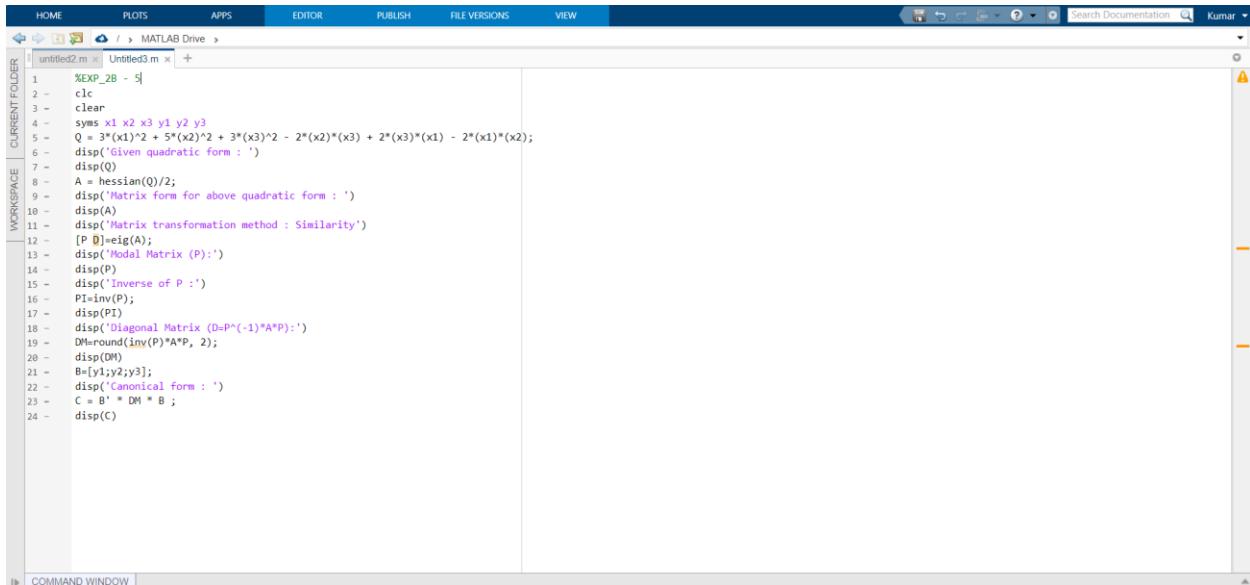
```

Figure 6 : 4 - Command Window output for Orthogonal Transformation

5. Transform the quadratic form $3x_1^2 + 5x_2^2 + 3x_3^2 - 2x_2x_3 + 2x_3x_1 - 2x_1x_2$ to canonical form and specify the matrix of transformation.

Sol:

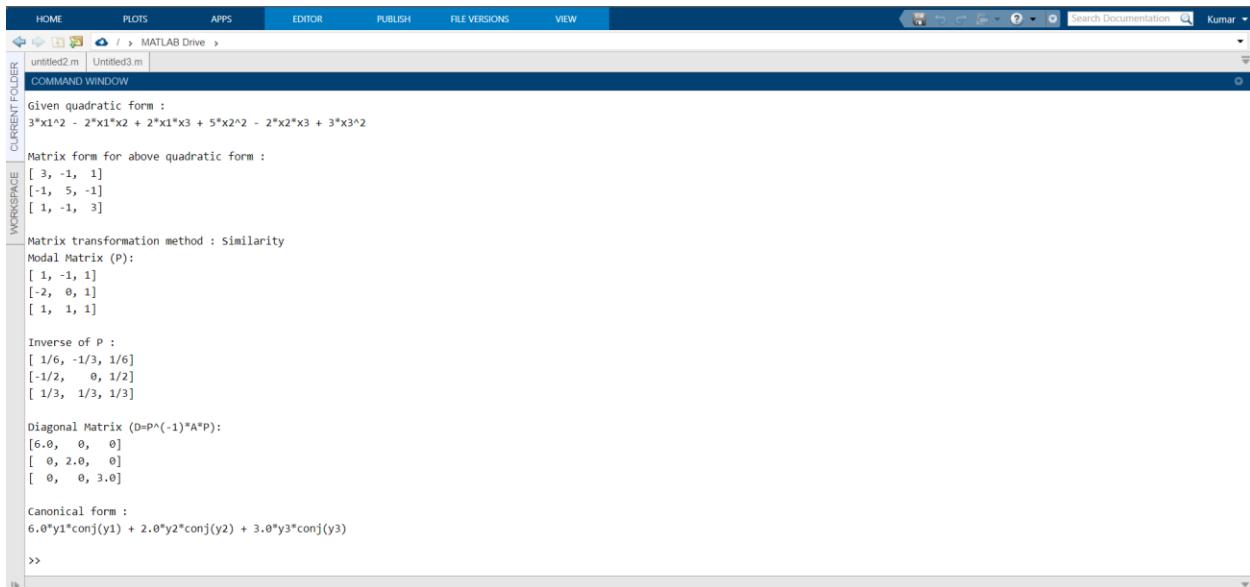
Similarity Transformation



The screenshot shows the MATLAB Editor window. The current file is 'untitled2.m'. The code performs the following steps:

- Defines symbols $x_1, x_2, x_3, y_1, y_2, y_3$.
- Creates a quadratic form $Q = 3*(x1)^2 + 5*(x2)^2 + 3*(x3)^2 - 2*(x2)*(x3) + 2*(x3)*(x1) - 2*(x1)*(x2)$.
- Prints the given quadratic form.
- Computes the Hessian matrix $A = \text{hessian}(Q)/2$.
- Prints the matrix form for above quadratic form.
- Prints the matrix transformation method: Similarity.
- Computes the eigenvalues and eigenvectors of A to find the Modal Matrix P .
- Prints the Modal Matrix P .
- Computes the inverse of P .
- Prints the Inverse of P .
- Computes the Diagonal Matrix $D = P^{-1} * A * P$.
- Prints the Diagonal Matrix D .
- Computes the matrix $B = [y_1; y_2; y_3]$.
- Prints the Canonical form.
- Computes $C = B^T * D * B$.
- Prints the Canonical form.

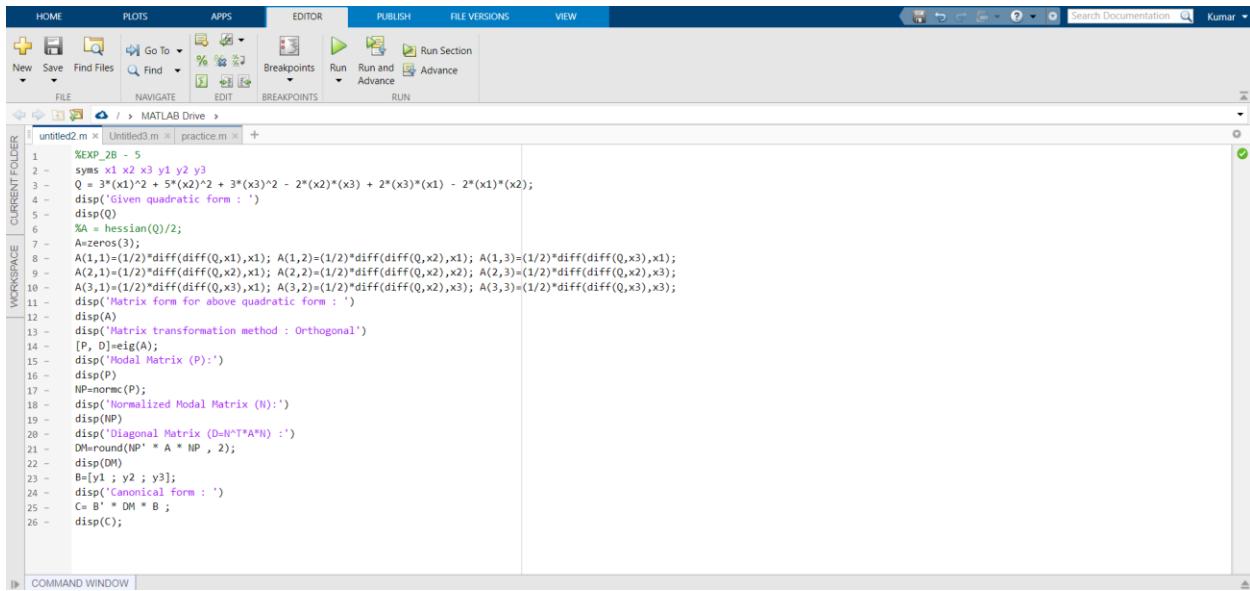
Figure 7 : 5 - MATLAB Code for Similarity Transformation



The screenshot shows the MATLAB Command Window. The output is identical to the code shown in Figure 7, displaying the steps and results of the similarity transformation for the given quadratic form.

Figure 8 : 5 - Command Window output for Similarity Transformation

Orthogonal Transformation



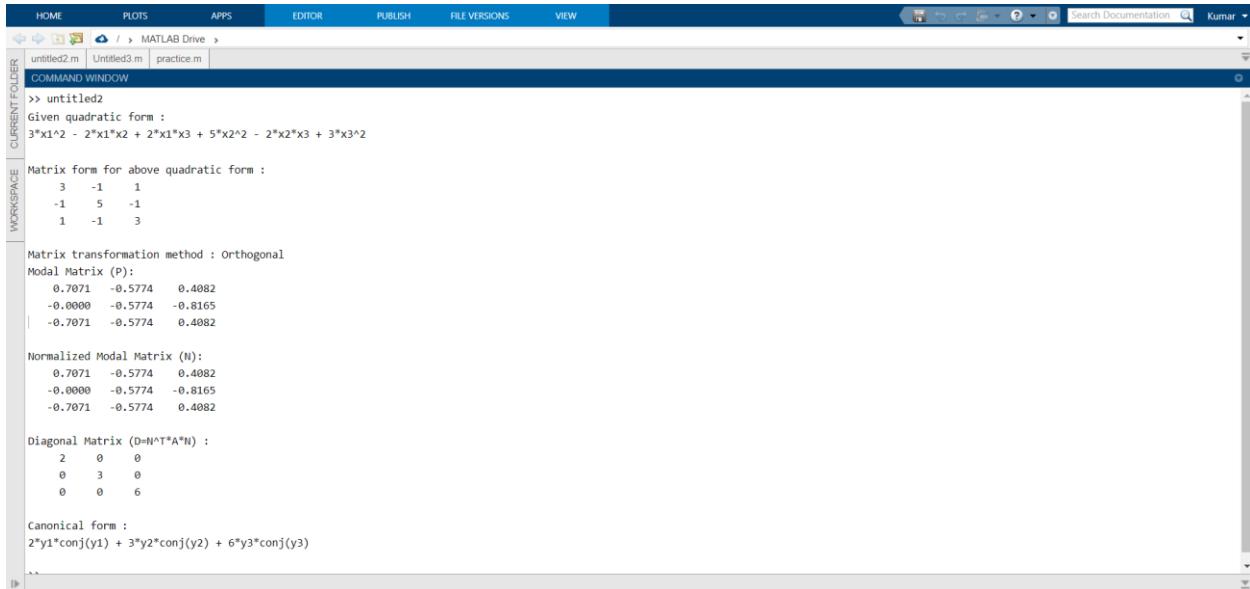
The screenshot shows the MATLAB desktop interface. The menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. The toolbar has options like New, Save, Find Files, Go To, Breakpoints, Run, and Advance. The current workspace contains three files: untitled2.m, Untitled3.m, and practice.m. The untitled2.m file is open in the editor, displaying the following MATLAB code:

```

1 %EXP_2B - 5
2
3 syms x1 x2 x3 y1 y2 y3
4 Q = 3*(x1)^2 + 5*(x2)^2 + 3*(x3)^2 - 2*(x2)*(x3) + 2*(x3)*(x1) - 2*(x1)*(x2);
5 disp('Given quadratic form : ')
6 disp(Q)
7 A=zeros(3);
8 A(1,1)=(1/2)*diff(diff(Q,x1),x1); A(1,2)=(1/2)*diff(diff(Q,x2),x1); A(1,3)=(1/2)*diff(diff(Q,x3),x1);
9 A(2,1)=(1/2)*diff(diff(Q,x2),x1); A(2,2)=(1/2)*diff(diff(Q,x2),x2); A(2,3)=(1/2)*diff(diff(Q,x2),x3);
10 A(3,1)=(1/2)*diff(diff(Q,x3),x1); A(3,2)=(1/2)*diff(diff(Q,x2),x3); A(3,3)=(1/2)*diff(diff(Q,x3),x3);
11 disp('Matrix form for above quadratic form : ')
12 disp(A)
13 disp('Matrix transformation method : Orthogonal')
14 [P, D]=eig(A);
15 disp('Modal Matrix (P):')
16 disp(P)
17 NP=normc(P);
18 disp('Normalized Modal Matrix (N):')
19 disp(NP)
20 disp('Diagonal Matrix (D=N^T*A*N):')
21 DM=round(NP' * A * NP , 2);
22 disp(DM)
23 B=[y1 ; y2 ; y3];
24 disp('Canonical form : ')
25 C= B' * DM * B ;
26 disp(C);

```

Figure 9 : 5 - MATLAB Code for Orthogonal Transformation



The screenshot shows the MATLAB Command Window. The workspace contains three files: untitled2.m, Untitled3.m, and practice.m. The command window displays the following output:

```

>> untitled2
Given quadratic form :
3*x1^2 - 2*x1*x2 + 2*x1*x3 + 5*x2^2 - 2*x2*x3 + 3*x3^2

Matrix form for above quadratic form :
 3   -1   1
 -1   5  -1
 1  -1   3

Matrix transformation method : Orthogonal
Modal Matrix (P):
 0.7071 -0.5774  0.4082
 -0.0000 -0.5774 -0.8165
 -0.7071 -0.5774  0.4082

Normalized Modal Matrix (N):
 0.7071 -0.5774  0.4082
 -0.0000 -0.5774 -0.8165
 -0.7071 -0.5774  0.4082

Diagonal Matrix (D=N^T*A*N) :
 2   0   0
 0   3   0
 0   0   6

Canonical form :
2*y1*conj(y1) + 3*y2*conj(y2) + 6*y3*conj(y3)

```

Figure 10 : 5 - Command Window output for Orthogonal Transformation

Department of Mathematics
School of Advanced Sciences

MAT 2002 – Applications of Differential and Difference Equations (MATLAB)
Experiment 3–A

Solution of a Linear differential equation by method of variation of parameters.

Method of variation of parameters:

We consider a second order linear differential equation of the form

$$F(D)y \equiv \frac{d^2y}{dx^2} + p(x)\frac{dy}{dx} + q(x)y = f(x) \quad (1)$$

Let the solutions of the homogeneous problem of $F(D)y = 0$ be $y_1(x)$ and $y_2(x)$.

Then the Complementary function (solution of the homogeneous problem) of (1) is

$$y_c(x) = C_1 y_1 + C_2 y_2 \quad (2)$$

Then by the method of variation of parameters the particular integral of (1) is of the form

$$y_p(x) = u y_1 + v y_2 \quad (3)$$

where the parameters C_1, C_2 of (2) are replaced with functions $u(x), v(x)$ given by

$$u(x) = -\int \frac{y_2 f(x)}{W(x)} dx \text{ and } v(x) = \int \frac{y_1 f(x)}{W(x)} dx,$$

where the wronskian $W(x) = \begin{vmatrix} y_1 & y_2 \\ y_1' & y_2' \end{vmatrix} = y_1 y_2' - y_2 y_1' \neq 0$.

In this experiment we consider the coefficients p, q to be constants.

MATLAB Code:

```
clear all
close all
clc
syms c1 c2 x m
F=input('Enter the coefficients [a,b,c]: ');
f=input('Enter the RHS function f(x): ');
a=F(1);b=F(2);c=F(3);
AE=a*m^2+b*m+c; % Auxilliary Equation
m=solve(AE);
m1=m(1); m2=m(2);
D=b^2-4*a*c;
if (D>0) % Roots are real and different
    y1=exp(m1*x);y2=exp(m2*x);
elseif (D==0)% Roots are real and equal
```

```

y1=exp(m1*x); y2=x*exp(m1*x);
else % Roots are complex
    alfa=real(m1); beta=imag(m1);
    y1=exp(alfa*x)*cos(beta*x);
    y2=exp(alfa*x)*sin(beta*x);
end
yc=c1*y1+c2*y2; % Complimentary Solution
%%% Particular Integral by Method of variation of parameters.
fx=f/a;
W=y1*diff(y2,x)-y2*diff(y1,x); %%% Wronskian%%
u=int(-y2*fx/W,x);
v=int(y1*fx/W,x);
yp=y1*u+y2*v; %%%Particular Integral%%
y_gen=yc+yp; %%%General Solution%%
check=input('If the problem has initial conditions then enter
1 else enter 2: ');
if(check==1)
cn=input('Enter the initial conditions [x0, y(x0), Dy(x0)]:');
dy_gen=diff(y_gen);
eq1=(subs(y_gen,x,cn(1))-cn(2));
eq2=(subs(dy_gen,x,cn(1))-cn(3));
[c1 c2]=solve(eq1,eq2);
y=simplify(subs(y_gen));
disp('The complete solution is');
disp(y);
ezplot(y, [cn(1),cn(1)+2]);
else
y=simplify(y_gen);
disp('The General Solution is ');
disp(y);
end

```

Example 1. Find the general solution of the differential equation $y'' - 4y = e^{2x}$.

```

Enter the coefficients [a,b,c]: [1 0 -4]
Enter the RHS function f(x): exp(2*x)
If the problem has initial conditions then enter 1 else enter 2: 2
The General Solution is
(exp(-2*x)*(16*c1 - exp(4*x) + 4*x*exp(4*x) + 16*c2*exp(4*x)))/16

```

Example 2. Find the general solution of the differential equation $y'' + y = \sec x \tan x$.

```

Enter the coefficients [a,b,c]: [1 0 1]
Enter the RHS function f(x): sec(x)*tan(x)
If the problem has initial conditions then enter 1 else enter 2: 2
The General Solution is
(log(tan(x)^2 + 1)*sin(x))/2 - sin(x) + c1*cos(x) - c2*sin(x) + x*cos(x)

```

Example 3: Suppose that a spring with a mass of 2 kg. has natural length 0.5 m. A force of 25.6 N is required to maintain it stretched to a length of 0.2 m. The spring is immersed in a fluid with damping constant $c=40$. Find the position of the mass at any time t if it starts from the equilibrium position and is given a push to start with an initial velocity of 0.6 m/s.

Solution: The differential equation pertaining the motion of the spring described by the

$$\text{differential equation } m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + ky = 0.$$

Given mass $m=2$, spring constant $k=\frac{25.6}{0.2}=128$, damping constant $c=40$.

Therefore, the differential equation is $\frac{d^2x}{dt^2} + 20 \frac{dx}{dt} + 64x = 0$ with $x(0)=0$; $x'(0)=0.6$.

Input/Output:

Enter the coefficients [a,b,c]: [1 20 64]

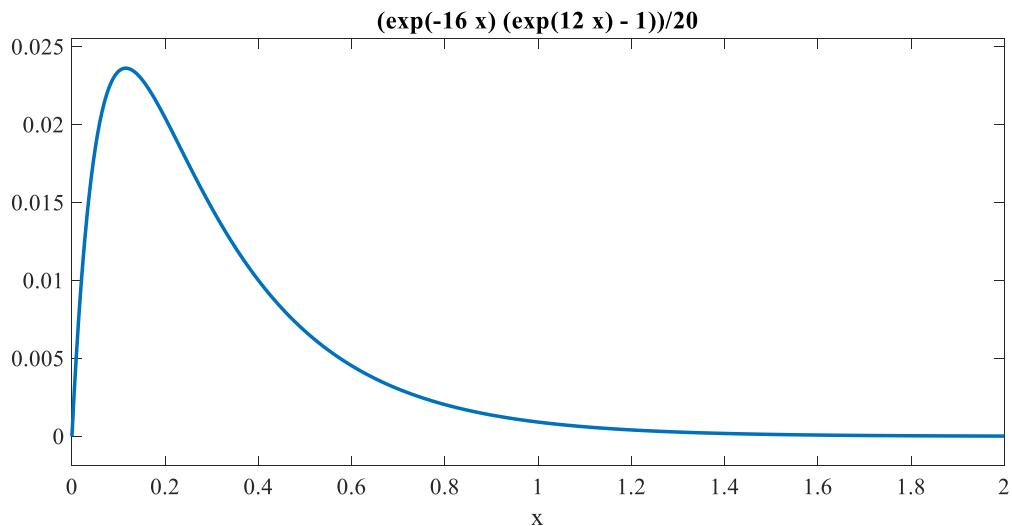
Enter the RHS function f(x): 0

If the problem has initial conditions then enter 1 else enter 2: 1

Enter the initial conditions [x0, y(x0), Dy(x0)]:[0 0 0.6]

The complete solution is

$$(\exp(-16*x) * (\exp(12*x) - 1)) / 20$$

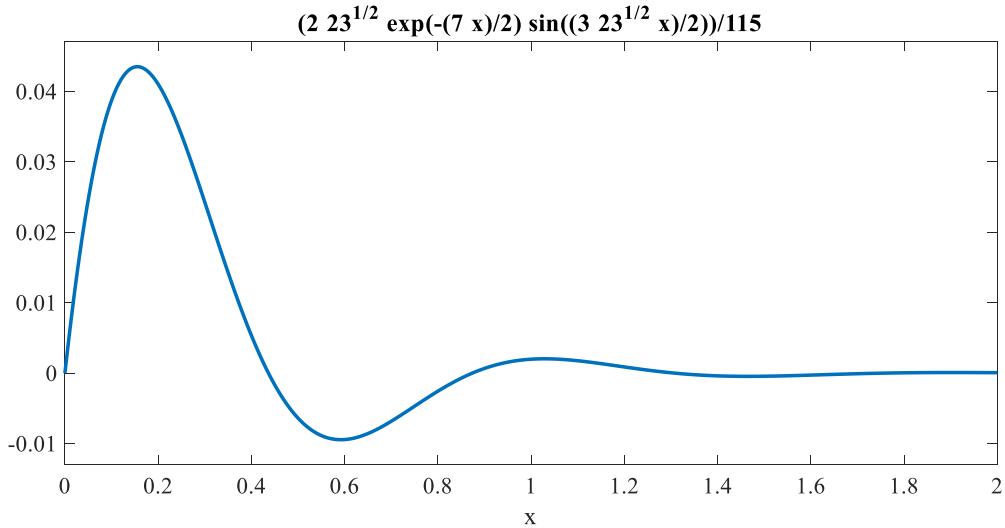


Inference: Here $c^2 - 4mk > 0$. Hence it is a case of over damping, therefore no oscillations occur.

Example 4: Consider the above problem with the spring constant $c=14$, other parameters being the same.

The spring constant here is $k=128$. Damping constant $c=14$, the mass $m=2$.

So the differential equation becomes $\frac{d^2x}{dt^2} + 7\frac{dx}{dt} + 64x = 0$, with $x(0) = 0$; $x'(0) = 0.6$.



Input/Output:

```
Enter the coefficients [a,b,c]: [1 7 64]
Enter the RHS function f(x): 0
If the problem has initial conditions then enter 1 else enter 2: 1
Enter the initial conditions [x0, y(x0), Dy(x0)]:[0 0 0.6]
The complete solution is
(2*23^(1/2)*exp(-(7*x)/2)*sin((3*23^(1/2)*x)/2))/115
```

Inference: Here $c^2 - 4mk < 0$. It is a case of under damping. Some oscillations occur.

Example 5: Find the charge in the RLC circuit at time t in the circuit when a resistance of 40Ω , inductance of $1H$ and a capacitance of $16 \times 10^{-4}F$ are connected in series with a source of voltage $E(t) = 100\cos 10t$, given that initially the charge and current are both 0.

The differential equation for the RLC circuit is

$$LQ'' + RQ' + \frac{1}{C}Q = E(t)$$

Here $R = 40$, $L = 1$, $C = 16 \times 10^{-4}$, $E(t) = 100\cos 10t$.

Therefore, the differential equation is

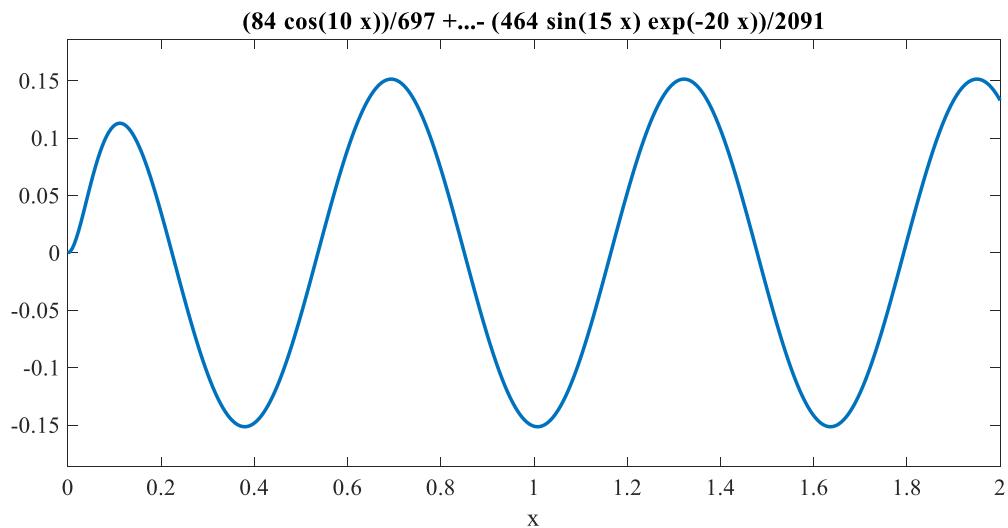
$$Q'' + 40Q' + 625Q = 100\cos 10t \text{ with } Q(0) = 0; Q'(0) = 0.$$

Input/Output

