MATLAB ACTIVITY BOOK FOR THIRD SEMESTER

Table of Contents

Activity 1: Fourier Series

Definition: **Fourier series of a piecewise continuous function $f(x)$ in the interval $[a, a+2l]$ is given by:**

$$F\{f(x)\} = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi x}{l}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{l}\right)$$ **where,**

$$a_0 = \frac{1}{l}\int_a^{a+2l} f(x)\mathrm{dx} \quad , \quad a_n = \frac{1}{l}\int_a^{a+2l} f(x)\cos\left(\frac{n\pi x}{l}\right)\mathrm{dx} \quad , \quad b_n = \frac{1}{l}\int_a^{a+2l} f(x)\sin\left(\frac{n\pi x}{l}\right)\mathrm{dx}.$$

Question: **Write a matlab code to find the Fourier series of $f(x) = e^x$ in the interval $[-\pi, \pi]$ upto 25 terms. Visualize by obtaining obtaining the graphs of both the function and the series on the same plot.**

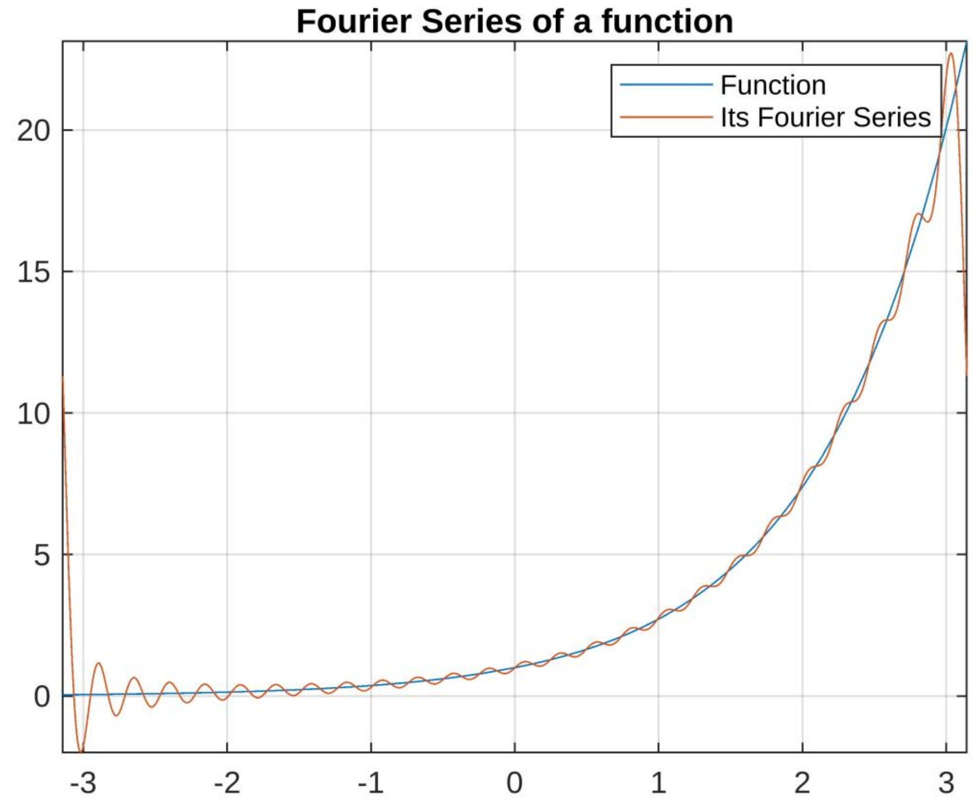
Code:

```
syms x
a=-pi; %lower limit of the interval
b=pi; %upper limit of the interval
I=[a b];
l=(b-a)/2; % the value of l
y = exp(x);

%Construction of Fourier Series
a0 = vpa((1/l)*int(y,x,a,b),3); %Formula for a0
i = 25;
sum=0;
for n=
1:i
an = (1/l)*int(y*cos((n*pi*x)/l),x,I); %Formula for an
bn = (1/l)*int(y*sin((n*pi*x)/l),x,I); %Formula for bn
sum = sum + an*cos((n*pi*x)/l) + bn*sin((n*pi*x)/l);
end
F=vpa((a0/2)+sum,3) %Required Fourier Series
```

F = $1.47\cos(2.0x) + 0.432\cos(4.0x) + 0.113\cos(8.0x) + 0.0286\cos(16.0x) - 0.0254\cos(17.0x) - 0.0897\cos(9.0x) +$

```
%Visualize by plotting
fplot(x,y,I)
grid on;
hold on;
fplot(x,F,I)
title('Fourier Series of a function')
legend('Function', 'Its Fourier Series')
hold off
```