```
Enter the coefficients [a,b,c]: [1,40,625]
Enter the RHS function f(x): 100*cos(10*x)
If the problem has initial conditions then enter 1 else enter 2: 1
Enter the initial conditions [x0, y(x0), Dy(x0)]:[0,0,0]
```

The complete solution is

$$(84 \cos(10x)) / 697 + (64 \sin(10x)) / 697 - (84 \cos(15x) \exp(-20x)) / 697 - (464 \sin(15x) \exp(-20x)) / 2091$$



Exercise:

1. Find the general solution of the differential equation $y'' - 2y' = e^x \sin x$.
2. Solve the initial value problem

$$y'' + 4y' + 20y = 23\sin x - 15\cos x, \quad y(0) = 0, \quad y'(0) = -1.$$

3. Find the current $I(t)$ in an RLC circuit with $R=11\Omega$, $L=0.1$ H, $C=10^{-2}$ F, which is connected to a source of voltage $E(t) = 100 \sin 400t$. Assume that the current and the charge are zero when $t=0$.
4. A spring with mass of 2kg has damping constant 14, and a force of 6N is required to keep the spring stretched 0.5 m beyond its natural length. The spring is stretched 1m beyond its natural length and then released with zero velocity. Find the position of the mass at any time t .

MAT2002 – Applications of Differential & Difference Equations

Solution of a Linear differential equation by method of variation of parameters.

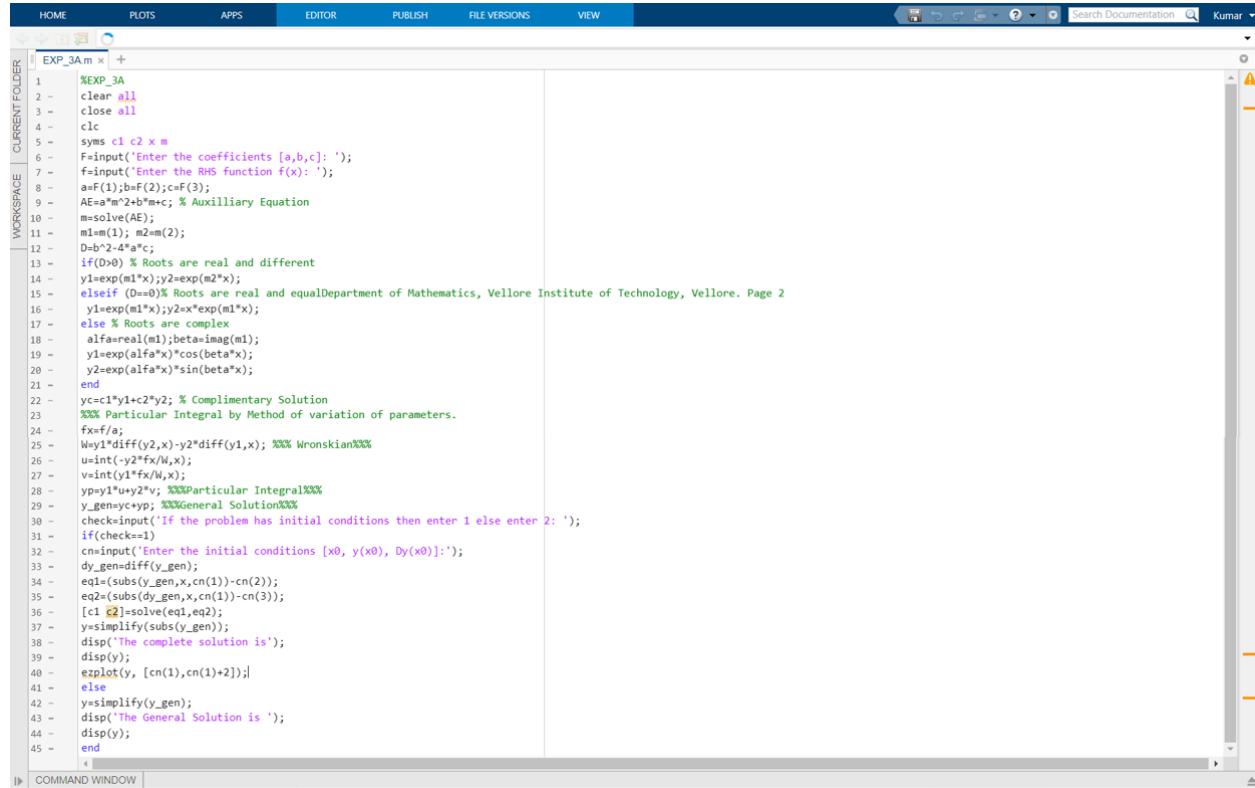
(Exp – 3A)

Faculty Name -
Monica C

Presented By -
Kumar Sparsh
19BCB0025

1. Find the general solution of the differential equation $y'' - 2y' = e^x \sin x$.

Sol:



The screenshot shows the MATLAB desktop interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. A search bar at the top right says "Search Documentation" with "Kumar" typed in. The central workspace shows a script named "EXP_3A.m" with the following code:

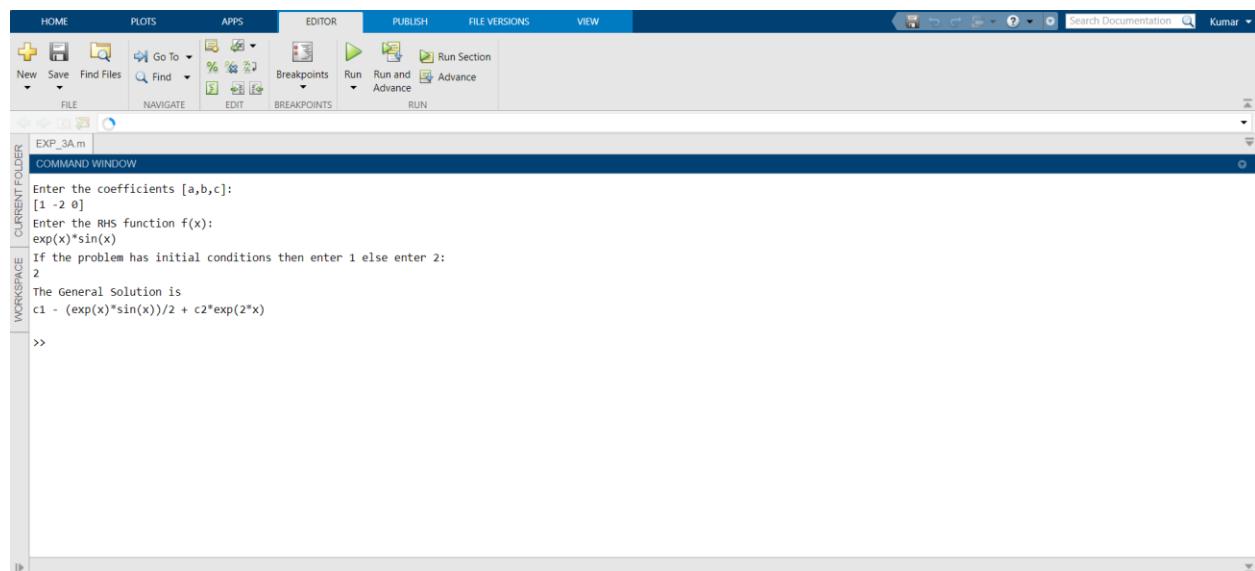
```

EXP_3A.m
1 %EXP_3A
2 clear all
3 close all
4 clc
5 c1 c2 x =
6 F=input('Enter the coefficients [a,b,c]: ');
7 f=input('Enter the RHS function f(x): ');
8 a=F(1);b=F(2);c=F(3);
9 Ae=a^2+b*m*c; % Auxilliary Equation
10 m=solve(Ae);
11 m1=m(1);m2=m(2);
12 D=b^2-4*a*c;
13 if(D>0) Roots are real and different
14 y1=exp(m1*x);y2=exp(m2*x);
15 elseif (D=0) Roots are real and equal
16 else Roots are complex
17 alfa=real(m1);beta=imag(m1);
18 y1=exp(alfa*x)*cos(beta*x);
19 y2=exp(alfa*x)*sin(beta*x);
20 end
21 yc=c1*y1+c2*y2; % Complimentary Solution
22 %% Particular Integral by Method of variation of parameters.
23 fx=f;
24 W=y1*diff(y2,x)-y2*diff(y1,x); %% Wronskian%%
25 u=int(-y2*fx/W,x);
26 v=int(y1*fx/W,x);
27 ypw1=u*y2*v; %%Particular Integral%%
28 y_gen=y+ypw1;
29 disp('General Solution');
30 check=input('If the problem has initial conditions then enter 1 else enter 2: ');
31 if(check==1)
32 cn=input('Enter the initial conditions [x0, y(x0), Dy(x0)]: ');
33 dy_gen=diff(y_gen);
34 eq1=subs(dy_gen,x,cn(1))-cn(2);
35 eq2=subs(dy_gen,x,cn(1))-cn(3));
36 [c1 c2]=solve(eq1,eq2);
37 y=simplify(subs(y_gen));
38 disp('The complete solution is');
39 disp(y);
40 ezplot(y, [cn(1),cn(1)+2]);
41 else
42 y=simplify(y_gen);
43 disp('The General Solution is ');
44 disp(y);
45 end

```

The command window at the bottom shows the output of the code.

Figure 1 : MATLAB Code for all following questions



The screenshot shows the MATLAB desktop interface with the command window active. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. A search bar at the top right says "Search Documentation" with "Kumar" typed in. The command window displays the following text:

```

New Save Find Files Go To % & Breakpoints Run Run and Advance
FILE NAVIGATE BREAKPOINTS RUN
EXP_3A.m
COMMAND WINDOW
Enter the coefficients [a,b,c]:
[1 -2 0]
Enter the RHS function f(x):
exp(x)*sin(x)
If the problem has initial conditions then enter 1 else enter 2:
2
The General Solution is
c1 - (exp(x)*sin(x))/2 + c2*exp(2*x)
>>

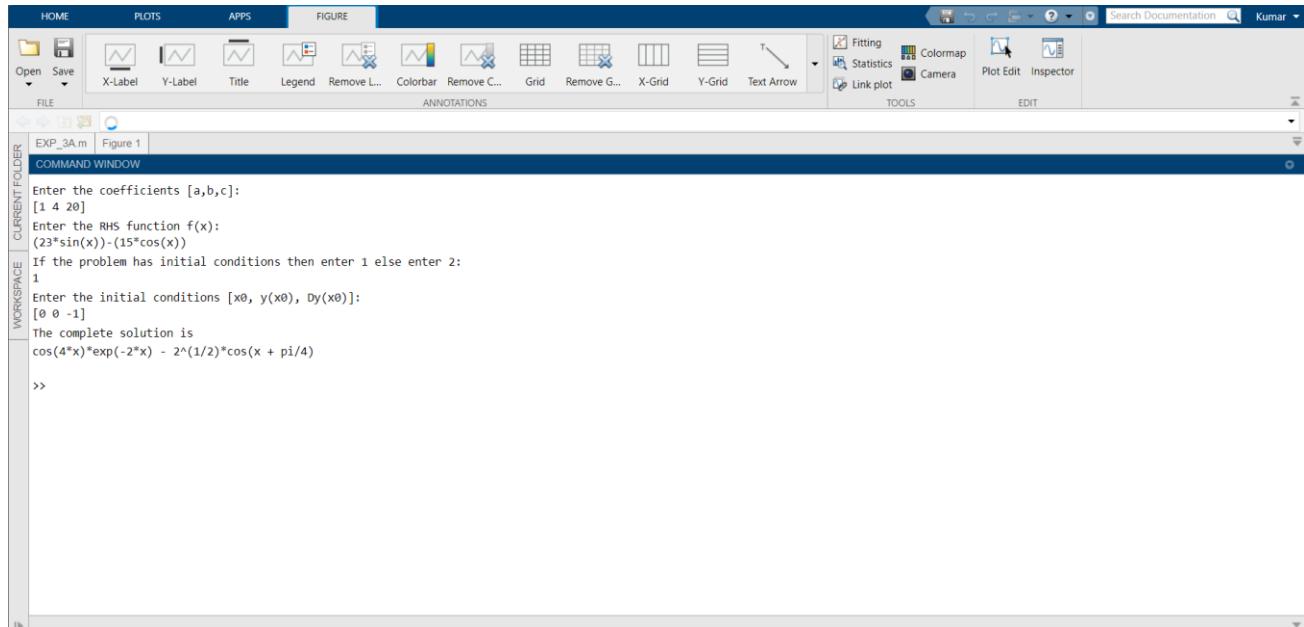
```

Figure 2 : Command Window Output

2. Solve the initial value problem

$$y'' + 4y' + 20y = 23\sin x - 15\cos x, \quad y(0) = 0, \quad y'(0) = -1.$$

Sol:



The screenshot shows the MATLAB Command Window interface. The workspace contains variables `a`, `b`, `c`, `f`, `x0`, `y0`, and `Dy0`. The command window displays the following text:

```

Enter the coefficients [a,b,c]:
[1 4 20]
Enter the RHS function f(x):
(23*sin(x))-(15*cos(x))
If the problem has initial conditions then enter 1 else enter 2:
1
Enter the initial conditions [x0, y(x0), Dy(x0)]:
[0 0 -1]
The complete solution is
cos(4*x)*exp(-2*x) - 2^(1/2)*cos(x + pi/4)
>>

```

Figure 3 : Command Window Output

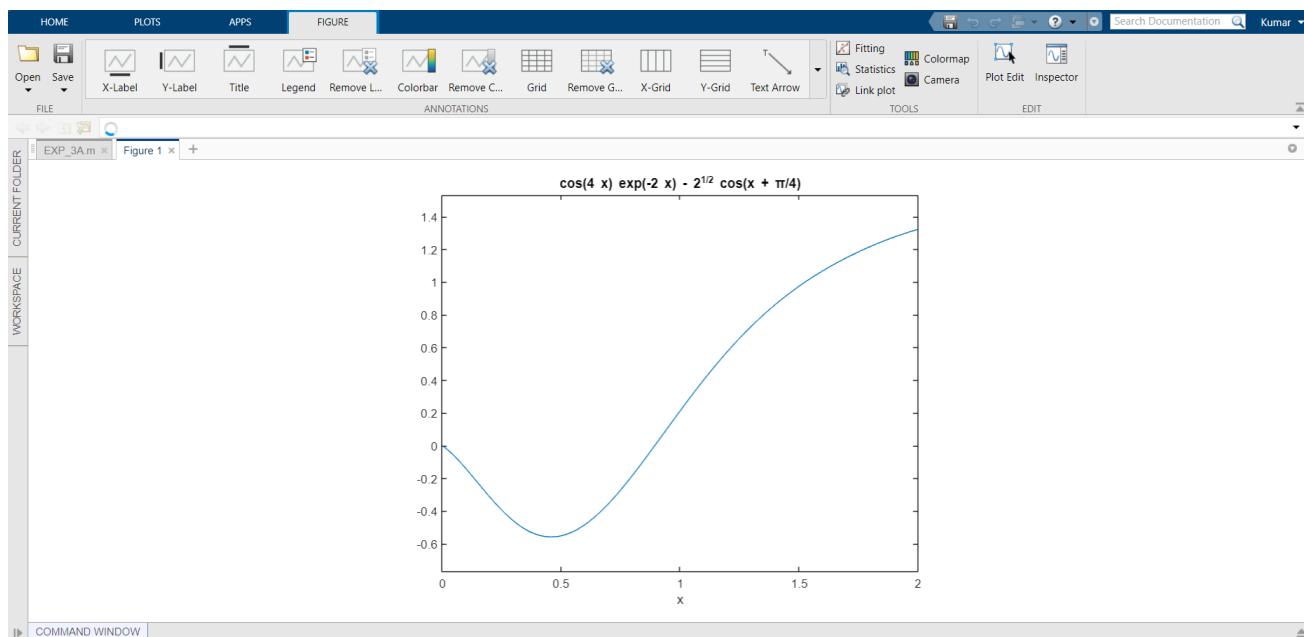


Figure 4 : Graph Output

3. Find the current $I(t)$ in an RLC circuit with $R=11\Omega$, $L=0.1$ H, $C=10^{-2}$ F , which is connected to a source of voltage $E(t) = 100 \sin 400t$. Assume that the current and the charge are zero when $t=0$.

Sol:

$$R = 11 \Omega$$

$$L = 0.1 \text{ H}$$

$$C = 10^{-2} \text{ F}$$

$$E(t) = 100 \sin 400t$$

$$LQ'' + RQ' + \frac{1}{C} Q = E(t) \quad \text{--- eq for RLC circuit}$$

$$(0.1)Q'' + (11)Q' + (100)Q = 100 \sin(400t)$$

with, $t=0 \Rightarrow x_0 = 0$

~~charge~~ is zero at $t=0 \Rightarrow y(x_0) = 0 \Rightarrow q = 0$

current is zero at $t=0 \Rightarrow Dy(x_0) = 0 \Rightarrow \frac{dq}{dt} = 0$

Figure 5 : Differential Equation formation

```

HOME PLOTS APPS FIGURE
Open Save X-Label Y-Label Title Legend Remove L... Colorbar Remove C... Grid Remove G... X-Grid Y-Grid Text Arrow
FILE ANNOTATIONS Fitting Statistics Colormap Link plot Camera Plot Edit Inspector
Search Documentation Kumar
EXP_3A.m Figure 1
CURRENT FOLDER
COMMAND WINDOW
Enter the coefficients [a,b,c]:
[0.1 11 100]
Enter the RHS function f(x):
100*(sin(400*x))
If the problem has initial conditions then enter 1 else enter 2:
1
Enter the initial conditions [x0, y(x0), Dy(x0)]:
[0 0 0]
The complete solution is
(400*exp(-10*x))/14409 - (44*cos(400*x))/27217 - (4*exp(-100*x))/153 - (159*sin(400*x))/27217
>> |

```

Figure 6 : Command Window Output

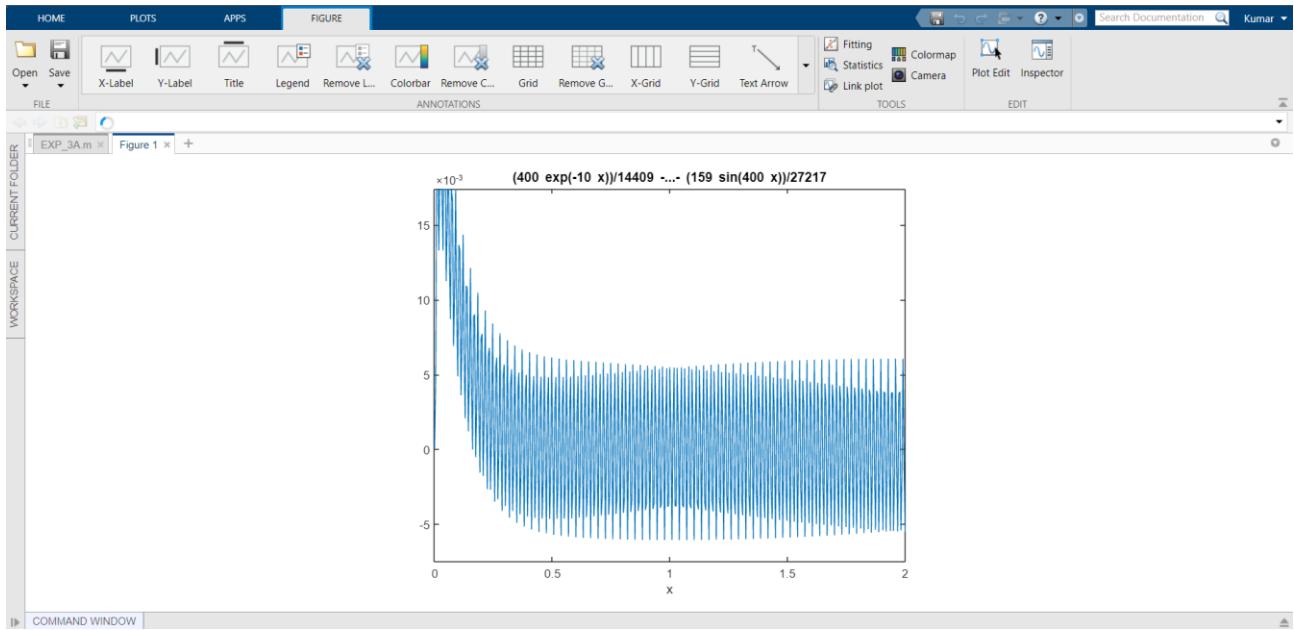


Figure 7 : Graph Output

4. A spring with mass of 2kg has damping constant 14, and a force of 6N is required to keep the spring stretched 0.5 m beyond its natural length. The spring is stretched 1m beyond its natural length and then released with zero velocity. Find the position of the mass at any time t.

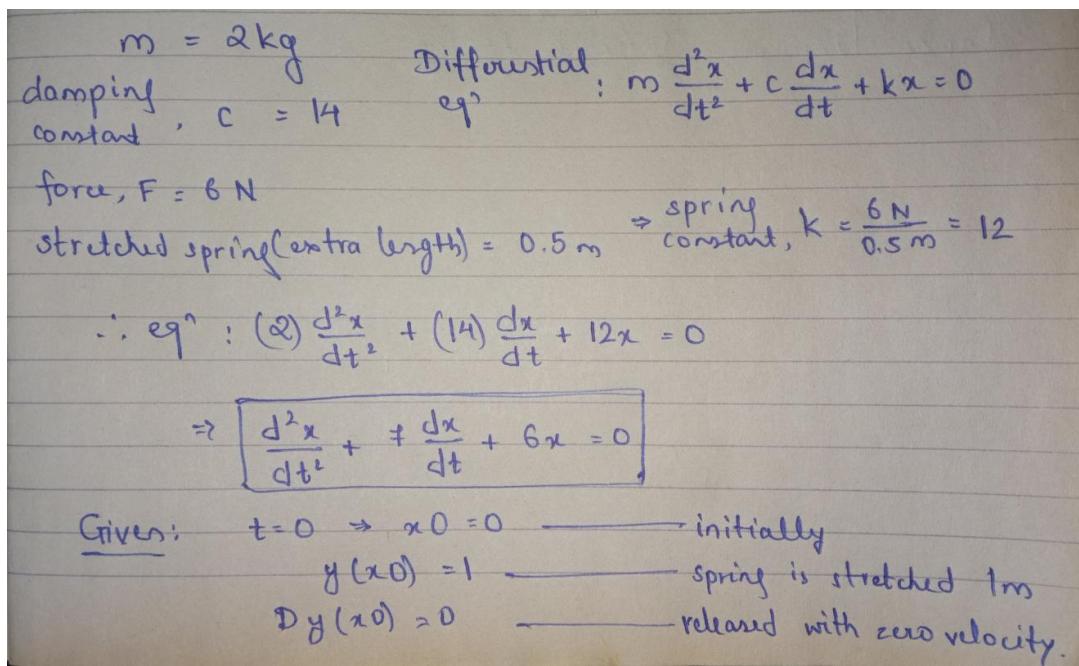
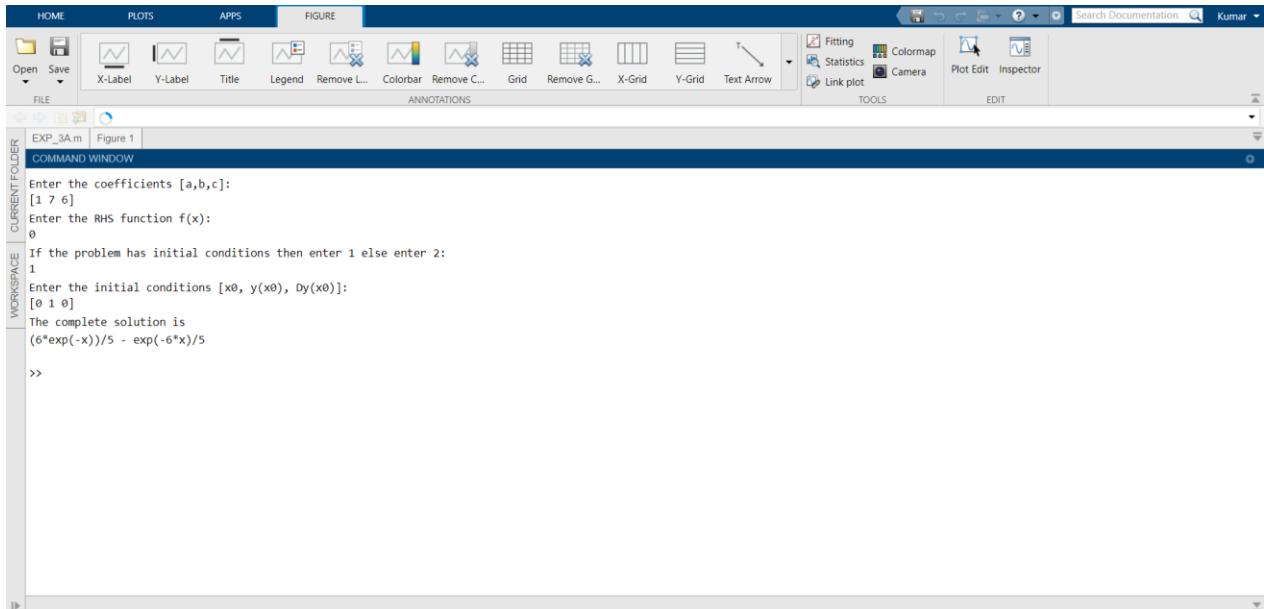
Sol:

Figure 8 : Differential Equation formation



The screenshot shows the MATLAB Command Window interface. The menu bar includes HOME, PLOTS, APPS, and FIGURE. The toolbar contains icons for Open, Save, X-Label, Y-Label, Title, Legend, Remove L..., Colorbar, Remove C..., Grid, Remove G..., X-Grid, Y-Grid, Text Arrow, Fitting, Statistics, Colormap, Camera, Plot Edit, and Inspector. The current workspace is EXP_3A.m, and the figure is Figure 1. The command window displays the following text:

```

Enter the coefficients [a,b,c]:
[1 7 6]
Enter the RHS function f(x):
0
If the problem has initial conditions then enter 1 else enter 2:
1
Enter the initial conditions [x0, y(x0), Dy(x0)]:
[0 1 0]
The complete solution is
(6*exp(-x))/5 - exp(-6*x)/5

>>

```

Figure 9 : Command Window Output

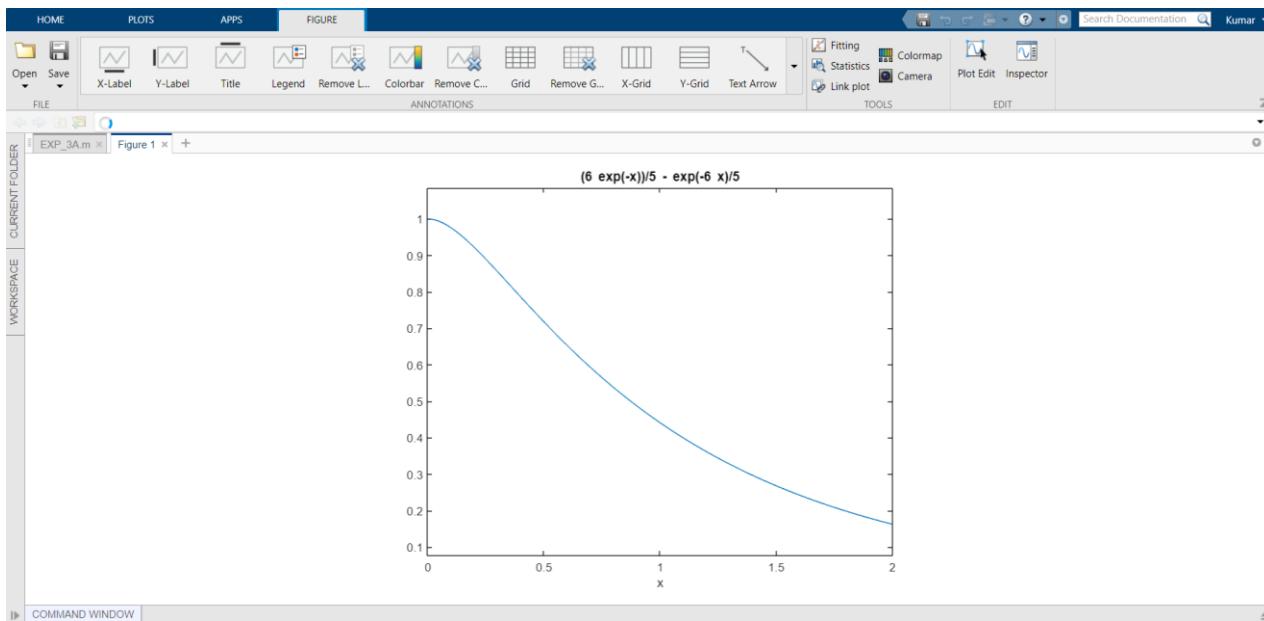


Figure 10 : Graph Output

Department of Mathematics
School of Advanced Sciences

MAT 2002 – Applications of Differential and Difference Equations (MATLAB)
Experiment 3-B

Solution of Linear differential equations by Laplace transforms

The Laplace Transform of a function $f(t)$ is defined as

$$F(s) = L[f(t)] = \int_0^{\infty} e^{-st} f(t) dt , \text{ provided the integral exists.}$$

MATLAB Commands used

Command	Purpose
laplace(f)	To find the Laplace transform of a scalar symbol f with default independent variable t. The default return is a function of s.
ilaplace(F)	To find the inverse Laplace transform of the scalar symbolic object F with default independent variable s. The default return is a function of t.
heaviside(t-a)	To input the heaviside's unit step function H(t-a).
dirac(t-a)	To input the dirac delta function δ(t-a).
collect(P, var)	Rewrites P in terms of the powers of the variable var

Example 1. The following MATLAB code finds the Laplace transform of $f(t)$.

```
clear all
clc
syms t
f=input('Enter the function of t: ');
F=laplace(f);
disp(['L{f(t)}=' , char(F)]);
```

Input/Output:

Enter the function of t: sin(t)
 $L\{f(t)\}=1/(s^2 + 1)$

Example 2: The following MATLAB code computes the Laplace Transform of

$$f(t) = \begin{cases} t^2, & t < 2 \\ t-1, & 2 < t < 3 \\ 7, & t > 3 \end{cases}$$

Input/Output:

Enter the function of t: $t^2 * (\text{heaviside}(t) - \text{heaviside}(t-2)) + (t-1) * (\text{heaviside}(t-2) - \text{heaviside}(t-3)) + 7 * \text{heaviside}(t-3)$
 $L\{f(t)\} = (7 * \exp(-3*s)) / s - (4 * \exp(-2*s)) / s - (4 * \exp(-2*s)) / s^2 - (2 * \exp(-2*s)) / s^3 + 2 / s^3 - (\exp(-3*s) * (2 * s - \exp(s) - s * \exp(s) + 1)) / s^2$

Example 3. The following MATLAB code computes the inverse Laplace transform of $F(s)$

```
syms s
F=input('Enter the function of s: ');
f=ilaplace(F);
disp(['f(t)=',char(f)]);
```

Input/Output

Enter the function of s: $6/(s^3+2*s^2-s-2)$

$f(t)=2*\exp(-2*t)-3*\exp(-t)+\exp(t)$

To solve and visualize solutions of a second order Linear differential equation using Laplace transform.

Working Procedure:

- Input the differential equation coefficients a,b,c and the RHS function $f(x)$ of the differential equation $ay'' + by' + cy = f(x)$.
- Input the initial conditions $y(0)$ and $y'(0)$.
- Apply Laplace Transform and find $Y(s)$.
- Apply inverse Transform and find $y(t)$.

MATLAB Code

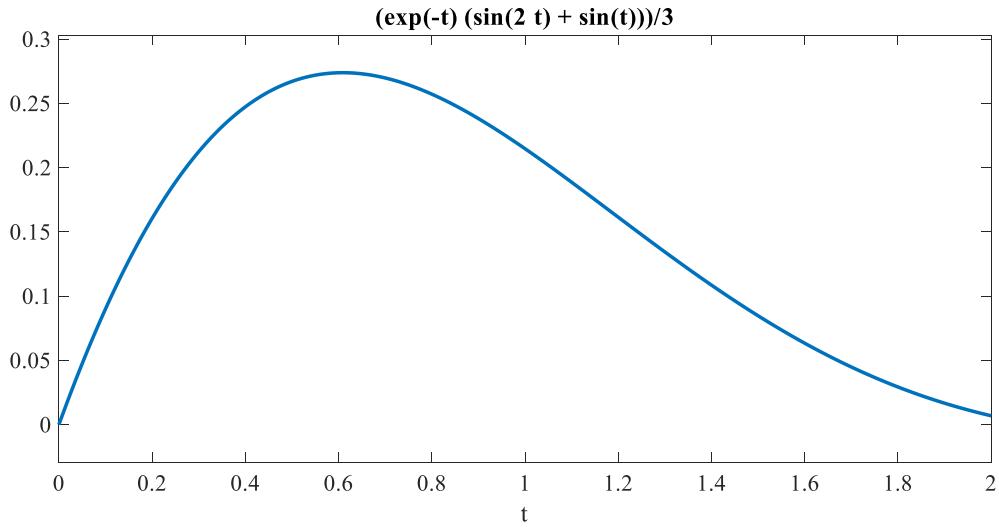
```
clear all
clc
syms t s y(t) Y
dy(t)=diff(y(t));
d2y(t)=diff(y(t),2);
F = input('Input the coefficients [a,b,c]: ');
a=F(1);b=F(2);c=F(3);
nh = input('Enter the non-homogenous part f(x): ');
eqn=a*d2y(t)+b*dy(t)+c*y(t)-nh;
LTY=laplace(eqn,t,s);
IC = input('Enter the initial conditions in the form [y0,Dy(0)]: ');
y0=IC(1);dy0=IC(2);
LTY=subs(LTY,{ 'laplace(y(t), t, s)', 'y(0)', 'D(y)(0)' },{Y,y0,dy0});
eq=collect(LTY,Y);
Y=simplify(solve(eq,Y));
yt=simplify(ilaplace(Y,s,t));
disp('The solution of the differential equation y(t)=')
disp(yt);
ezplot(yt,[y0,y0+2]);
```

Example 4. Solve the initial value problem $\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + 5y = e^{-t} \sin t$, $y(0) = 0$, $y'(0) = 1$.

Input/Output

Input the coefficients [a,b,c]: [1 2 5]
 Enter the non-homogenous part f(x): exp(-t)*sin(t)
 Enter the initial conditions in the form [y0,Dy(0)]: [0,1]

The solution of the differential equation $y(t) =$
 $(\exp(-t) * (\sin(2*t) + \sin(t))) / 3$



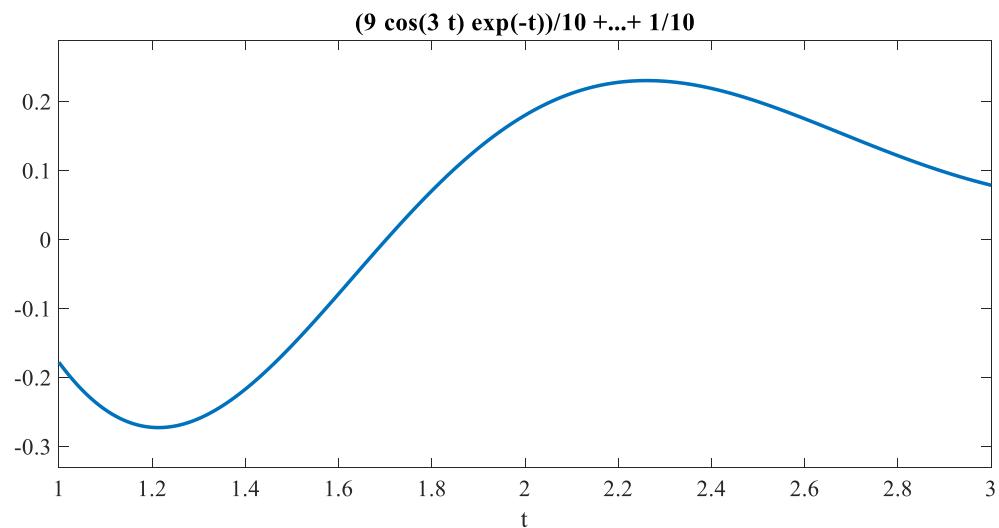
Example 5.

Solve $y'' + 2y' + 10y = 1 + 5\delta(t - 5)$, $y(0) = 1$, $y'(0) = 2$.

Input/Output

Input the coefficients [a,b,c]: [1 2 10]
 Enter the non-homogenous part f(x): 1+5*dirac(t-5)
 Enter the initial conditions in the form [y0,Dy(0)]: [1,2]

The solution of the differential equation $y(t) =$
 $(9*\cos(3*t)*\exp(-t)) / 10 + (29*\sin(3*t)*\exp(-t)) / 30 + (5*\text{heaviside}(t - 5)*\exp(5 - t)*\sin(3*t - 15)) / 3 + 1/10$



Exercise

1. Solve $y'' - 2y' + y = e^t$, subject to $y(0) = 2$, $y'(0) = -1$
2. Solve $y'' + y = f(t)$, $y(0) = 1$, $y'(0) = 0$ where $f(t) = \begin{cases} 3, & t \leq 4 \\ 2t - 5, & t > 4 \end{cases}$.
3. Using Laplace transforms find the current $i(t)$ in the circuit with a resistance $R = 4\Omega$, inductance $L = 1H$, capacitance $C = 0.05F$ connected in a series with a source of voltage $v(t) = \begin{cases} 34e^{-t}, & 0 < t < 4 \\ 0, & t > 4 \end{cases}$ volts.

MAT2002 – Applications of Differential & Difference Equations

**Solution of Linear differential equations by Laplace transforms
(Exp – 3B)**

Faculty Name -
Monica C

Presented By -
Kumar Sparsh
19BCB0025

1. Solve $y'' - 2y' + y = e^t$, subject to $y(0) = 2$, $y'(0) = -1$

Sol:

```
%EXP-3B - 19BCB0025
clear all
clc
syms t s y(t)
dy(t)=diff(y(t));
d2y(t)=diff(y(t),2);
F = input('Input the coefficients [a,b,c]: ');
a=F(1);b=F(2);c=F(3);
nh = input('Enter the non-homogenous part f(x): ');
eqn=a*d2y(t)+b*dy(t)+c*y(t)-nh;
LTY=laplace(eqn,t,s);
IC = input('Enter the initial conditions in the form [y0,Dy(0)]: ');
y0=IC(1),dy0=IC(2);
disp(LTY);
LTY=subs(LTY,[laplace(y(t), t, s),y(0),subs(diff(y(t), t), t, 0)],{Y,y0,dy0});
eq=collect(LTY,Y);
Y=simplify(solve(eq,Y));
yt=simplify(ilaplace(Y,s,t));
disp('The solution of the differential equation y(t)=')
disp(yt);
ezplot(yt,[y0,y0+2]);
```

Figure 1 : MATLAB Code for all following questions

```
New to MATLAB? See resources for Getting Started.

Input the coefficients [a,b,c]:
[1 -2 1]
Enter the non-homogenous part f(x):
exp(t)
Enter the initial conditions in the form [y0,Dy(0)]:
[2 -1]
2*y(0) - 2*s*laplace(y(t), t, s) - 1/(s - 1) - s*y(0) + s^2*laplace(y(t), t, s) - subs(diff(y(t), t), t, 0) + laplace(y(t), t, s)

The solution of the differential equation y(t)=
(exp(t)*(t^2 - 6*t + 4))/2

>>
```

Figure 2 : Command Window Output

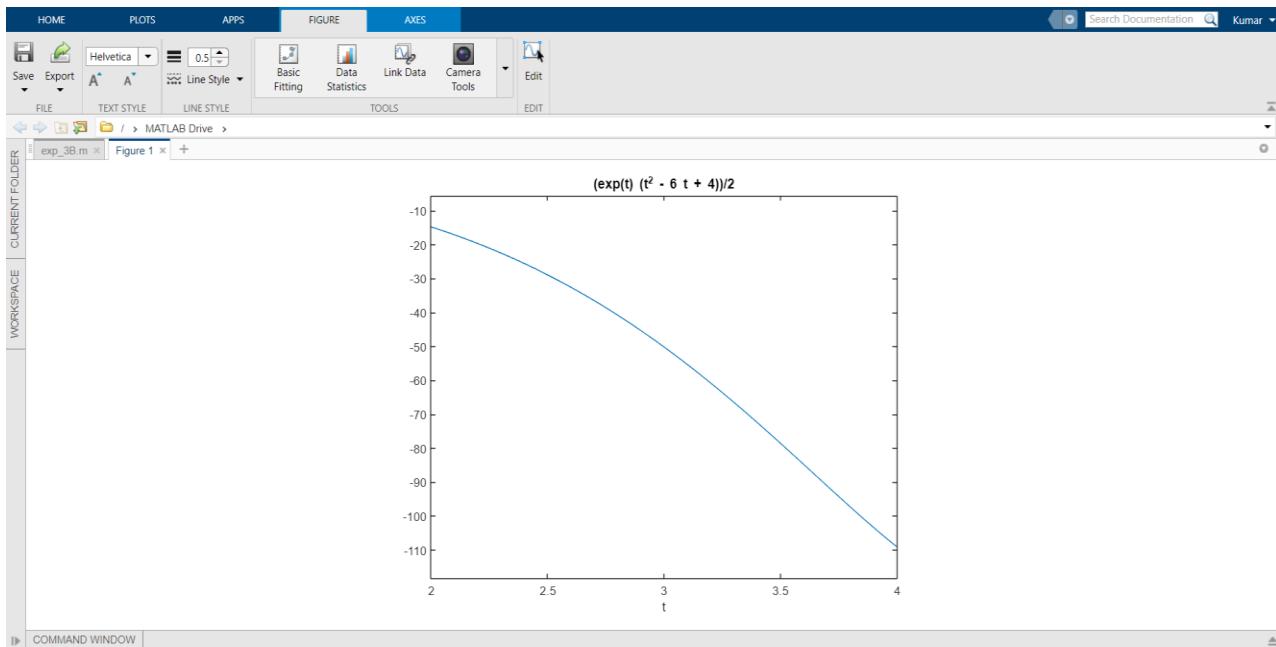


Figure 3 : Graph Output

2. Solve $y'' + y = f(t)$, $y(0) = 1$, $y'(0) = 0$ where $f(t) = \begin{cases} 3, & t \leq 4 \\ 2t - 5, & t > 4 \end{cases}$.

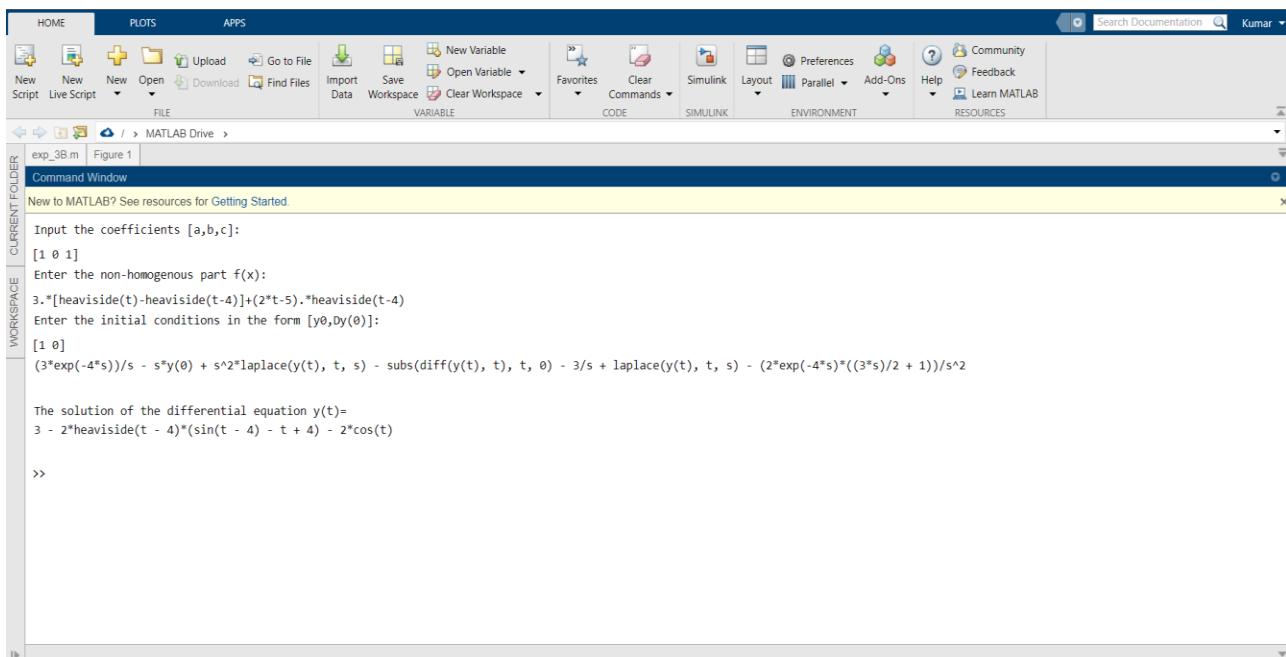
Sol:

Figure 4 : Command Window Output

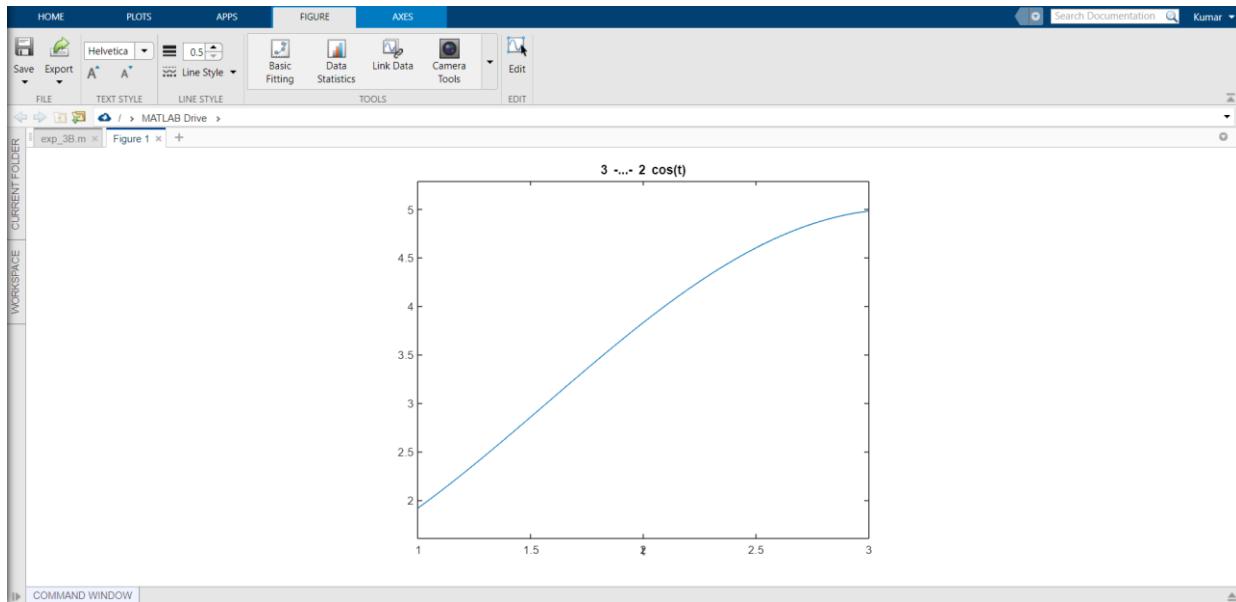


Figure 5 : Graph Output

3. Using Laplace transforms find the current $i(t)$ in the circuit with a resistance $R = 4\Omega$, inductance $L = 1H$, capacitance $C = 0.05F$ connected in a series with a source of

$$\text{voltage } v(t) = \begin{cases} 34e^{-t}, & 0 < t < 4 \\ 0, & t > 4 \end{cases} \text{ volts.}$$

Sol:

```

Input the coefficients [a,b,c]:
[1 4 20]
Enter the non-homogenous part f(x):
34*exp(-t).*[heaviside(t)-heaviside(t-4)]
Enter the initial conditions in the form [y0,Dy0]:
[0 0]
4*s*laplace(y(t), t, s) - 4*y(0) + (34*exp(- 4*s - 4))/(s + 1) - 34/(s + 1) - s*y(0) + s^2*laplace(y(t), t, s) - subs(diff(y(t), t), t, 0) + 20*laplace(y(t), t, s)

The solution of the differential equation y(t)=
2*exp(-t) - 2*exp(-2*t)*(cos(4*t) + sin(4*t)/4) + (heaviside(t - 4)*exp(4 - 2*t)*(4*cos(4*t - 16) - 4*exp(t - 4) + sin(4*t - 16)))/2