EXERCISES

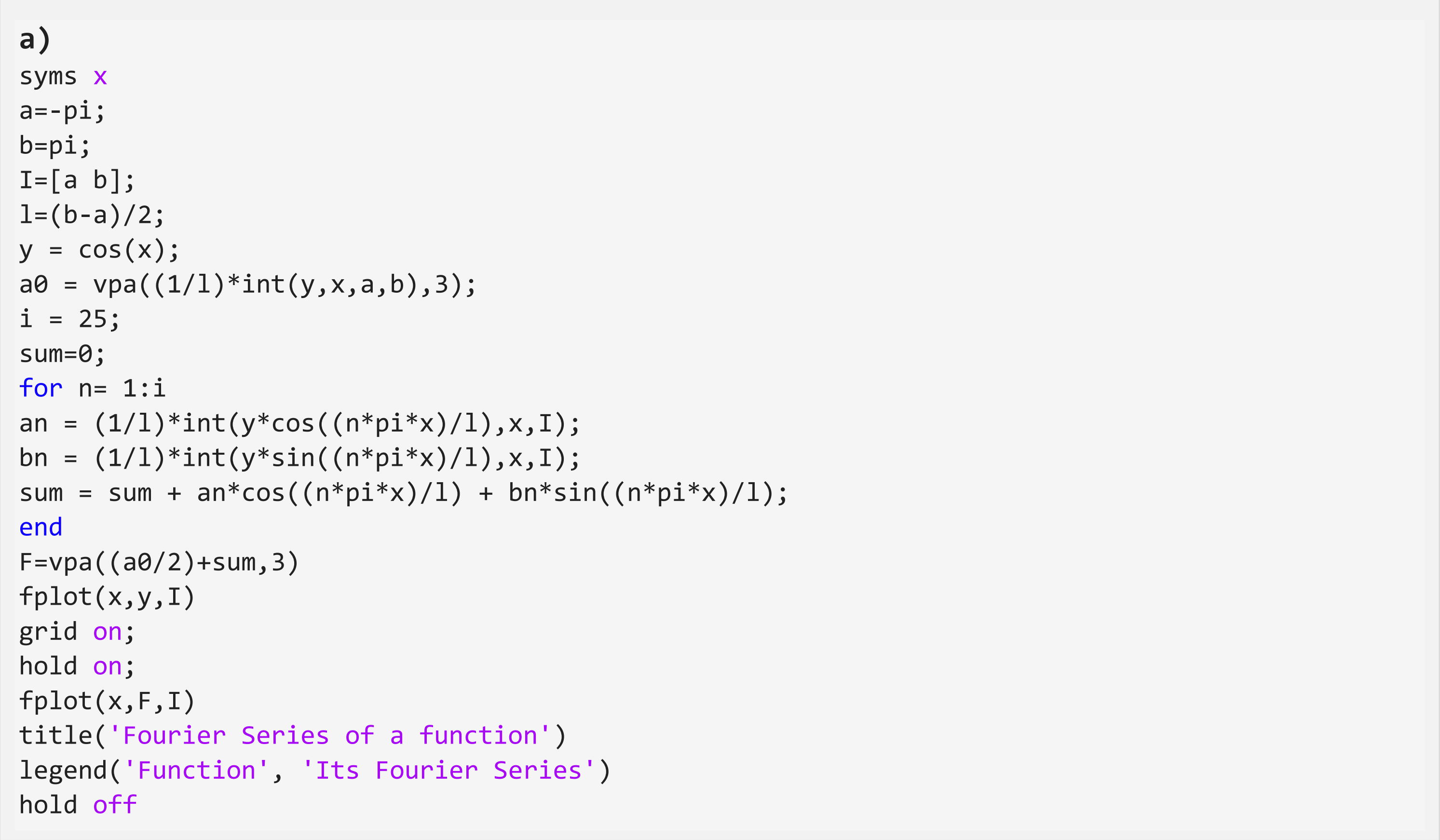
1. Solve the problem manually and verify with the series obtained in the output for the first three terms.

2. **Increase and decrease the value of 'i' in the above code. What is your observation on the accuracy of the series on changing the value of 'i' ?**

Increasing the value of i(i.e., the number of Fourier terms) increases the accuracy of the Fourier series approximation. However, diminishing returns occur after a certain point, as the additional terms contribute less to improving the accuracy.