>>

```

Figure 6 : Command Window Output

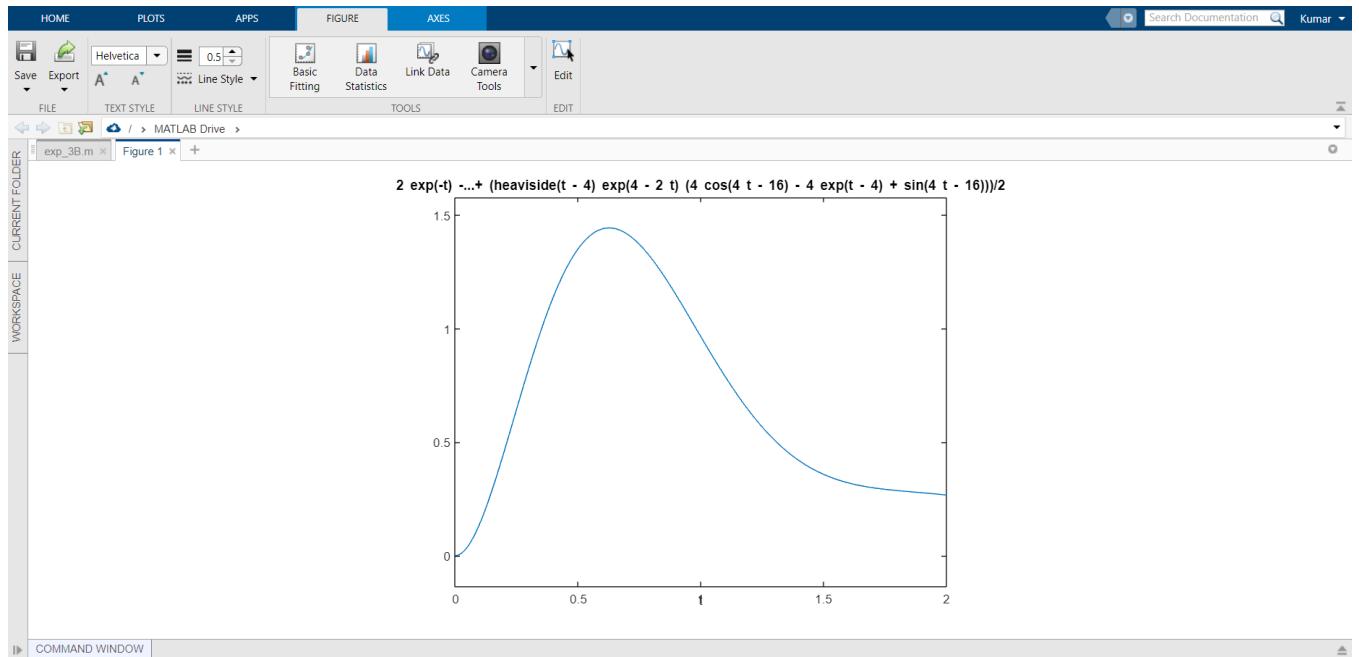


Figure 7 : Graph Output



DEPARTMENT OF MATHEMATICS SCHOOL OF ADVANCED SCIENCES

Solution of homogeneous system of first order and second order differential equations by Matrix method.

Course Code: MAT2002

Course Name: Application of Differential and Difference Equations

Experiment: 4-A

Duration: 90 Minutes

System of First Order Linear Differential Equations

A system of n linear first order differential equations in n unknowns (an $n \times n$ system of linear equations) has the general form:

$$\begin{cases} x'_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + g_1(t) \\ x'_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + g_2(t) \\ \vdots = \vdots + \vdots + \vdots + \vdots + \vdots \\ x'_n = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n + g_n(t) \end{cases} \quad (1)$$

where the coefficients a_{ij} 's are arbitrary constants, and g_i 's are arbitrary functions of t . If every term g_i is constant zero, then the system is said to be homogeneous.

The system (1) is most often given in a shorthand format as a matrix-vector equation, in the form:

$$X' = AX + G$$

where $X' = [x'_i]_{n \times 1}$, $A = [a_{ij}]_{n \times n}$, $X = [x_i]_{n \times 1}$, and $G = [g_i(t)]_{n \times 1}$.

If the coefficient matrix A has two distinct real eigenvalues λ_1 and λ_2 and their respective eigenvectors are X_1 and X_2 , then the 2×2 system

$$X' = AX$$

has a general solution

$$X = C_1 X_1 e^{\lambda_1 t} + C_2 X_2 e^{\lambda_2 t}$$

System of Second Order Linear Differential Equations

Consider the system of second order linear differential equations of the form

$$\begin{cases} x''_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ x''_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots = \vdots + \vdots + \vdots + \vdots + \vdots \\ x''_n = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n \end{cases} \quad (2)$$

where the coefficients a_{ij} 's are arbitrary constants.

Then, the solution of (2), $X'' = AX$, is

$$X = PY$$

where Y is the solution of $Y'' = DY$, P is the modal matrix of A and D is it's diagonal matrix.

1. Solve:

$$\begin{aligned}x'_1 &= x_1 + 2x_2 \\x'_2 &= 0.5x_1 + x_2 \\x_1(0) &= 16, x_2(0) = -2\end{aligned}$$

MATLAB CODE

```
clc
clear

syms t C1 C2

A=input('Enter A: ');
[P,D]=eig(A);
L1=D(1);L2=D(4);

y1=C1*exp(L1*t);y2=C2*exp(L2*t);
Y=[y1;y2];
X=P*Y;

Cond=input('Enter the initial conditions [t0, x10,x20]: ');
t0=Cond(1);x10=Cond(2);x20=Cond(3);

eq1=subs(X(1)-x10,t0);eq2=subs(X(2)-x20,t0);
[C1, C2] = solve(eq1,eq2);

X=subs(X);
```

INPUT

Enter A: [1 2;0.5 1]

Enter the initial conditions [t0, x10,x20]: [0 16 -2]

OUTPUT

```
X =  
10*exp(t/4503599627370496) + 6*exp(2*t)  
3*exp(2*t) - 5*exp(t/4503599627370496)
```

2. The governing equations of a certain vibrating system are

$$\begin{aligned}x_1'' &= 2x_1 + x_2 \\x_2'' &= 9x_1 + 2x_2\end{aligned}$$

Solve the system of equations by matrix method.

MATLAB CODE

```
clc  
clear  
  
A=input('Enter A: ');  
  
[P D]=eig(A);  
  
Sol1 = dsolve(['D2y = ',num2str(D(1)),'*y']);  
Sol2 = dsolve(['D2y = ',num2str(D(4)),'*y']);  
  
X = P*[Sol1;Sol2];  
  
disp('x1=');disp(X(1))  
disp('x2=');disp(X(2))
```

INPUT

```
Enter A: [-5 2;2 -2]
```

OUTPUT

```
x1=  
(10^(1/2)*(C1*exp(5^(1/2)*t) + C2*exp(-5^(1/2)*t))/10 - (10^(1/2)*  
(C3*cos(t) + C4*sin(t)))/10  
  
x2=  
(3*10^(1/2)*(C3*cos(t) + C4*sin(t))/10 + (3*10^(1/2)*  
(C1*exp(5^(1/2)*t) + C2*exp(-5^(1/2)*t))/10
```

Exercise

3. Solve the following:

- (a) $x'_1 = 3x_1 - 2x_2; x'_2 = 2x_1 - 2x_2; x_1(0) = 1, x_2(0) = -1.$
- (b) $x'_1 = -x_2 + x_3; x'_2 = 4x_1 - x_2 - 4x_3; x'_3 = -3x_1 - x_2 + 4x_3.$

4. Solve the following:

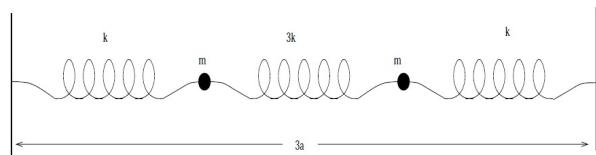
- (a) $x''_1 = -5x_1 + 2x_2; x''_2 = 2x_1 - 2x_2.$
- (b) $x''_1 + 2x_1 - x_2 = 0; x''_2 - x_1 + 2x_2 = 0.$

5. Two particles of equal mass $m = 1$ move in one dimension at the junction of three springs. The springs each have unstretched length $a = 1$ and have spring stiffness constants, k , $3k$ and k (with $k = 1$) respectively see Figure. Applying Newton's second law and Hooke's, this mass-spring system gives rise to the differential equation system

$$x''_1 = -4x_1 + 3x_2$$

$$x''_2 = 3x_1 - 4x_2$$

Find the displacements $x_1(t)$ and $x_2(t)$.



- 6. Reduce the third order equation $y''' + 2y'' - y' - 2y = 0$ to the system of first order linear equations and solve by matrix method.
- 7. Consider tanks T_1 and T_2 which contain initially 100 gallons of water each. In T_1 water is pure whereas 150 pounds of salt is dissolved in T_2 . By circulating the liquid at the rate of 2 gallons per minute and stirring, the amount of salt $y_1(t)$ in T_1 and $y_2(t)$ in T_2 change with time t , find the amount of salt in the two tanks after a time t .

MATLAB code for solving system of first order differential equations by diagonalization process

```
clc
clear all
close all

syms x1(t) x2(t)

A=input('Enter the coefficient matrix A: ');
F=input('Enter the nonhomogenous part:');

[P D]=eig(A)

IP=inv(P);

X=[x1;x2];
FF=IP*F;

sol1=dsolve(diff(x1,1)-D(1)*x1-FF(1)==0);
sol2=dsolve(diff(x2,1)-D(4)*x2-FF(2)==0);

disp('The solution of the given system is : ')
Y=P*[sol1;sol2];
y1=simplify(Y(1))
y2=simplify(Y(2))
```

Example:

Solve the system of first order DE

$$y'_1 + 3y_1 - y_2 = 3t$$

$$y'_2 - 2y_1 + 4y_2 = e^{-t}$$

Solution:

$$Y = c_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} e^{-2t} + c_2 \begin{pmatrix} 1 \\ -2 \end{pmatrix} e^{-5t} + \begin{pmatrix} \frac{6}{5} \\ \frac{3}{5} \end{pmatrix} t - \begin{pmatrix} \frac{27}{50} \\ \frac{21}{50} \end{pmatrix} + \begin{pmatrix} \frac{1}{4} \\ \frac{1}{2} \end{pmatrix} e^{-t}$$

y1 =

$$(6*t)/5 + \exp(-t)/4 + (2^{(1/2)}*C1*\exp(-2*t))/2 - (5^{(1/2)}*C2*\exp(-5*t))/5 - 27/50$$

y2 =

$$(3*t)/5 + \exp(-t)/2 + (2^{(1/2)}*C1*\exp(-2*t))/2 + (2*5^{(1/2)}*C2*\exp(-5*t))/5 - 21/50$$

MAT2002 – Applications of Differential & Difference Equations

**Solution of homogeneous system
of first order and second order
differential equations by Matrix
method**

(Exp – 4A)

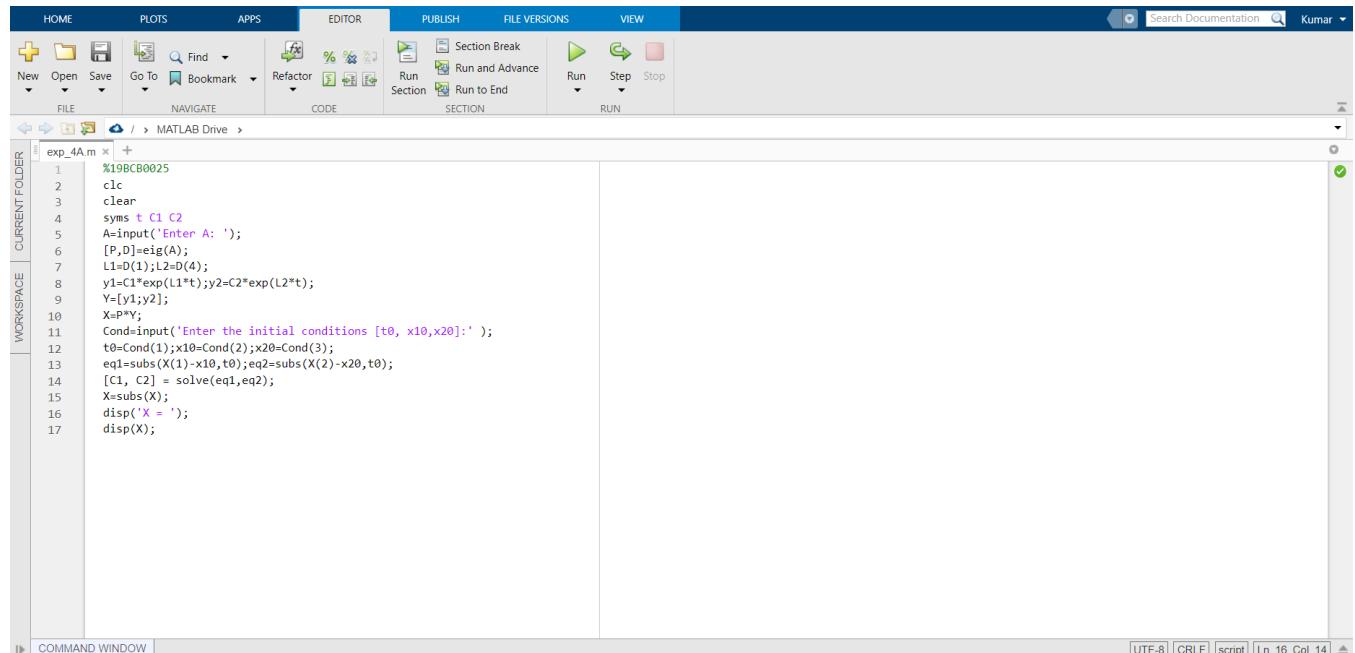
Faculty Name -
Monica C

Presented By -
Kumar Sparsh
19BCB0025

3. Solve the following:

(a) $x'_1 = 3x_1 - 2x_2; x'_2 = 2x_1 - 2x_2; x_1(0) = 1, x_2(0) = -1.$

Sol:

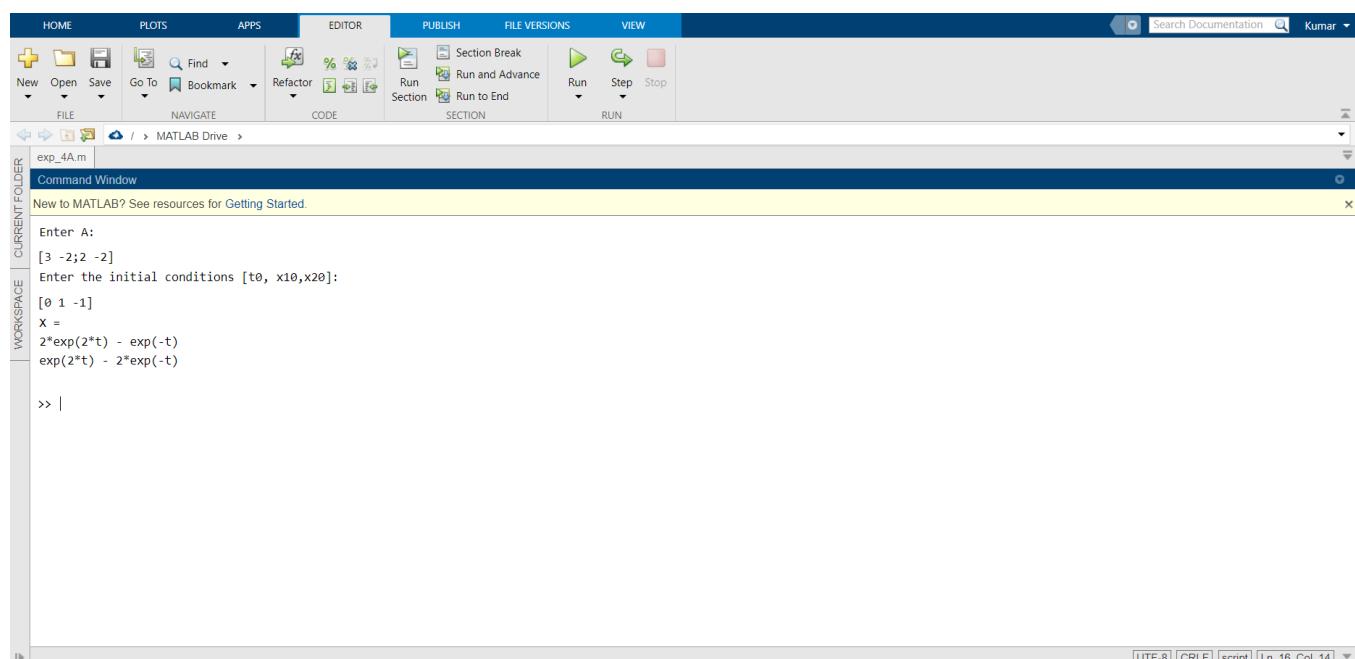


```

exp_4A.m
1 %19BCB0025
2 clc
3 clear
4 syms t C1 C2
5 A=input('Enter A: ');
6 [P,D]=eig(A);
7 L1=D(1);L2=D(4);
8 y1=C1*exp(L1*t);y2=C2*exp(L2*t);
9 Y=[y1;y2];
10 X=P\Y;
11 Cond=input('Enter the initial conditions [t0, x10,x20]: ');
12 t0=Cond(1);x10=Cond(2);x20=Cond(3);
13 eq1=subs(X(1)-x10,t0);eq2=subs(X(2)-x20,t0);
14 [C1, C2] = solve(eq1,eq2);
15 X=subs(X);
16 disp('X = ');
17 disp(X);

```

Figure 1 : MATLAB Code



```

Command Window
New to MATLAB? See resources for Getting Started.

Enter A:
[3 -2;2 -2]
Enter the initial conditions [t0, x10,x20]:
[0 1 -1]
X =
2*exp(2*t) - exp(-t)
exp(2*t) - 2*exp(-t)

>> |

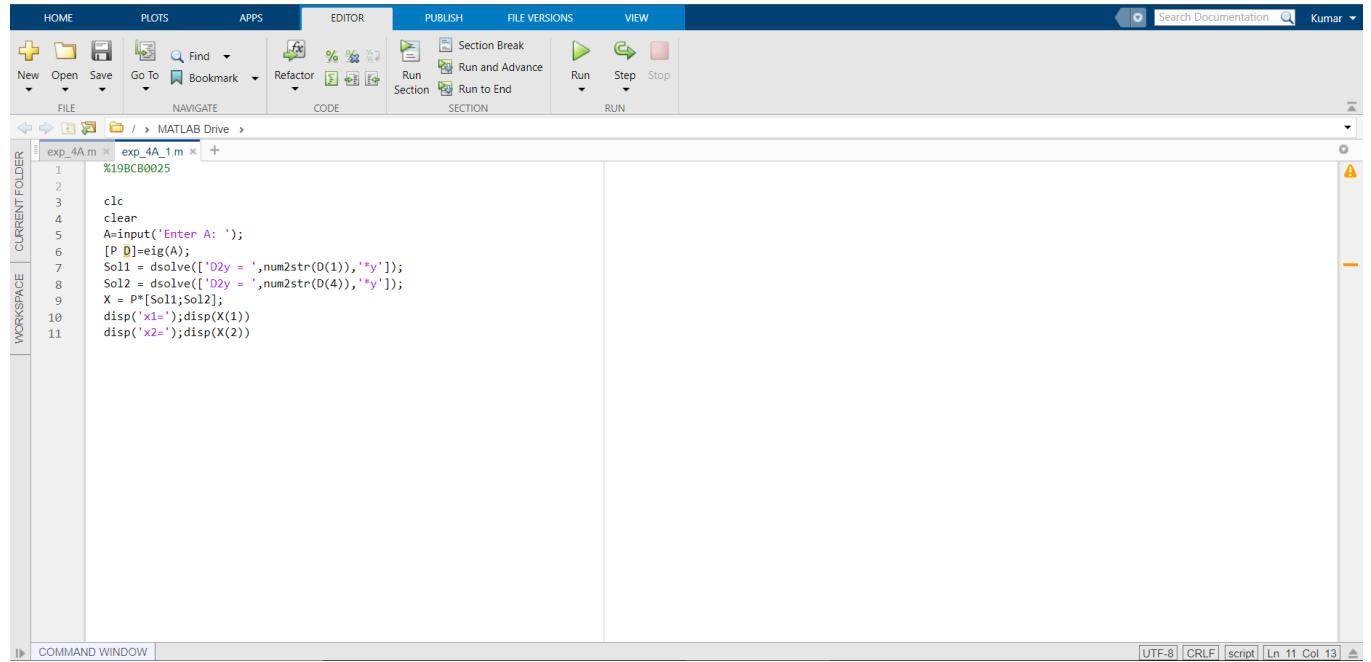
```

Figure 2 : Command Window Output

4. Solve the following:

(a) $x_1'' = -5x_1 + 2x_2; x_2'' = 2x_1 - 2x_2.$

Sol:



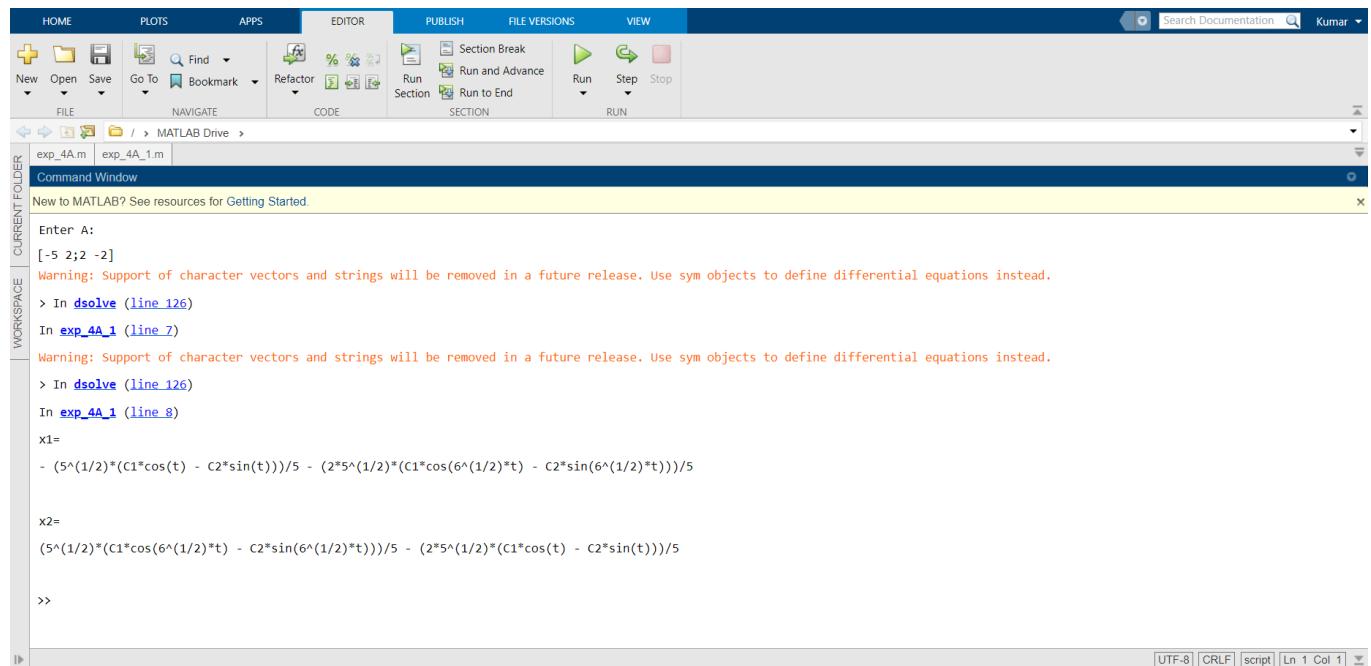
The screenshot shows the MATLAB interface with the Editor tab selected. The code in the editor window is as follows:

```

1 %> exp_4A.m < exp_4A_1.m +
2 %> %> 19BCB0025
3 clc
4 clear
5 A=input('Enter A: ');
6 [P D]=eig(A);
7 Sol1 = dsolve(['D2y = ',num2str(D(1)),'*y']);
8 Sol2 = dsolve(['D2y = ',num2str(D(4)),'*y']);
9 X = P*[Sol1;Sol2];
10 disp('x1=');disp(X(1))
11 disp('x2=');disp(X(2))

```

Figure 3 : MATLAB Code for the following examples



The screenshot shows the MATLAB interface with the Command Window tab selected. The command window displays the following output:

```

>> exp_4A.m
>> exp_4A_1.m
>> Command Window
New to MATLAB? See resources for Getting Started.