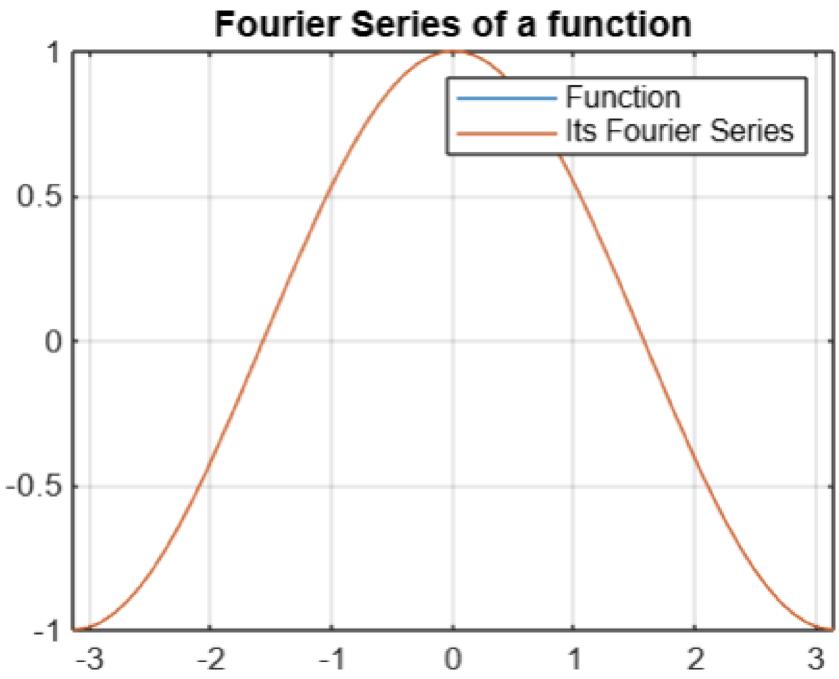
3. Write a Matlab code to obtain the Fourier series for the following functions and obtain the outputs and paste the graphs:

a) $\cos(x)$ **in the interval** $[-\pi, \pi]$



```
a)
syms x
a=-pi;
b=pi;
I=[a b];
l=(b-a)/2;
y = cos(x);
a0 = vpa((1/l)*int(y,x,a,b),3);
i = 25;
sum=0;
for n= 1:i
an = (1/l)*int(y*cos((n*pi*x)/l),x,I);
bn = (1/l)*int(y*sin((n*pi*x)/l),x,I);
sum = sum + an*cos((n*pi*x)/l) + bn*sin((n*pi*x)/l);
end
F=vpa((a0/2)+sum,3)
fplot(x,y,I)
grid on;
hold on;
fplot(x,F,I)
title('Fourier Series of a function')
legend('Function', 'Its Fourier Series')
hold off
```

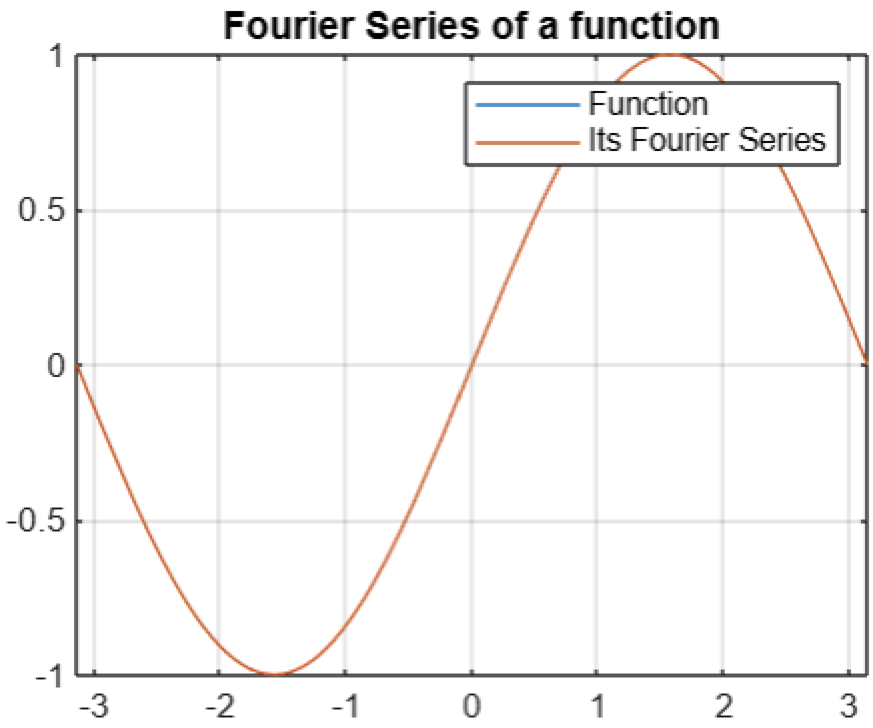
F=`1.0cos(x)