Enter A:
[-5 2;2 -2]
Warning: Support of character vectors and strings will be removed in a future release. Use sym objects to define differential equations instead.
> In dsolve (line 126)
In exp_4A_1 (line 7)
Warning: Support of character vectors and strings will be removed in a future release. Use sym objects to define differential equations instead.
> In dsolve (line 126)
In exp_4A_1 (line 8)
X1=
- (5^(1/2)*(C1*cos(t) - C2*sin(t)))/5 - (2*5^(1/2)*(C1*cos(6^(1/2)*t) - C2*sin(6^(1/2)*t)))/5

X2=
(5^(1/2)*(C1*cos(6^(1/2)*t) - C2*sin(6^(1/2)*t)))/5 - (2*5^(1/2)*(C1*cos(t) - C2*sin(t)))/5

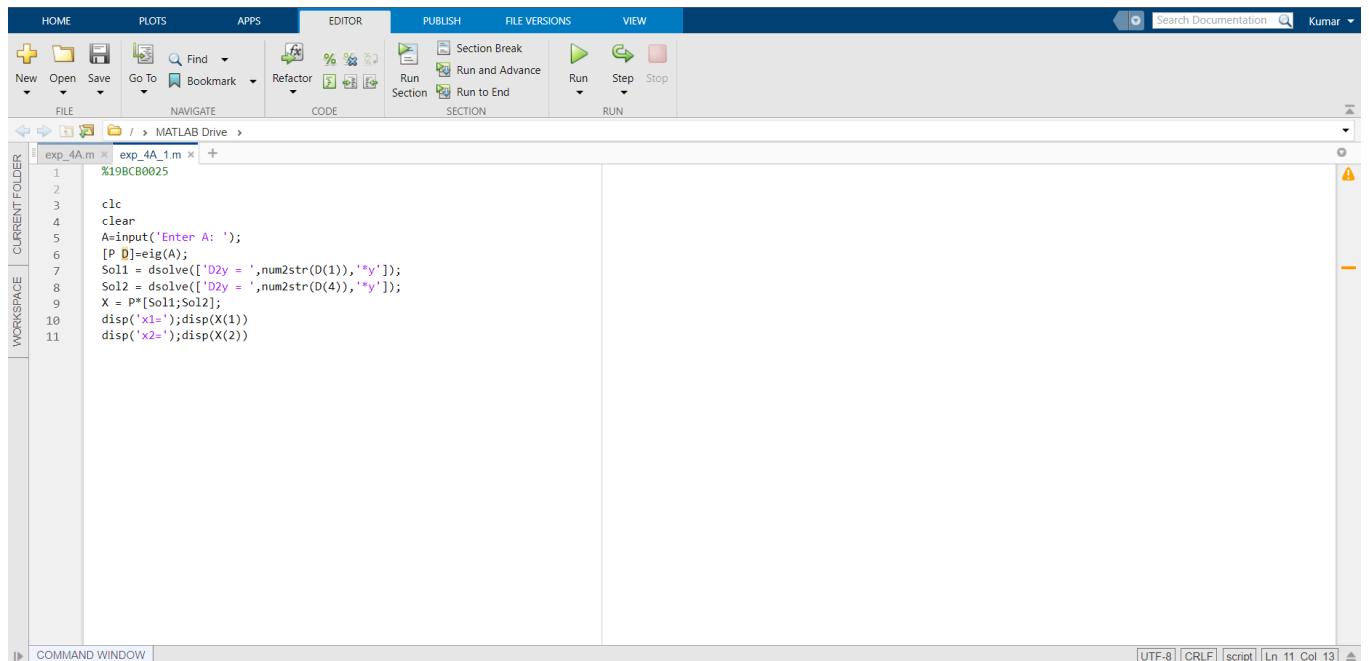
>>

```

Figure 3 : Command Window Output

$$(b) \quad x_1'' + 2x_1 - x_2 = 0; \quad x_2'' - x_1 + 2x_2 = 0.$$

Sol:



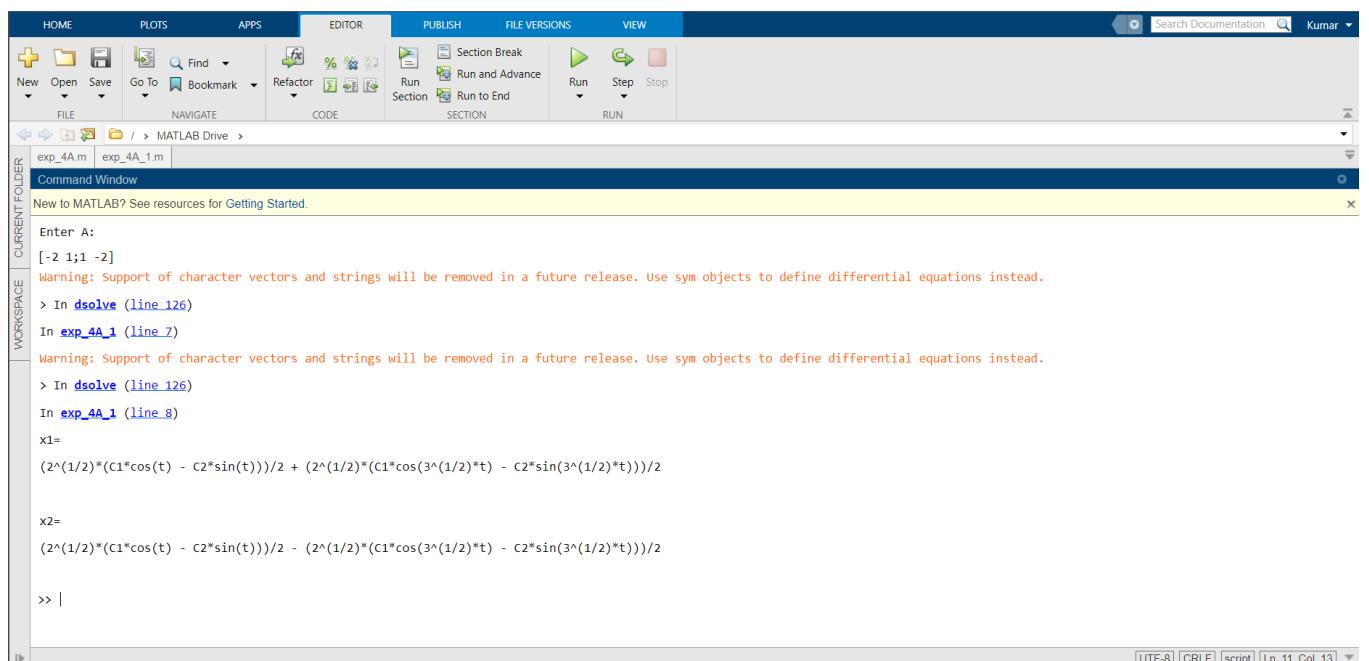
The screenshot shows the MATLAB Editor interface. The current file is `exp_4A.m`. The code is as follows:

```

1 %19BCB0025
2
3 clc
4 clear
5 A=input('Enter A: ');
6 [P D]=eig(A);
7 Sol1 = dsolve(['D2y = ',num2str(D(1)),'*y']);
8 Sol2 = dsolve(['D2y = ',num2str(D(4)),'*y']);
9 X = P*[Sol1;Sol2];
10 disp('x1=');disp(X(1))
11 disp('x2=');disp(X(2))

```

Figure 5 : MATLAB Code for the following examples



The screenshot shows the MATLAB Command Window. The user has entered the matrix `A` as `[-2 1;1 -2]`. The command `dsolve` was used to solve the system of differential equations. The output shows the general solution for x_1 and x_2 in terms of constants $C1$ and $C2$.

```

Enter A:
[-2 1;1 -2]
Warning: Support of character vectors and strings will be removed in a future release. Use sym objects to define differential equations instead.
> In dsolve (line 126)
In exp_4A_1 (line 7)
Warning: Support of character vectors and strings will be removed in a future release. Use sym objects to define differential equations instead.
> In dsolve (line 126)
In exp_4A_1 (line 8)
x1=
(2^(1/2)*(C1*cos(t) - C2*sin(t)))/2 + (2^(1/2)*(C1*cos(3^(1/2)*t) - C2*sin(3^(1/2)*t)))/2

x2=
(2^(1/2)*(C1*cos(t) - C2*sin(t)))/2 - (2^(1/2)*(C1*cos(3^(1/2)*t) - C2*sin(3^(1/2)*t)))/2

>> |

```

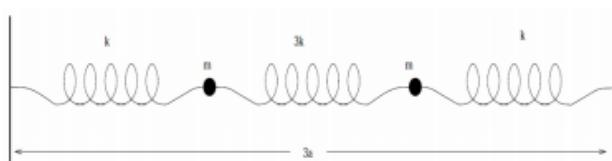
Figure 4 : Command Window Output

5. Two particles of equal mass $m = 1$ move in one dimension at the junction of three springs. The springs each have unstretched length $a = 1$ and have spring stiffness constants, k , $3k$ and k (with $k = 1$) respectively see Figure. Applying Newton's second law and Hooke's, this mass-spring system gives rise to the differential equation system

$$x_1'' = -4x_1 + 3x_2$$

$$x_2'' = 3x_1 - 4x_2$$

Find the displacements $x_1(t)$ and $x_2(t)$.



Sol:

```

HOME PLOTS APPS EDITOR PUBLISH FILE VERSIONS VIEW
New Open Save Go To Find Refactor Run Section Run and Advance Run Step Stop
FILE NAVIGATE CODE SECTION RUN
exp_4A.m exp_4A_1.m
Command Window
New to MATLAB? See resources for Getting Started.
Enter A:
[-4 3; 3 -4]
Warning: Support of character vectors and strings will be removed in a future release. Use sym objects to define differential equations instead.
> In dsolve (line 126)
In exp_4A_1 (line 7)
Warning: Support of character vectors and strings will be removed in a future release. Use sym objects to define differential equations instead.
> In dsolve (line 126)
In exp_4A_1 (line 8)
x1=
(2^(1/2)*(C1*cos(t) - C2*sin(t)))/2 + (2^(1/2)*(C1*cos(7^(1/2)*t) - C2*sin(7^(1/2)*t)))/2
x2=
(2^(1/2)*(C1*cos(t) - C2*sin(t)))/2 - (2^(1/2)*(C1*cos(7^(1/2)*t) - C2*sin(7^(1/2)*t)))/2
>>
UTF-8 | CRLF | script | Ln 11 Col 13

```

Figure 7 : Command Window Output



DEPARTMENT OF MATHEMATICS
SCHOOL OF ADVANCED SCIENCES
Series Solutions of Ordinary Differential Equations.

Course Code: MAT2002

Course Name: Application of Differential and Difference Equations

Experiment: 4-B

Duration: 90 Minutes

Series Solution when $x = 0$ ia an Ordinary Point of the Equation

$$P_0 \frac{d^2y}{dx^2} + P_1 \frac{dy}{dx} + P_2 y = 0 \quad (1)$$

where P 's are polynomial in x and $P_0 \neq 0$ at $x = 0$.

1. Assume its solution to be of the form

$$y = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n + \cdots \quad (2)$$

2. Calculate $\frac{dy}{dx}$, $\frac{d^2y}{dx^2}$, from (2) and substitute the values of y , $\frac{dy}{dx}$, $\frac{d^2y}{dx^2}$ in (1).
3. Equate to zero the coefficients of the various powers of x and determine a_2, a_3, a_4, \dots in terms of a_0, a_1 .
4. Substituting the values of a_2, a_3, a_4, \dots in (2), we get the desired series solution having a_0, a_1 as its arbitrary constants.
1. Solve in series the equation $\frac{d^2y}{dx^2} + y = 0$.

MATLAB CODE

```
clc  
clear  
  
syms x a0 a1 a2 a3  
a = [a0 a1 a2 a3];  
y = sum(a.*(x.^ [0:3]));  
  
dy = diff(y);  
d2y = diff(dy);  
gde = collect(d2y+y,x);  
cof=coeffs(gde,x);  
  
A2=solve(cof(1),a2);  
A3=solve(cof(2),a3);  
  
y=subs(y,a2,a3,A2,A3);  
y=coeffs(y,[a1 a0]);  
disp('Solution is')  
disp(['y=A(',char(y(1)),'+ ...)+B(',char(y(2)),'+ ...)'])
```

OUTPUT

```
Solution is  
y=A(1 - x^2/2+ ...)+B(x - x^3/6+ ...)
```

Exercise

2. Solve the following:

- (a) $\frac{d^2y}{dx^2} + xy = 0$
- (b) $\frac{d^2y}{dx^2} + x^2y = 0$
- (c) $y'' + xy' + y = 0.$
- (d) $(1 - x^2)y'' + 2y = 0; y(0) = 4, y'(0) = 5$

3. The half-life of radium is 1600 years, i.e., it takes 1600 years for half of any quantity to decay. If a sample initially contains 50 g, how long will it be until it contains 45 g by power series method?

MAT2002 – Applications of Differential & Difference Equations

Series Solutions of Ordinary Differential Equations

(Exp – 4B)

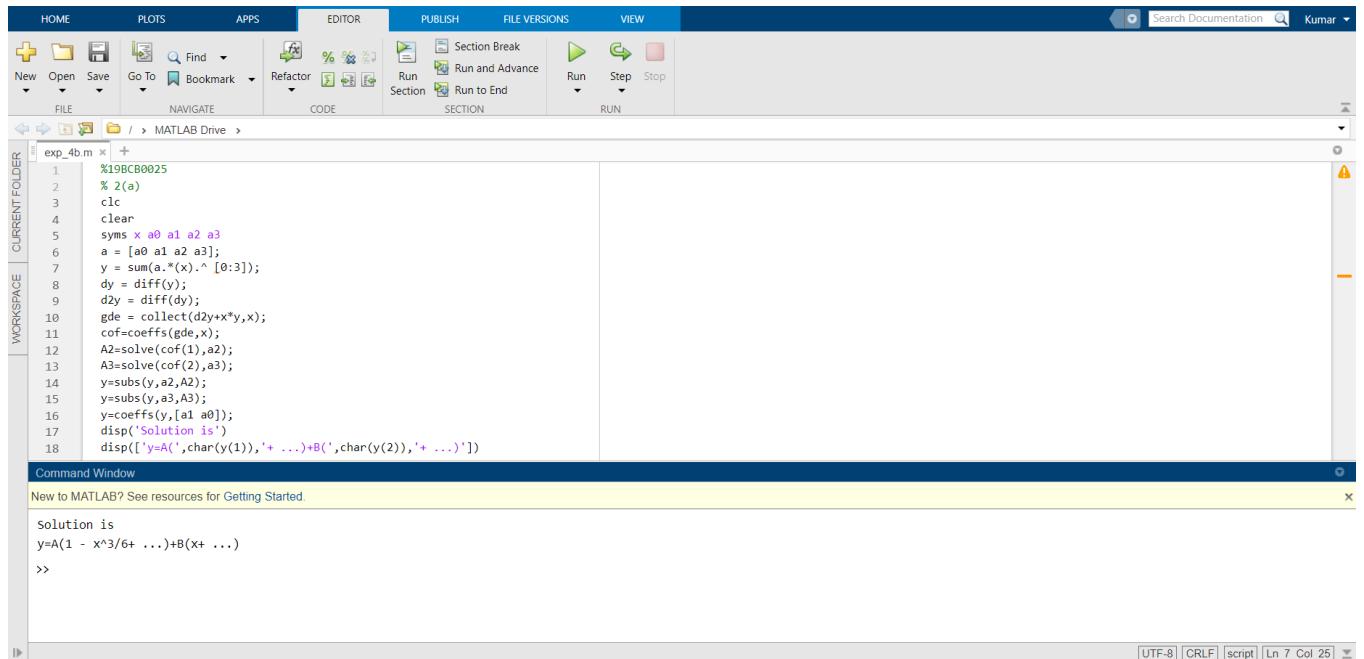
Faculty Name -
Monica C

Presented By -
Kumar Sparsh
19BCB0025

2. Solve the following:

$$(a) \frac{d^2y}{dx^2} + xy = 0$$

Sol:



The screenshot shows the MATLAB desktop interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. The EDITOR tab is active, showing the script file 'exp_4b.m'. The code in the editor is:

```

exp_4b.m
1 %19BCB0025
2 % 2(a)
3 clc
4 clear
5 syms x a0 a1 a2 a3
6 a = [a0 a1 a2 a3];
7 y = sum(a.*x.^ [0:3]);
8 dy = diff(y);
9 d2y = diff(dy);
10 gde = collect(d2y+x*y,x);
11 cof=coeffs(gde,x);
12 A2=solve(cof(1),a2);
13 A3=solve(cof(2),a3);
14 y=subs(y,a2,A2);
15 y=subs(y,a3,A3);
16 y=coeffs(y,[a1 a0]);
17 disp('Solution is')
18 disp(['y=A(' ,char(y(1)), '+ ...)+B(' ,char(y(2)), '+ ...)' ])

```

The Command Window below shows the output:

```

solution is
y=A(1 - x^3/6+ ...)+B(x+ ...)
>>

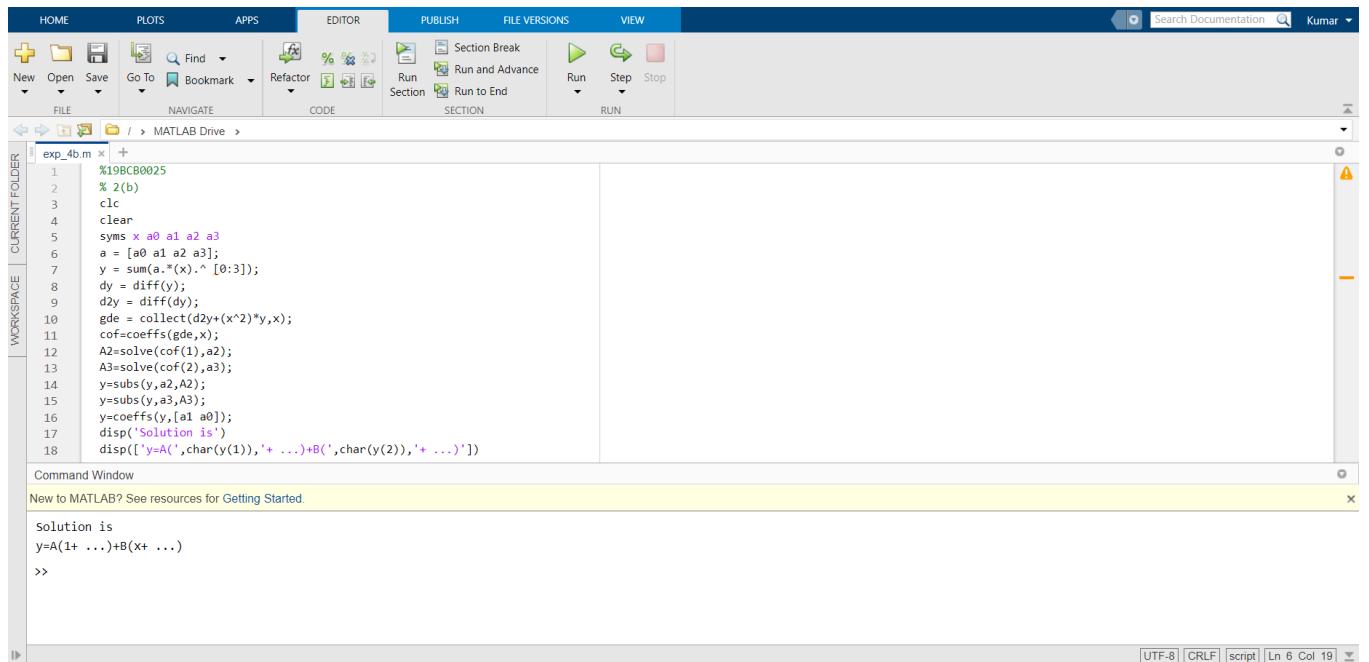
```

At the bottom right of the Command Window, there are status indicators: UTF-8, CRLF, script, Ln 7 Col 25.

Figure 1 : MATLAB Code & Command Window Output

$$(b) \frac{d^2y}{dx^2} + x^2y = 0$$

Sol:



The screenshot shows the MATLAB R2022a interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. The EDITOR tab is active, showing the code for exp_4b.m. The code is as follows:

```

exp_4b.m
1 %19BCB0025
2 % 2(b)
3 clc
4 clear
5 syms x a0 a1 a2 a3
6 a = [a0 a1 a2 a3];
7 y = sum(a,*(x).^(0:3));
8 dy = diff(y);
9 d2y = diff(dy);
10 gde = collect(d2y+(x^2)*y,x);
11 cof=coeffs(gde,x);
12 A2=solve(cof(1),a2);
13 A3=solve(cof(2),a3);
14 y=subs(y,a2,A2);
15 y=subs(y,a3,A3);
16 y=coeffs(y,[a1 a0]);
17 disp('Solution is')
18 disp(['y=A(' ,char(y(1)), '+ ...)+B(' ,char(y(2)), '+ ...)' ])

```

The Command Window below shows the output of the script:

```

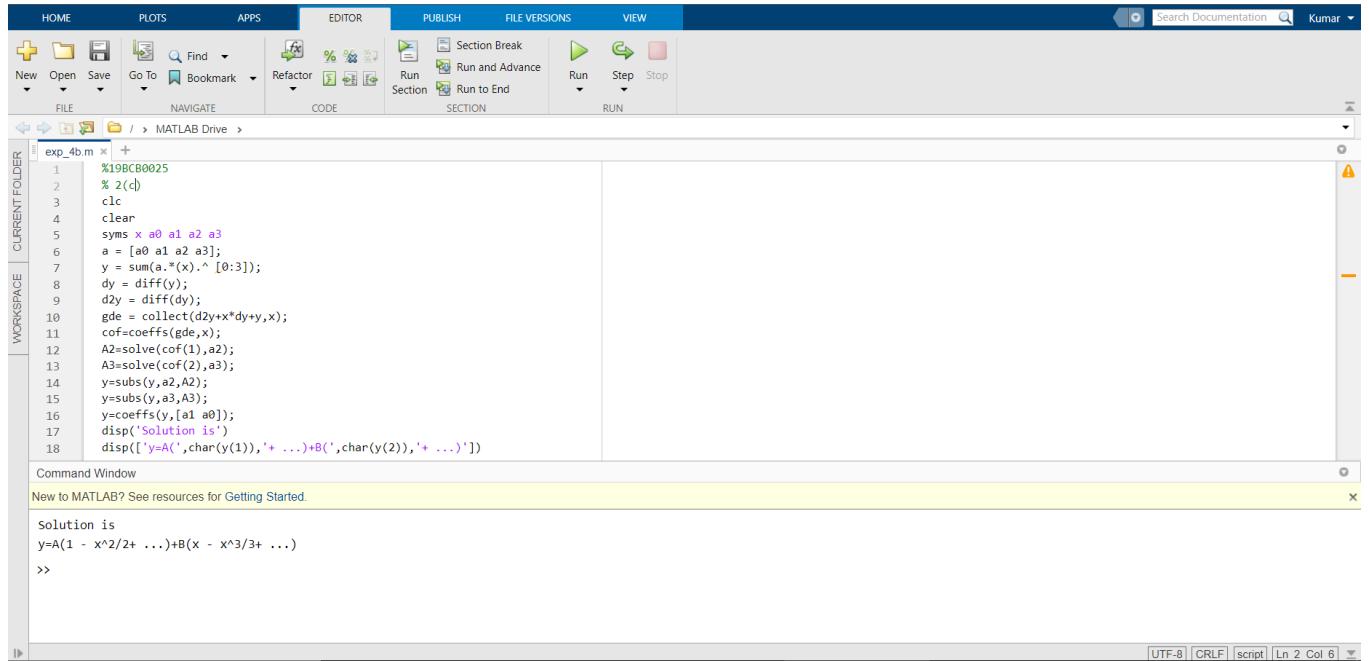
solution is
y=A(1+ ...) +B(x+ ...)
>>

```

Figure 2 : MATLAB Code & Command Window Output

$$(c) \quad y'' + xy' + y = 0.$$

Sol:



The screenshot shows the MATLAB interface with the following details:

- Editor Tab:** Contains the MATLAB script `exp_4b.m` which defines coefficients a_0, a_1, a_2, a_3 , calculates y and its derivatives, and solves the differential equation to find the general solution $y = A(1 - x^2/2 + \dots) + B(x - x^3/3 + \dots)$.
- Command Window:** Displays the command `>>` and the resulting output: `Solution is` followed by the general solution formula.
- Status Bar:** Shows file encoding as `UTF-8`, line endings as `CRLF`, and current line and column as `Ln 2 Col 6`.

```

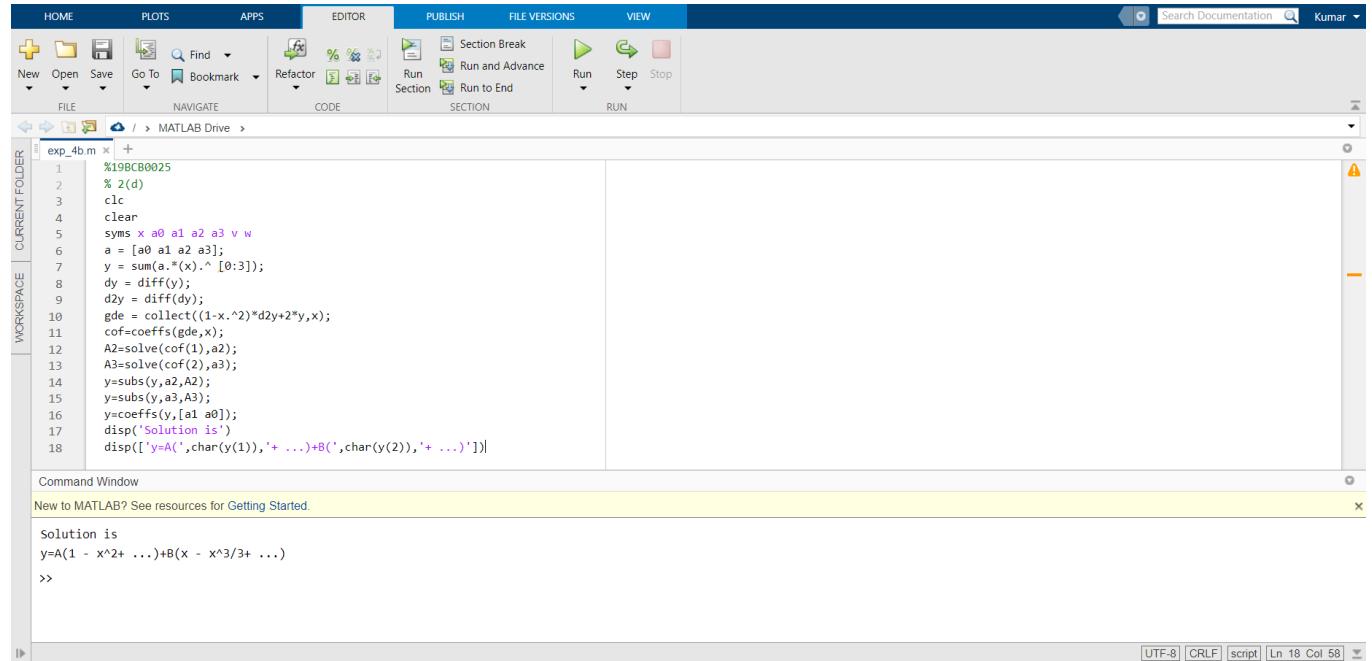
exp_4b.m
1 %19BCB0025
2 % 2(c)
3 clc
4 clear
5 syms x a0 a1 a2 a3
6 a = [a0 a1 a2 a3];
7 y = sum(a.*x.^ [0:3]);
8 dy = diff(y);
9 d2y = diff(dy);
10 gde = collect(d2y+x*dy+y,x);
11 cof=coeffs(gde,x);
12 A2=solve(cof(1),a2);
13 A3=solve(cof(2),a3);
14 y=subs(y,a2,A2);
15 y=subs(y,a3,A3);
16 y=coeffs(y,[a1 a0]);
17 disp('Solution is')
18 disp(['y=A(' ,char(y(1)), '+ ...)+B(' ,char(y(2)), '+ ...)' ])

```

Figure 3 : MATLAB Code & Command Window Output

$$(d) (1 - x^2)y'' + 2y = 0;$$

Sol:



The screenshot shows the MATLAB desktop interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. The EDITOR tab is active, showing the code for 'exp_4b.m'. The WORKSPACE browser on the left shows variables like 'a0', 'a1', 'a2', 'a3', 'v', and 'w'. The Command Window at the bottom displays the executed code and its output.

```
%19BCB0025
% 2(d)
clc
clear
syms x a0 a1 a2 a3 v w
a = [a0 a1 a2 a3];
y = sum(a.*x).^ [0:3]);
dy = diff(y);
d2y = diff(dy);
gde = collect((1-x.^2)*d2y+2*y,x);
cof=coeffs(gde,x);
A2=solve(cof(1),a2);
A3=solve(cof(2),a3);
y=subs(y,a2,A2);
y=subs(y,a3,A3);
y=coeffs(y,[a1 a0]);
disp('Solution is')
disp(['y=A('',char(y(1)),'+ ...)+B('',char(y(2)),'+ ...)''])
```

New to MATLAB? See resources for [Getting Started](#).

solution is

y=A(1 - x^2+ ...) +B(x - x^3/3+ ...)

>>

Figure 4 : MATLAB Code & Command Window Output

Department of Mathematics
School of Advanced Sciences

MAT 2002 – Applications of Differential and Difference Equations (MATLAB)

Experiment 5–A

Z-transforms and their applications for solving Difference equations

Z-Transform

If the function u_n is defined for discrete values ($n = 0, 1, 2, \dots$) and $u_n = 0$ for $n < 0$, then its Z-transform is defined as

$$Z\{u_n\} = \bar{U}(z) = \sum_{n=0}^{\infty} \frac{u_n}{z^n}$$

whenever the infinite series converges.

The inverse Z-transform is written as $Z^{-1}\{\bar{U}(z)\} = u_n$.

MATLAB Syntax used:

<code>ztrans(f)</code>	The z -transform of the scalar symbol f with default independent variable n . The default return is a function of z .
<code>ztrans(f, w)</code>	Makes f a function of the symbol w instead of the default z .
<code>ztrans(f, k, w)</code>	Takes f to be a function of the symbolic variable k .
<code>iztrans(F)</code>	The inverse z -transform of the scalar symbolic object F with default independent variable z . The default return is a function of n .
<code>iztrans(F, k)</code>	Makes f a function of k instead of the default n . Here k is a scalar symbolic object.
<code>iztrans(F, w, k)</code>	Takes F to be a function of w instead of the default variable z and returns a function of k .
<code>collect(P, var)</code>	Rewrites P in terms of the powers of the variable var .
<code>stem(Y)</code>	Plots the data sequence Y as stems that extend from equally spaced and automatically generated values along the x -axis. When Y is a matrix, stem plots all elements in a row against the same x value.

Example 1. Find the z-transform of the function $y_n = \frac{1}{4^n}$, $n \geq 0$.

```
>>syms z n;
>>ztrans(1/4^n)

Output: ans =
z / (z - 1/2)
```

Example 2. Find the inverse z-transform of the following $y(z) = \frac{2z}{2z-1}$

```
>>syms z n;
>>iztrans(2*z/(2*z-1))
```

Output: ans =
 $(1/2)^n$

Solution of linear difference equations with constant coefficients by Z-transforms.

Consider the Linear difference equation

$$ay_{n+2} + by_{n+1} + cy_n = f(n) \quad (1)$$

subject to the initial conditions

$$y_0 = \alpha, \quad y_1 = \beta. \quad (2)$$

The working procedure:

1. Input the difference equation coefficients and the right hand side function of (1).
2. Input the initial conditions (2).
3. Apply Z-Transform and find $Y(z)$.
4. Apply inverse Z – Transform and find y_n .

MATLAB CODE

```
clear all
clc
syms n z y(n) Y
yn=y(n);
yn1=y(n+1);
yn2=y(n+2);
F = input('Input the coefficients [a,b,c]: ');
a=F(1);b=F(2);c=F(3);
nh = input('Enter the non-homogenous part f(n): ');
eqn=a*yn2+b*yn1+c*yn-nh;
ZTY=ztrans(eqn);
IC=input('Enter the initial conditions in the form [y0,y1]: ');
y0=IC(1);y1=IC(2);
ZTY=subs(ZTY,{ 'ztrans(y(n),n,z)', 'y(0)', 'y(1)' },{Y,y0,y1});
eq=collect(ZTY,Y);
Y=simplify(solve(eq,Y));
yn=simplify(iztrans(Y));
disp('The solution of the difference equation yn=')
disp(yn);
m=0:20;
y=subs(yn,n,m);
stem(y)
title('Difference equation');
xlabel('n'); ylabel('y(n)');
```

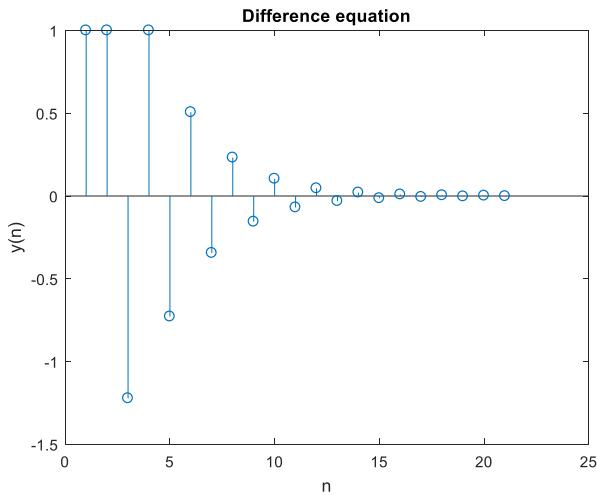
Example 3. Solve $9y_{n+2} + 9y_{n+1} + 2y_n = 0$, $n \geq 0$, with $y_0 = 1$ and $y_1 = 1$.

Input:

```
Input the coefficients [a,b,c]: [9 9 2]
Enter the non-homogenous part f(n): 0
Enter the initial conditions in the form [y0,y1]:[1 1]
```

Output:

```
The solution of the difference equation yn=
5*(-1/3)^n - 4*(-2/3)^n
```



Example 4: Solve the equation $y_{n+2} - 3y_{n+1} + 2y_n = 3^n$, $y_0 = 0$, $y_1 = 1$.

Input:

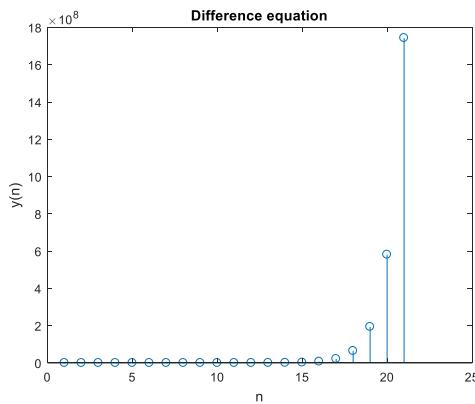
Input the coefficients [a,b,c]: [1 -3 2]

Enter the non-homogenous part f(n): 3^n

Enter the initial conditions in the form [y0,y1]:[0,1]

Output:

The solution of the difference equation $y_n = 3^n/2 - 1/2$



Exercise

1. Solve $y_{n+2} - 5y_{n+1} + 6y_n = 5^n$, $n \geq 0$, $y_0 = 1$ and $y_1 = 1$.
2. Solve $y(n+2) - y(n) = 2^n$, $n \geq 0$, $y_0 = 0$ and $y_1 = 1$.
3. Solve $y(n+2) + 2y(n+1) + y(n) = n$, $n \geq 0$, $y_0 = 0$ and $y_1 = 0$.
4. Solve $y(n+2) - 4y(n+1) + 3y(n) = n \cdot 2^n$, $n \geq 0$, $y_0 = 0$ and $y_1 = 0$.
5. Formulate the difference equation for Fibonacci numbers and hence solve by Z-transforms.

MAT2002 – Applications of Differential & Difference Equations

Z-transforms and their applications for solving Difference equations

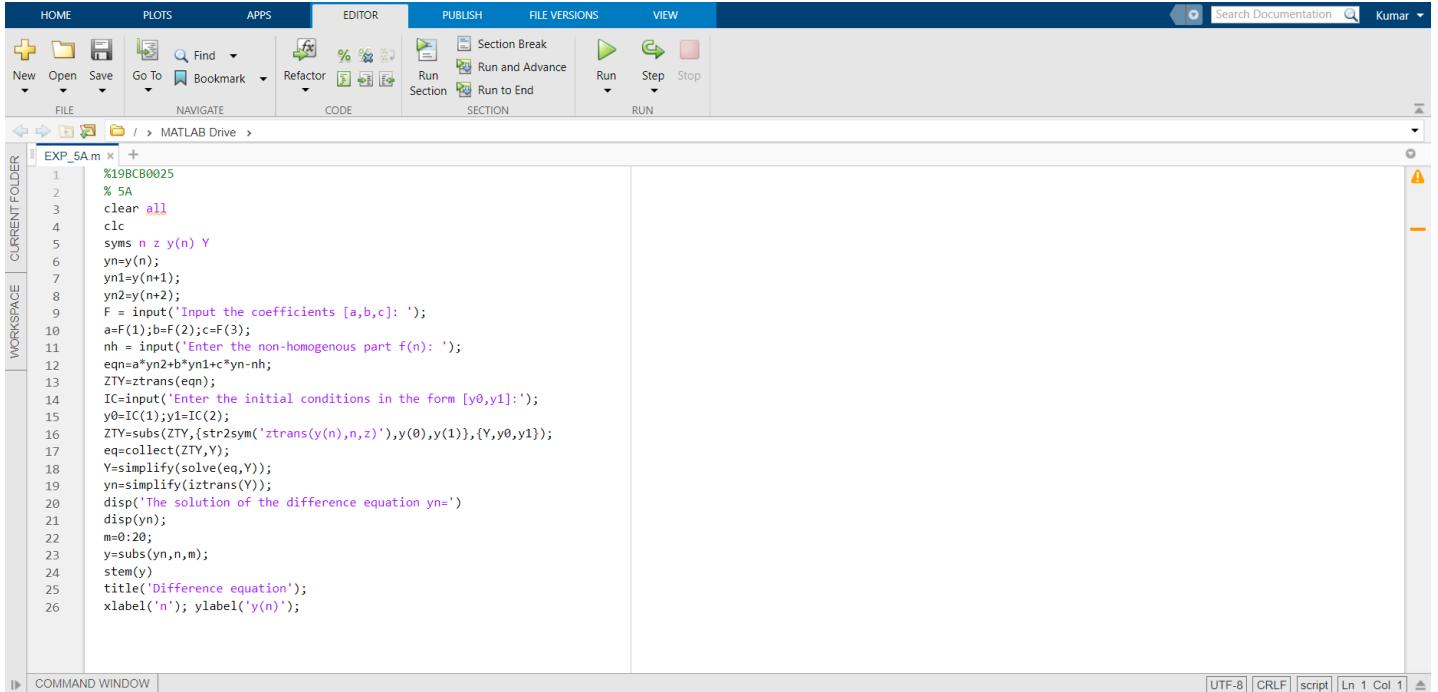
(Exp – 5A)

Faculty Name -
Monica C

Presented By -
Kumar Sparsh
19BCB0025

1. Solve $y_{n+2} - 5y_{n+1} + 6y_n = 5^n, n \geq 0$, $y_0 = 1$ and $y_1 = 1$.

Sol:



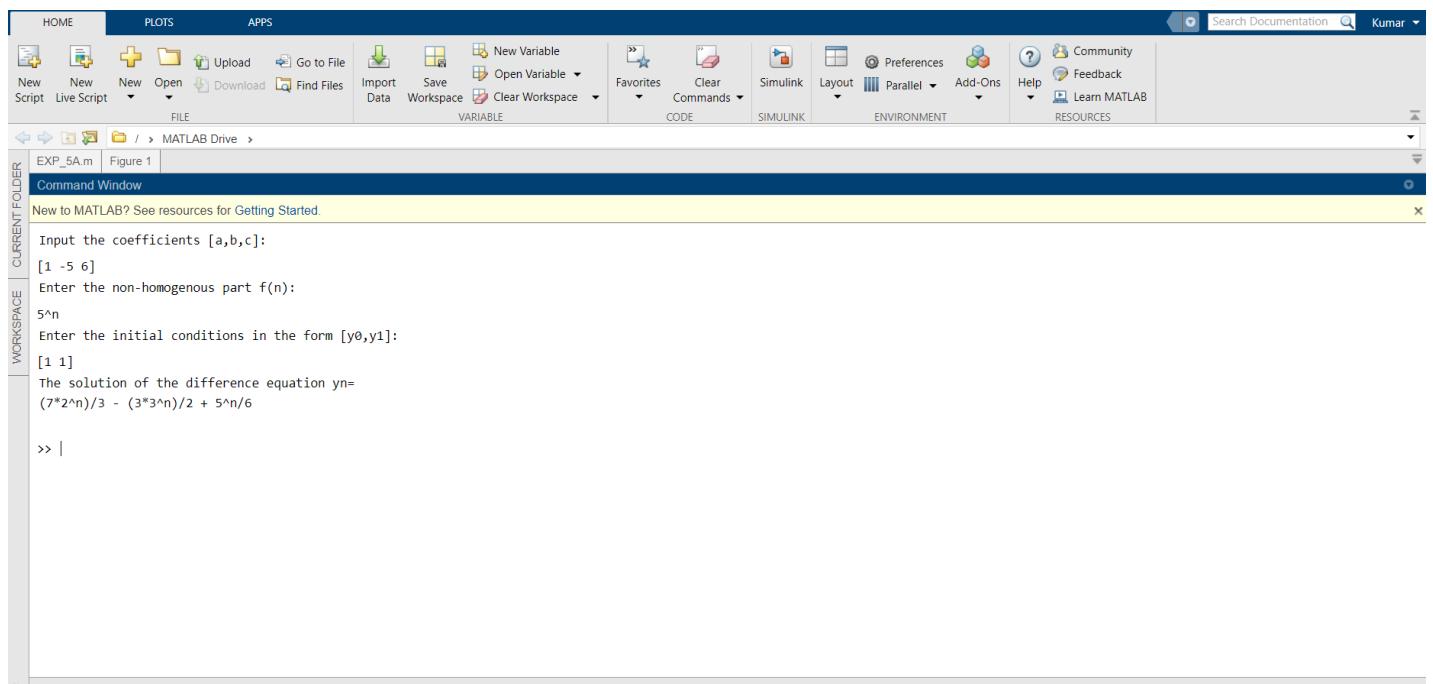
```

HOME PLOTS APPS EDITOR PUBLISH FILE VERSIONS VIEW
New Open Save Go To Find Refactor % Run Section Break
FILE NAVIGATE CODE Run and Advance Run Step Stop
SECTION RUN

EXP_5A.m x +
1 %19BCB0025
2 % 5A
3 clear all
4 clc
5 syms n z y(n) Y
6 yn=y(n);
7 yn1=y(n+1);
8 yn2=y(n+2);
9 F = input('Input the coefficients [a,b,c]: ');
10 a=F(1);b=F(2);c=F(3);
11 nh = input('Enter the non-homogenous part f(n): ');
12 eqn=a*yn2+b*yn1+c*yn-nh;
13 ZTY=ztrans(eqn);
14 IC=input('Enter the initial conditions in the form [y0,y1]: ');
15 y0=IC(1);y1=IC(2);
16 ZTY=subs(ZTY,{'str2sym('ztrans(y(n),n,z)'),y(0),y(1)},[Y,y0,y1]);
17 eq=collect(ZTY,Y);
18 Y=simplify(solve(eq,Y));
19 yn=simplify(iztrans(Y));
20 disp('The solution of the difference equation yn=')
21 disp(yn);
22 m=0:20;
23 y=subs(yn,n,m);
24 stem(y)
25 title('Difference equation');
26 xlabel('n'); ylabel('y(n)');

```

Figure 1 : MATLAB Code for all following questions



```

HOME PLOTS APPS
New New New Open Upload Go to File Import Save Open Variable Favorites Clear Simulink Layout Preferences Parallel Add-Ons Help Feedback Community
New Script Live Script FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES
Search Documentation Kumar

EXP_5A.m Figure 1
Command Window
New to MATLAB? See resources for Getting Started.

Input the coefficients [a,b,c]:
[1 -5 6]
Enter the non-homogenous part f(n):
5^n
Enter the initial conditions in the form [y0,y1]:
[1 1]
The solution of the difference equation yn=
(7*2^n)/3 - (3*3^n)/2 + 5^n/6

>> |

```

Figure 2 : Command Window Output

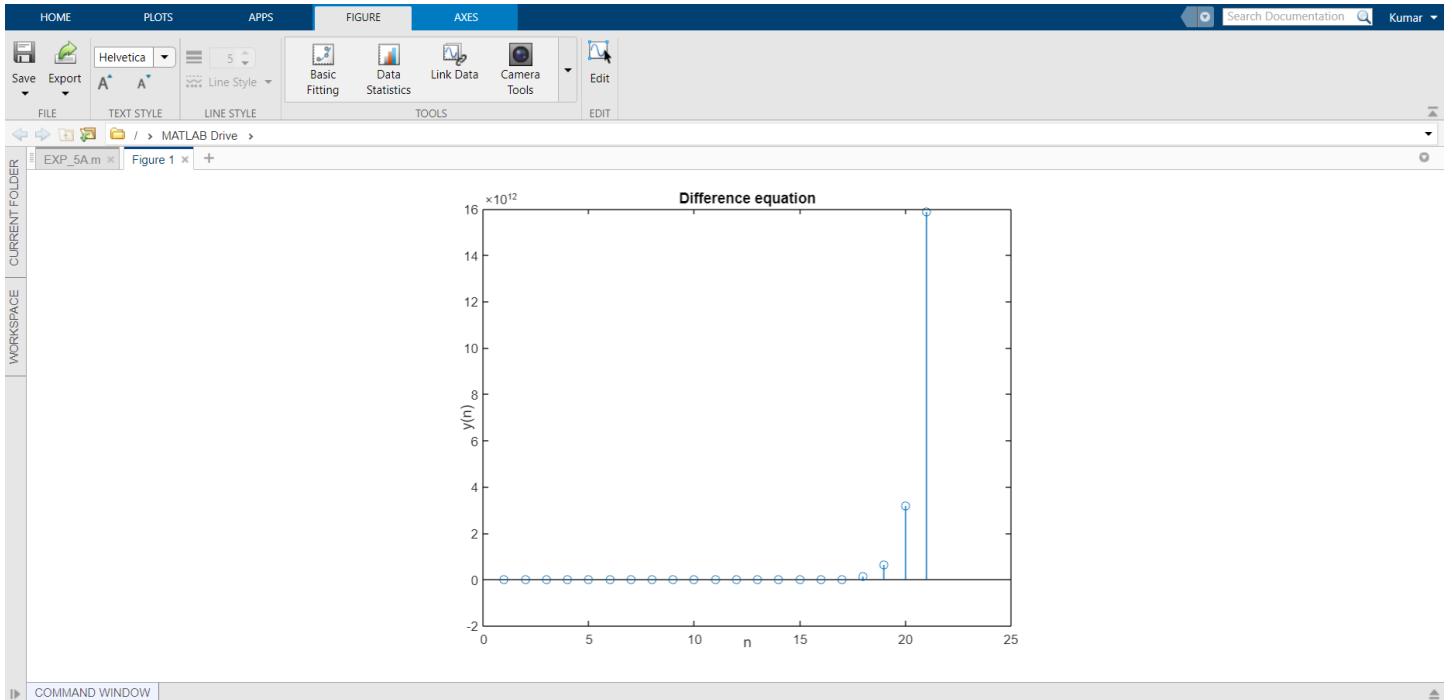


Figure 3 : Graph Output

2. Solve $y(n+2) - y(n) = 2^n, n \geq 0, y_0 = 0$ and $y_1 = 1$.

Sol:

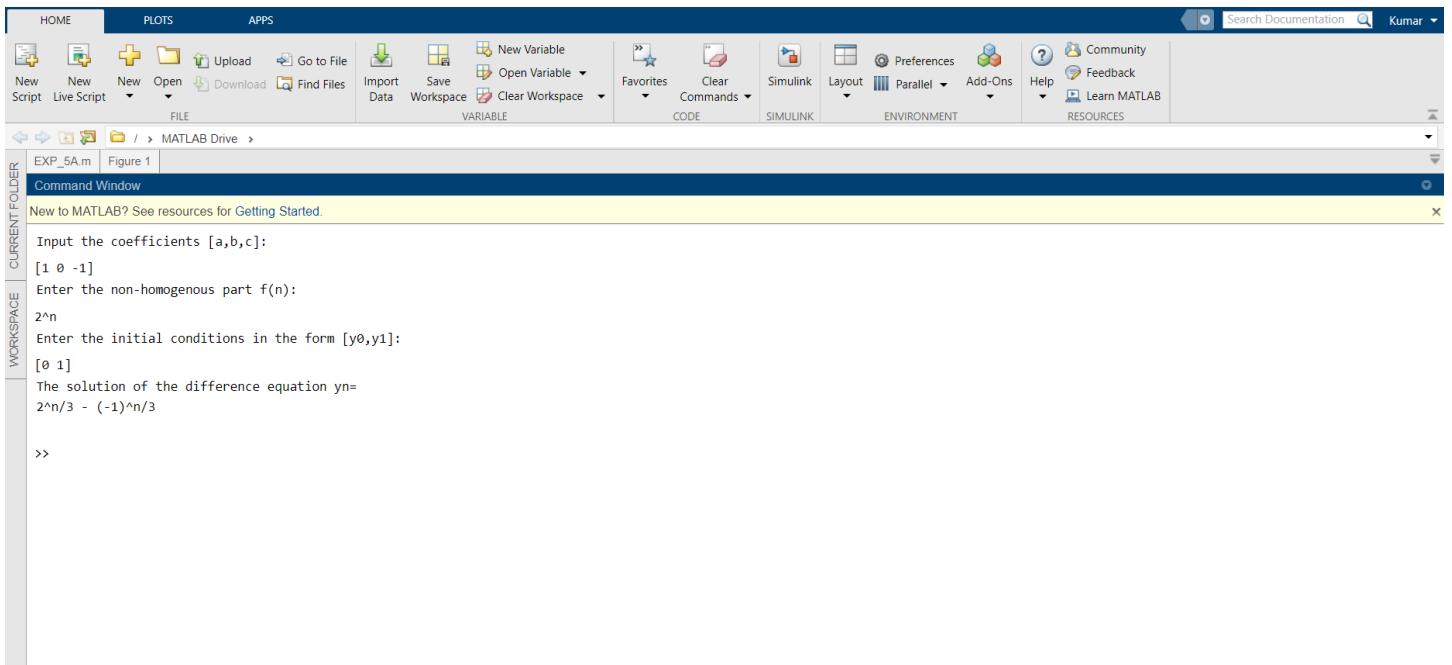


Figure 1 : Command Window Output

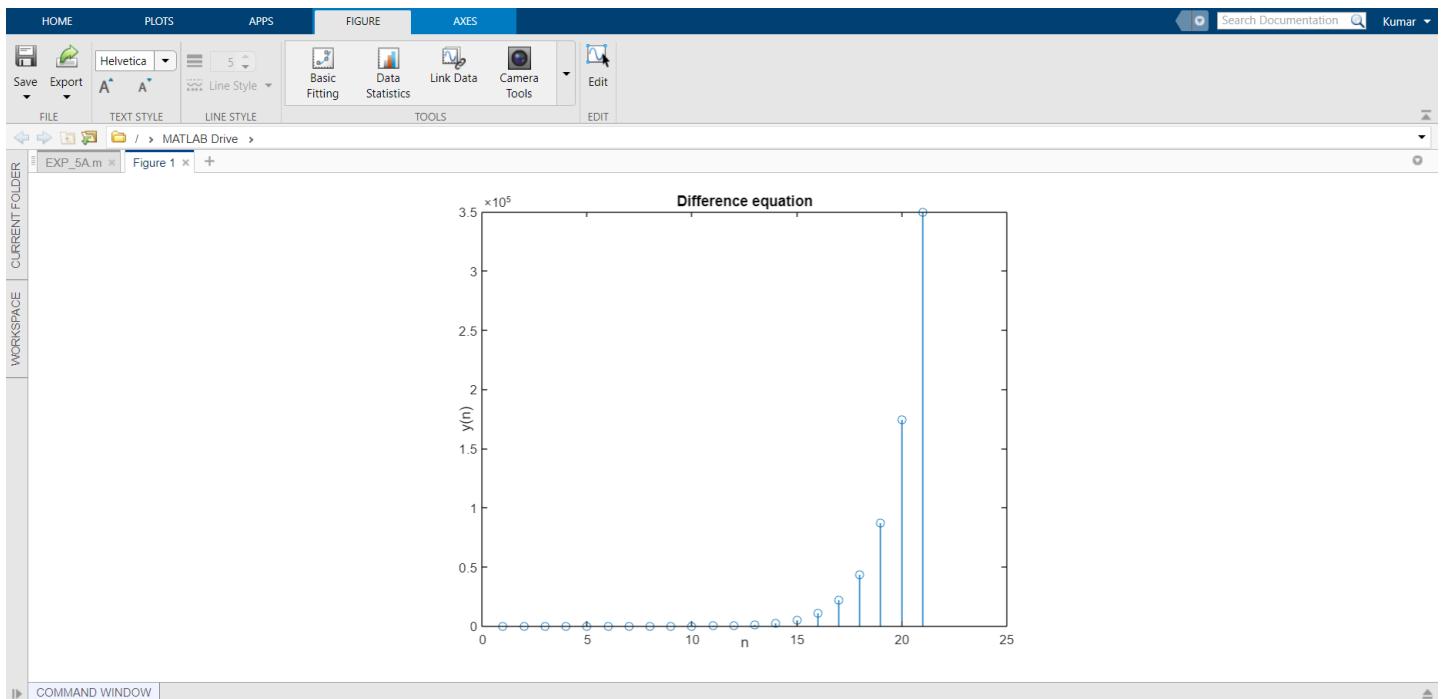


Figure 2 : Graph Output

3. Solve $y(n+2) + 2y(n+1) + y(n) = n, n \geq 0$, $y_0 = 0$ and $y_1 = 0$.

Sol:

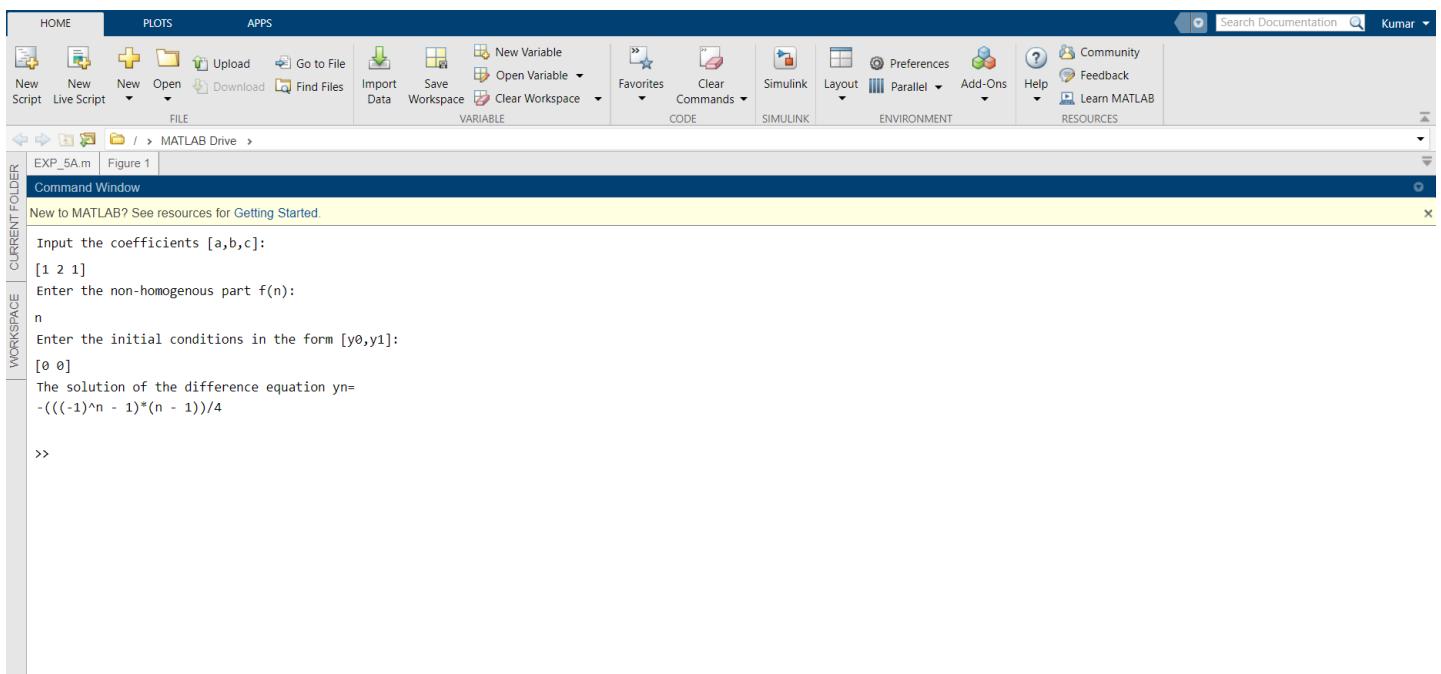


Figure 3 : Command Window Output

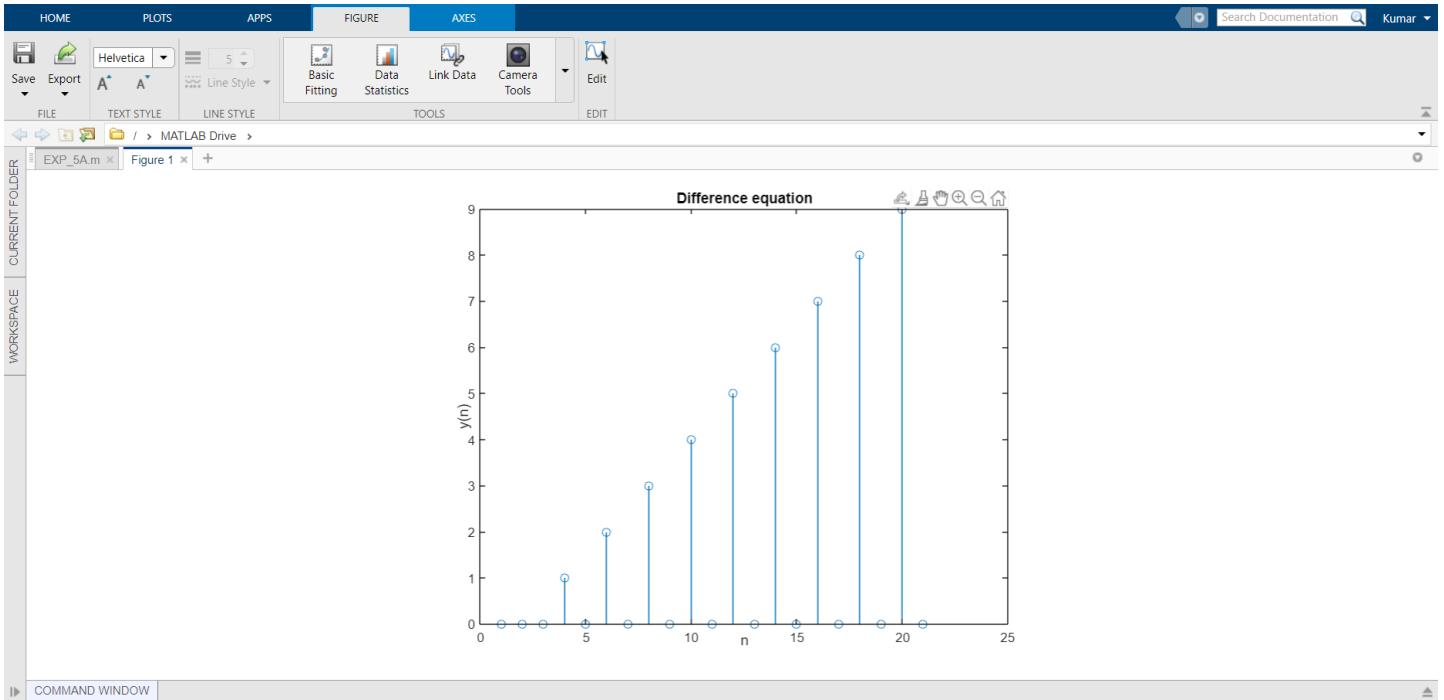


Figure 4 : Graph Output

4. Solve $y(n+2) - 4y(n+1) + 3y(n) = n \cdot 2^n, n \geq 0, y_0 = 0$ and $y_1 = 0$.

Sol:

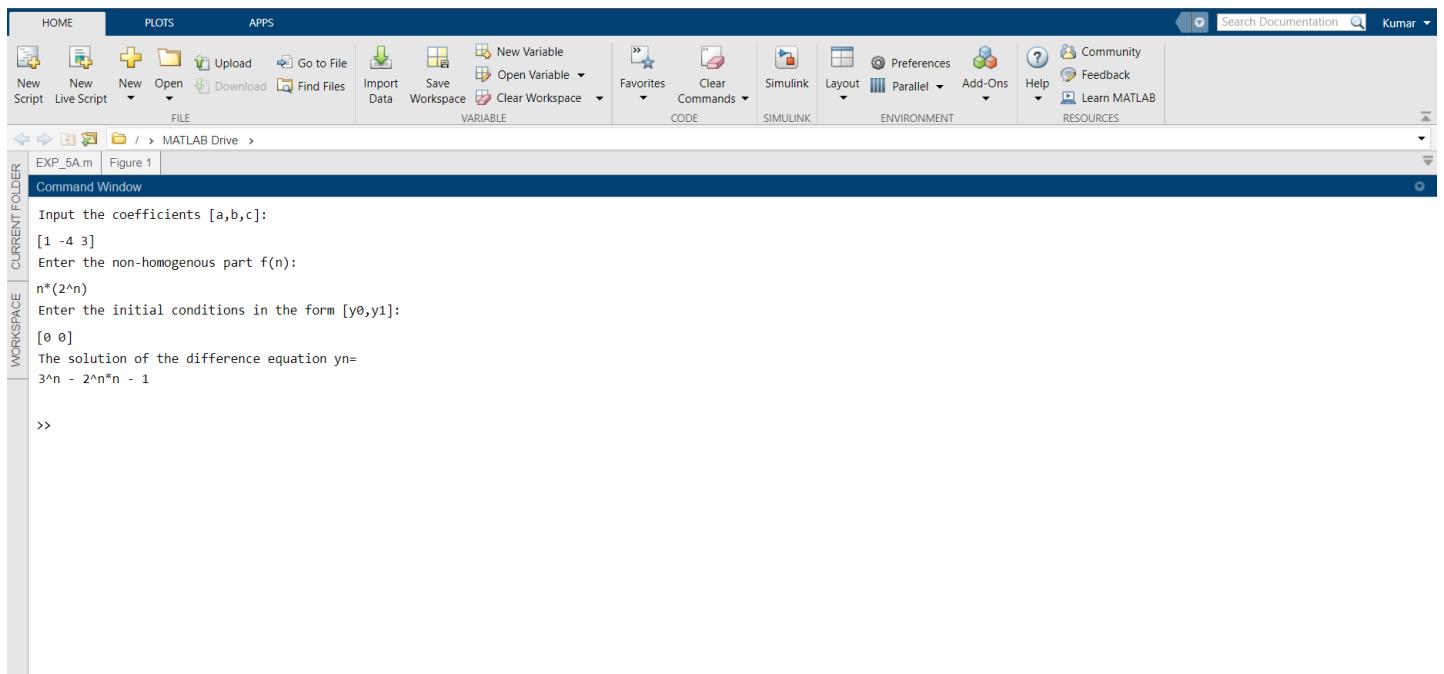


Figure 5 : Command Window Output

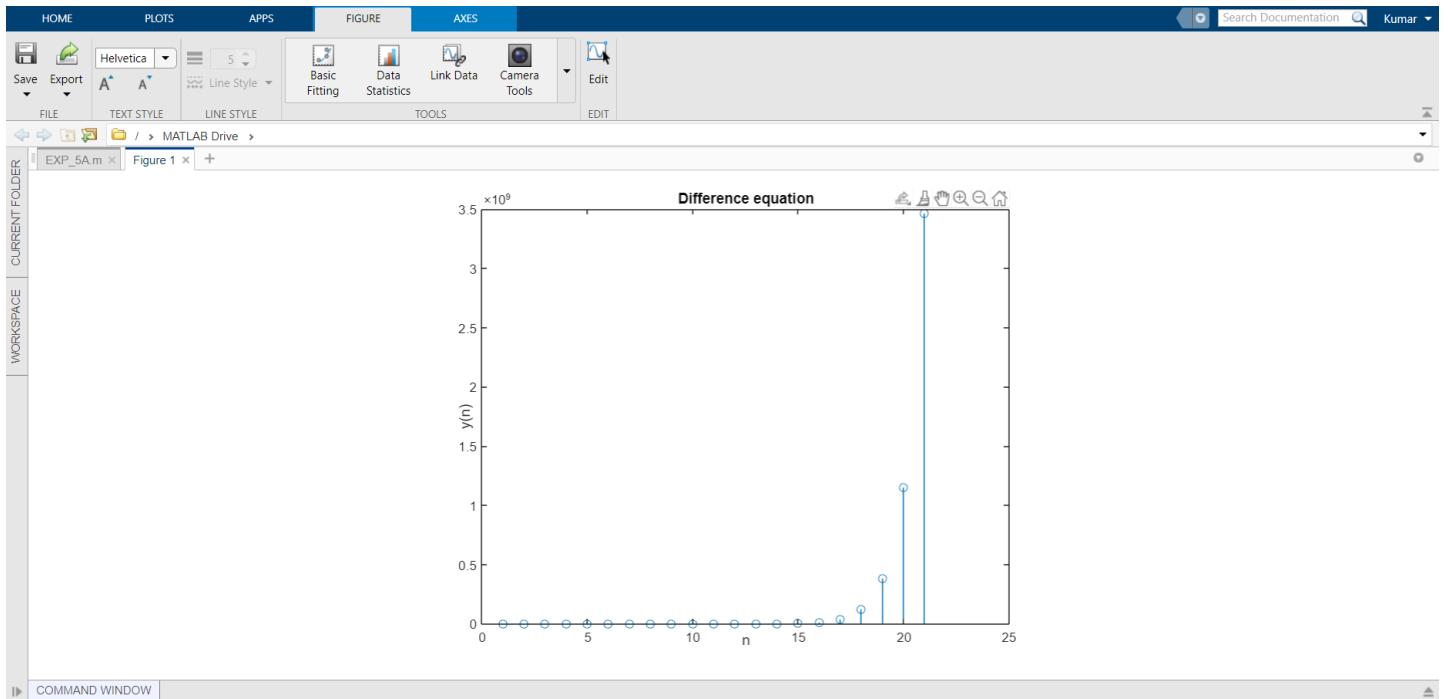


Figure 6 : Graph Output

5. Formulate the difference equation for Fibonacci numbers and hence solve by Z-transforms.

Sol:

$$f(n+2) = f(n+1) + f(n), \quad f(0) = 0, \quad f(1) = 1.$$

Figure 7 : Function for Fibonacci Sequence with Initial Condition

$$y_{n+2} - y_{n+1} - y_n = 0.$$

Figure 8 : Difference Equation for Fibonacci numbers

```

Input the coefficients [a,b,c]:
[1 -1 -1]
Enter the non-homogenous part f(n):
0
Enter the initial conditions in the form [y0,y1]:
[0 1]
The solution of the difference equation y(n)=
2*(-1)^(n/2)*cos(n*(pi/2 + asin(i1/2))) - (2^2*(1 - n)*5^(1/2)*(5^(1/2) + 1)^(n - 1))/5 + (2^(2 - n)*5^(1/2)*(1 - 5^(1/2))^(n - 1))/5

Warning: Using only the real component of complex data.
> In matlab.graphics.chart.internal.getRealData (line 52)
In stem (line 40)
In EXP_5A (line 24)
>>

```

Figure 9 : Command Window Output

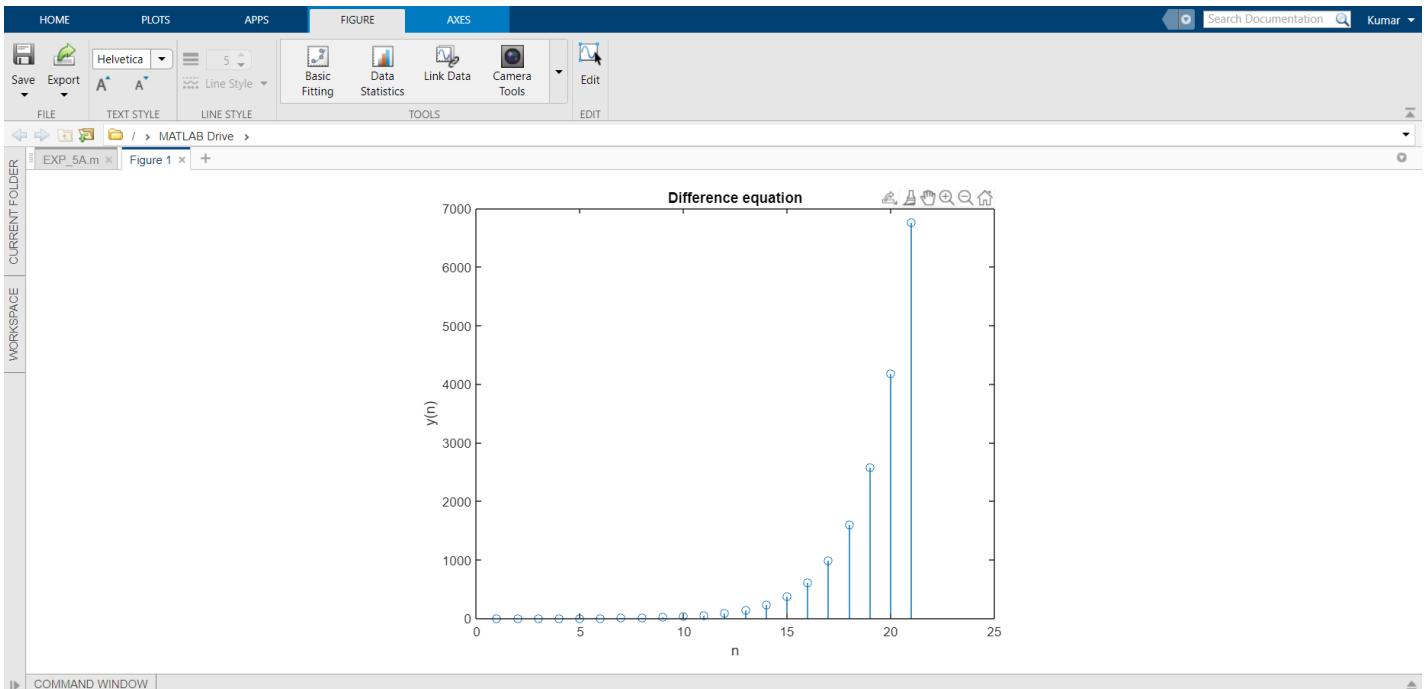


Figure 10 : Graph Output

Department of Mathematics
School of Advanced Sciences
MAT 2002 – Applications of Differential and Difference Equations (MATLAB)
Experiment 5-B
Solution of Homogeneous Linear Difference equations

Consider a second order homogeneous linear difference equation of the form

$$ay_{n+2} + by_{n+1} + cy_n = 0 \quad (1)$$

subject to the initial conditions

$$y_0 = \alpha, \quad y_1 = \beta \quad (2)$$

where the a, b, c are constants.

To solve the equation (1) we take the trial solution $y_n = \lambda^n$. Substituting in (1) we get the quadratic equation

$$a\lambda^2 + b\lambda + c = 0. \quad (3)$$

This is called the characteristic equation of (1).

The solutions of (1) are given as per the following cases:

Case 1. If $b^2 - 4ac > 0$, then the roots of (3) are real and different (say λ_1, λ_2).

In this case the solutions of (1) are $y_1 = \lambda_1^n$ and $y_2 = \lambda_2^n$ and hence the general solution of (1) is given by $y_n = k_1 y_1 + k_2 y_2 = k_1 \lambda_1^n + k_2 \lambda_2^n$.

Case 2. If $b^2 - 4ac = 0$, then the roots (say $\lambda_1 = \lambda_2$) of (3) are real and equal.

In this case the solutions of (1) are $y_1 = \lambda_1^n$ and $y_2 = n \cdot \lambda_1^n$ and hence the general solution of (1) is given by $y_n = k_1 y_1 + k_2 y_2 = k_1 \lambda_1^n + k_2 n \cdot \lambda_1^n$.

Case 3. If $b^2 - 4ac < 0$, then the roots of (3) are complex (say $\lambda_1 = \alpha + i\beta, \lambda_2 = \alpha - i\beta$).

In this case the solutions of (1) are $y_1 = \rho^n \cos n\theta$ and $y_2 = \rho^n \sin n\theta$, where $\rho = \sqrt{\alpha^2 + \beta^2}$, $\theta = \tan^{-1}(\beta/\alpha)$ and hence the general solution of (1) is given by

$$y_n = k_1 y_1 + k_2 y_2 = \rho^n [k_1 \cos n\theta + k_2 \sin n\theta]$$

MATLAB Code

```
clear all
clc
syms n k1 k2 L
F = input('Input the coefficients [a,b,c]: ');
a=F(1);b=F(2);c=F(3);
ch_eqn=a*L^2+b*L+c; %Characteristic equation
L=solve(ch_eqn);
L1=L(1);L2=L(2);
D=b^2-4*a*c;
if(D>0) % Roots are real and different
    y1=L1^n;
    y2=L2^n;
elseif (D==0)% Roots are real and equal
    y1=L1^n;
    y2=n*L1^n;
else % Roots are complex
    rho=abs(L1); t=angle(L1);
    y1 = (rho^n)*cos(n*t);
    y2 = (rho^n)*sin(n*t);
end
yn = k1*y1+k2*y2;
check=input('If initial conditions are known, then enter 1 else
enter 0: ');
if (check == 1)
    IC=input('Enter the initial conditions [y(0),y(1)]');
    eq1=(subs(yn,n,0)-IC(1));
    eq2=(subs(yn,n,1)-IC(2));
    [k1,k2]=solve(eq1,eq2);
    yn=simplify(subs(yn));
    m=0:20;
    y=subs(yn,n,m);
    stem(y)
    title('Difference equation');
    xlabel('n'); ylabel('y(n)');
end
disp('The Solution of the given Homogeneous equation is y_n= ');
disp(collect(collect(yn,y1),y2))
```

Example 1. The deer population of a region was 200 at a certain time. After 1 year the population increased to 220. Given that the increase in population from $(n+1)^{\text{st}}$ and $(n+2)^{\text{nd}}$ years is twice the increase from n^{th} and $(n+1)^{\text{st}}$ years. Write a recurrence relation that defines the deer population at time n and hence solve it.

Solution: Let y_n denote the deer population at time n . Given that $y_0 = 200$, $y_1 = 220$.

Also given that $y_{n+2} - y_{n+1} = 2(y_{n+1} - y_n)$, which may be rewritten as

$$y_{n+2} - 3y_{n+1} + 2y_n = 0, \quad n \geq 0. \quad (1.1)$$

The characteristic equation is $\lambda^2 - 3\lambda + 2 = 0$, whose roots are 1 and 2.

The general solution of the equation (1.1) is $y_n = k_1(1)^n + k_2(2)^n$.

Substituting the conditions $y_0 = 200$, $y_1 = 220$ we get

$k_1 + k_2 = 200$ and $k_1 + 2k_2 = 220$, solving we get $k_1 = 180$, $k_2 = 20$.

Thus the particular solution is $y_n = 180 + 20(2)^n$ denotes the population at any time n (years).

Input

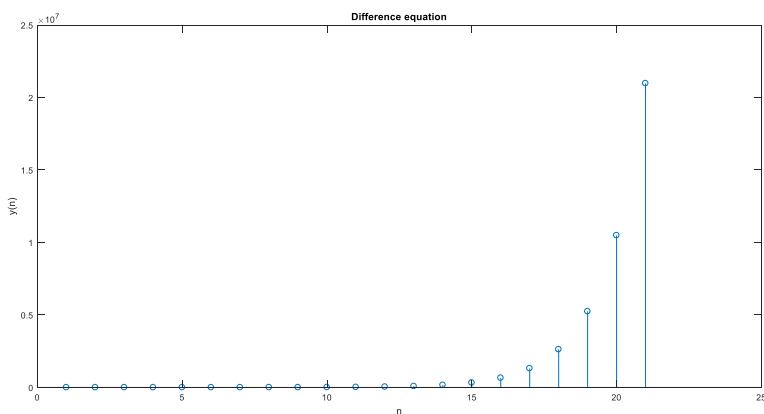
Input the coefficients [a,b,c]: [1 -3 2]

If initial conditions are known, then enter 1 else enter 0: 1

Enter the initial conditions [y(0),y(1)] [200 220]

Output

The Solution of the given Homogeneous equation is $y_n = 20 \cdot 2^n + 180$



Example 2: Find the general solution of the difference equation $y_{n+2} - y_{n+1} + y_n = 0$.

Input

Input the coefficients [a,b,c]: [1 -1 1]

If initial conditions are known, then enter 1 else enter 0: 0

Output

The Solution of the given Homogeneous equation is $y_n = k1 \cdot \cos(\pi n / 3) - k2 \cdot \sin(\pi n / 3)$

Exercise

1. Find the complete solution of the following difference equations

- a. $y_{n+2} - 9y_n = 0$
- b. $y_{n+2} + 4y_{n+1} + y_n = 0$
- c. $y_{n+2} + 4y_{n+1} + 4y_n = 0$
- d. $y_{n+2} - 2y_{n+1} + 2y_n = 0$

2. Solve the following difference equation subject to the given conditions

- a. $y_{n+2} - 6y_{n+1} + 8y_n = 0$, $y_0 = 1$ and $y_1 = 0$.
- b. $2y_{n+2} - 7y_{n+1} + 3y_n = 0$, $y_0 = 1$ and $y_1 = 1$.
- c. $y_{n+2} + 8y_{n+1} + 16y_n = 0$, $y_0 = 2$ and $y_1 = -20$.

MAT2002 – Applications of Differential & Difference Equations

Solution of Homogeneous Linear Difference equations

(Exp – 5B)

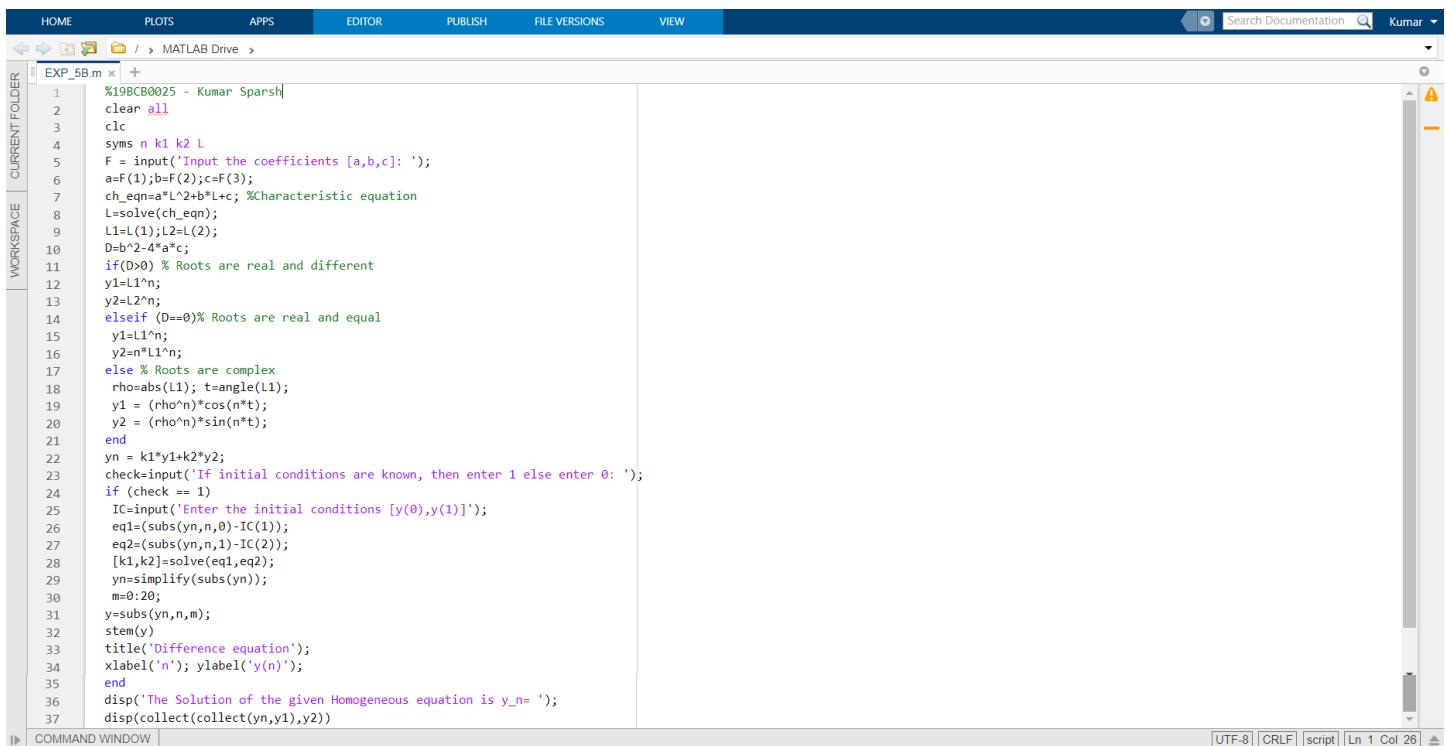
Faculty Name -
Monica C

Presented By -
Kumar Sparsh
19BCB0025

1. Find the complete solution of the following difference equations

- a. $y_{n+2} - 9y_n = 0$
- b. $y_{n+2} + 4y_{n+1} + y_n = 0$
- c. $y_{n+2} + 4y_{n+1} + 4y_n = 0$
- d. $y_{n+2} - 2y_{n+1} + 2y_n = 0$

Sol:



The screenshot shows the MATLAB desktop interface. The current folder is set to 'EXP_5B.m'. The code in the editor is as follows:

```

1 %19BCB0025 - Kumar Sparsh
2 clear all
3 clc
4 syms n k1 k2 L
5 F = input('Input the coefficients [a,b,c]: ');
6 a=F(1);b=F(2);c=F(3);
7 ch_eqn=a*L^2+b*L+c; %Characteristic equation
8 L=solve(ch_eqn);
9 L1=L(1);L2=L(2);
10 D=b^2-4*a*c;
11 if(D>0) % Roots are real and different
12 y1=L1^n;
13 y2=L2^n;
14 elseif (D==0)% Roots are real and equal
15 y1=L1^n;
16 y2=n*L1^n;
17 else % Roots are complex
18 rho=abs(L1); t=angle(L1);
19 y1 = (rho^n)*cos(n*t);
20 y2 = (rho^n)*sin(n*t);
21 end
22 yn = k1*y1+k2*y2;
23 check=input('If initial conditions are known, then enter 1 else enter 0: ');
24 if (check == 1)
25 IC=input('Enter the initial conditions [y(0),y(1)]');
26 eq1=subs(yn,n,0)-IC(1);
27 eq2=subs(yn,n,1)-IC(2);
28 [k1,k2]=solve(eq1,eq2);
29 yn=simplify(subs(yn));
30 m=20;
31 y=subs(yn,n,m);
32 stem(y)
33 title('Difference equation');
34 xlabel('n'); ylabel('y(n)');
35 end
36 disp('The Solution of the given Homogeneous equation is y_n= ');
37 disp(collect(collect(yn,y1),y2))

```

Figure 1 : MATLAB Code for all following questions

a.

The screenshot shows the MATLAB Command Window interface. The menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. The EDITOR tab is active. The workspace browser on the left shows 'CURRENT FOLDER' and 'WORKSPACE'. The command window displays the following text:

```
EXP_5B.m
Command Window
Input the coefficients [a,b,c]:
[1 0 -9]
If initial conditions are known, then enter 1 else enter 0:
0
The Solution of the given Homogeneous equation is y_n=
k2*3^n + k1*(-3)^n

>> |
```

At the bottom right of the command window, there are buttons for UTF-8, CRLF, script, Ln 1 Col 26, and a dropdown arrow.

Figure 2 : 1 (a) - Command Window Output

b.

The screenshot shows the MATLAB Command Window interface. The menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. The EDITOR tab is active. The workspace browser on the left shows 'CURRENT FOLDER' and 'WORKSPACE'. The command window displays the following text:

```
EXP_5B.m
Command Window
Input the coefficients [a,b,c]:
[1 4 1]
If initial conditions are known, then enter 1 else enter 0:
0
The Solution of the given Homogeneous equation is y_n=
k2*(3^(1/2) - 2)^n + k1*(- 3^(1/2) - 2)^n

>> |
```

At the bottom right of the command window, there are buttons for UTF-8, CRLF, script, Ln 1 Col 26, and a dropdown arrow.

Figure 3 : 1 (b) - Command Window Output

c.

EXP_5B.m

Command Window

```
Input the coefficients [a,b,c]:
[1 4 4]
If initial conditions are known, then enter 1 else enter 0:
0
The Solution of the given Homogeneous equation is y_n=
(k1 + k2*n)*(-2)^n

>>
```

UTF-8 | CRLF | script | Ln 1 Col 26

Figure 4 : 1 (c) - Command Window Output

d.

EXP_5B.m

Command Window

```
Input the coefficients [a,b,c]:
[1 -2 2]
If initial conditions are known, then enter 1 else enter 0:
0
The Solution of the given Homogeneous equation is y_n=
2^(n/2)*k1*cos((pi*n)/4) - 2^(n/2)*k2*sin((pi*n)/4)

>>
```

UTF-8 | CRLF | script | Ln 1 Col 26

Figure 5 : 1 (d) - Command Window Output

2. Solve the following difference equation subject to the given conditions

- $y_{n+2} - 6y_{n+1} + 8y_n = 0$, $y_0 = 1$ and $y_1 = 0$.
- $2y_{n+2} - 7y_{n+1} + 3y_n = 0$, $y_0 = 1$ and $y_1 = 1$.
- $y_{n+2} + 8y_{n+1} + 16y_n = 0$, $y_0 = 2$ and $y_1 = -20$.

Sol:

(a)

The screenshot shows the MATLAB interface with the Command Window active. The command `[1 -6 8]` is entered, followed by a prompt for initial conditions. The user enters `1` for y_0 . The solution is then displayed as $2^{*}2^n - 2^{*}(2^{*}n)$.

```
EXP_5B.m | Figure 1 | Command Window
Input the coefficients [a,b,c]:
[1 -6 8]
If initial conditions are known, then enter 1 else enter 0:
1
Enter the initial conditions [y(0),y(1)]
[1 0]
The Solution of the given Homogeneous equation is y_n=
2^*2^n - 2^*(2^*n)

>>
```

Figure 6 : 2 (a) - Command Window Output

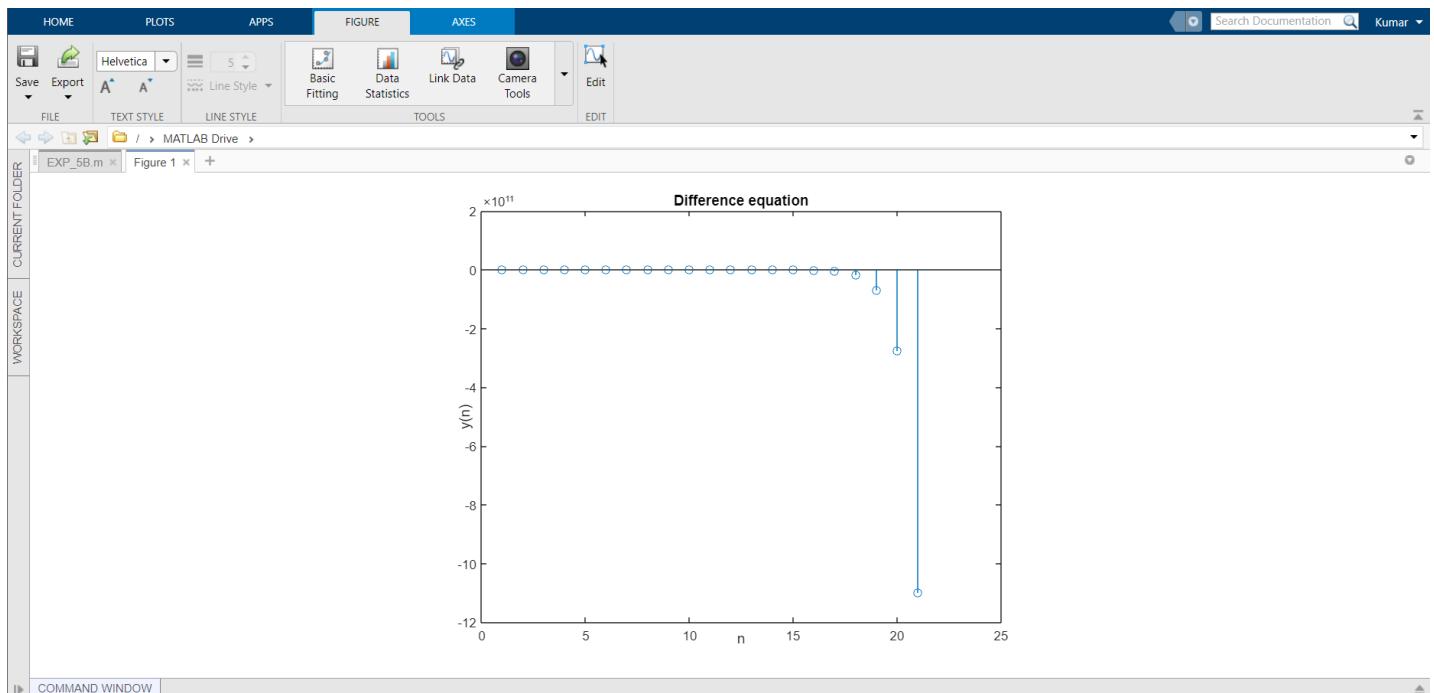


Figure 7 : 2 (a) - Graph Output

b.

```

Input the coefficients [a,b,c]:
[2 -7 3]
If initial conditions are known, then enter 1 else enter 0:
1
Enter the initial conditions [y(0),y(1)]
[1 1]
The solution of the given Homogeneous equation is y_n=
(6^n/5 + 4/5)/2^n
>>

```

The figure shows the MATLAB Command Window interface. The menu bar includes HOME, PLOTS, APPS, FIGURE (selected), AXES, FILE, TEXT STYLE, LINE STYLE, TOOLS, and EDIT. The toolbar includes Save, Export, Basic Fitting, Data Statistics, Link Data, Camera Tools, and Edit. The current folder is set to MATLAB Drive > EXP_5B.m. The command window displays the following text:

Input the coefficients [a,b,c]:
[2 -7 3]
If initial conditions are known, then enter 1 else enter 0:
1
Enter the initial conditions [y(0),y(1)]
[1 1]
The solution of the given Homogeneous equation is y_n=
 $(6^n/5 + 4/5)/2^n$
>>

Figure 8 : 2 (b) - Command Window Output

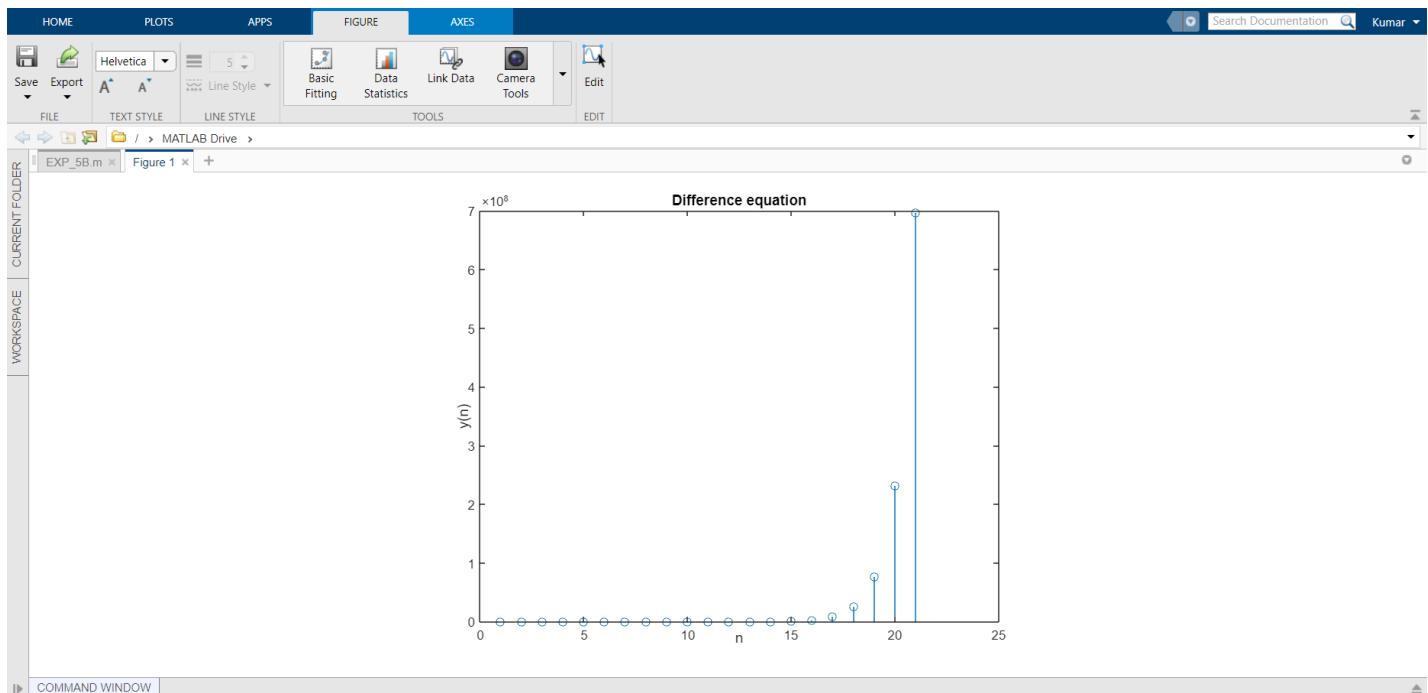


Figure 10 : 2 (b) - Graph Output

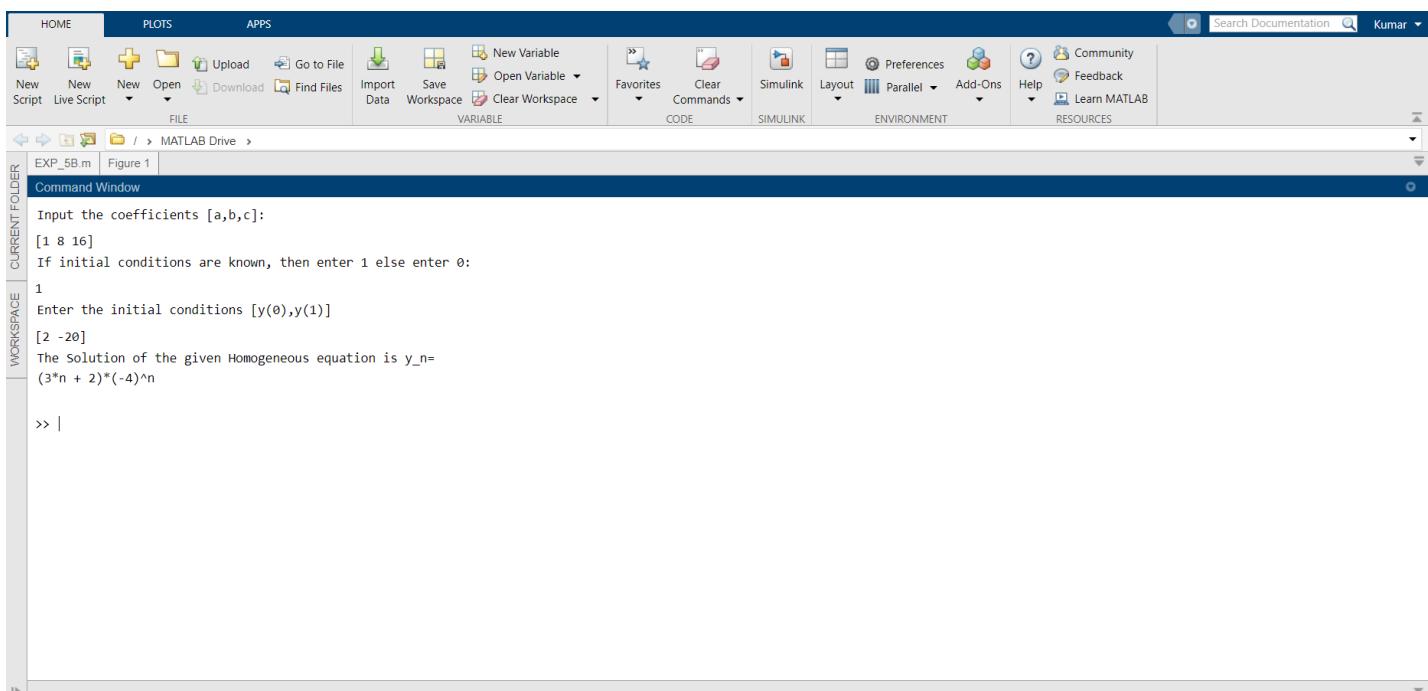
C.

Figure 9 : 2 (c) - Command Window Output

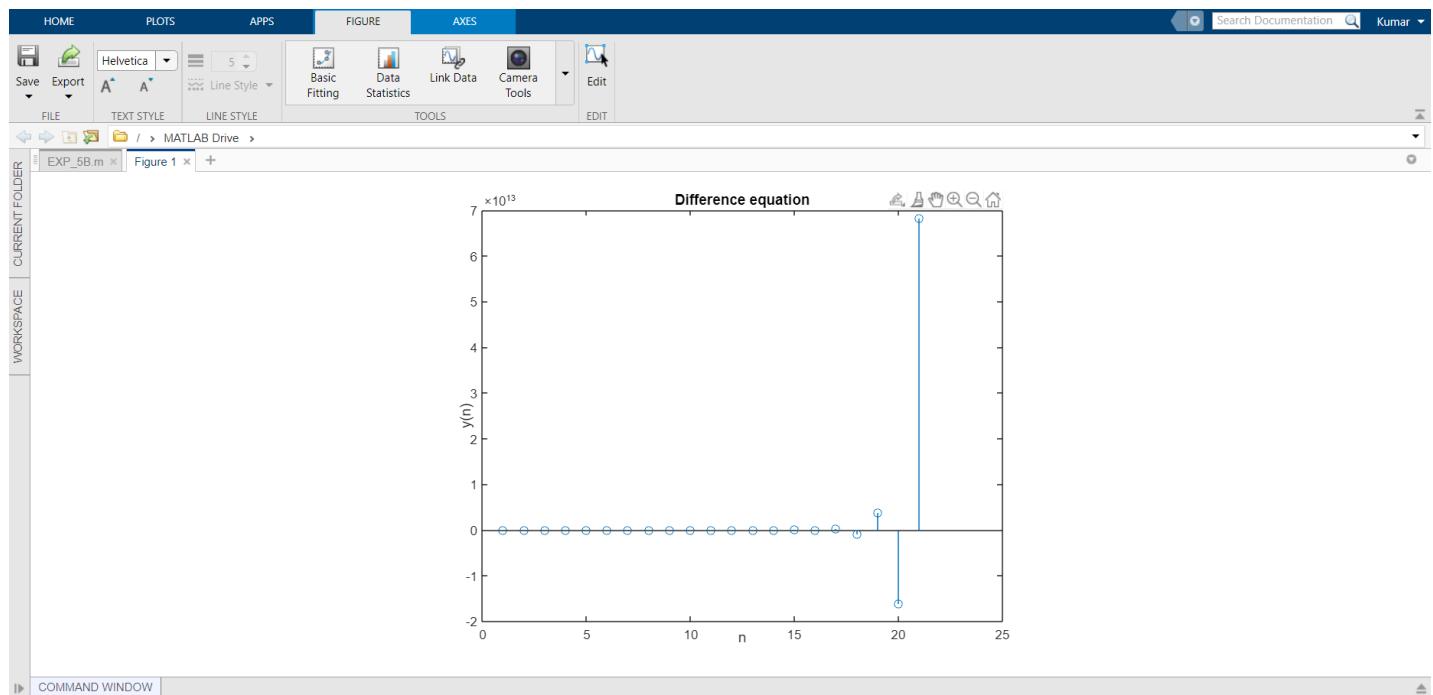


Figure 11 : 2 (c) - Graph Output