b) $\sin(x)$ **in the interval** $[-\pi, \pi]$

```
syms x
a=-pi;
b=pi;
I=[a b];
l=(b-a)/2;
y = sin(x);
a0 = vpa((1/l)*int(y,x,a,b),3);
i = 25;
sum=0;
for n= 1:i
an = (1/l)*int(y*cos((n*pi*x)/l),x,I);
bn = (1/l)*int(y*sin((n*pi*x)/l),x,I);
sum = sum + an*cos((n*pi*x)/l) + bn*sin((n*pi*x)/l);
end
F=vpa((a0/2)+sum,3)
fplot(x,y,I)
grid on;
hold on;
fplot(x,F,I)
title('Fourier Series of a function')
legend('Function', 'Its Fourier Series')
hold off
```

F=1.0sin(x)



Why is the value of b_n zero in a)?

The reason why b_n is zero for all n in this case is because y=cos(x) is an even function.

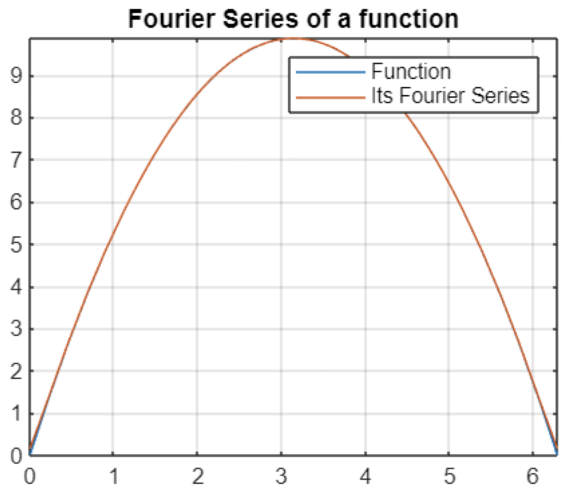
Why is the value of a_n zero in b)?

The reason why a_n is zero for all n in this case is because y=sin(x) is an odd function.

4. Obtain the outputs for the following functions and paste the graphs:

```
syms x
a=-pi;
b=pi;
I=[a b];
l=(b-a)/2;
y = x*(2*pi-x);
a0 = vpa((1/l)*int(y,x,a,b),3);
i = 25;
sum=0;
for n= 1:i
an = (1/l)*int(y*cos((n*pi*x)/l),x,I);
bn = (1/l)*int(y*sin((n*pi*x)/l),x,I);
sum = sum + an*cos((n*pi*x)/l) + bn*sin((n*pi*x)/l);
end
F=vpa((a0/2)+sum,3)
fplot(x,y,I)
grid on;
hold on;
fplot(x,F,I)
title('Fourier Series of a function')
legend('Function', 'Its Fourier Series')
hold off
```

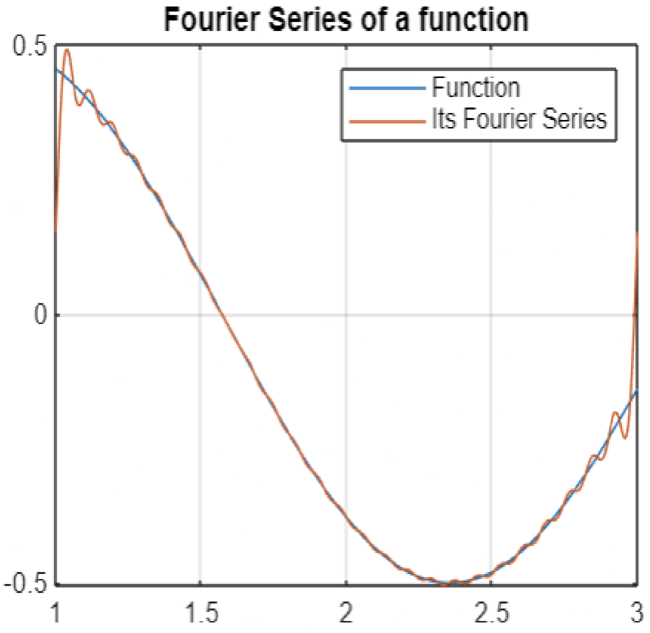
F=6.58-0.25cos(4.0x)-0.625cos(8.0x)-0.0156cos(16.0x)-0.0138cos(17.0x)-0.0494cos(9.0x)…



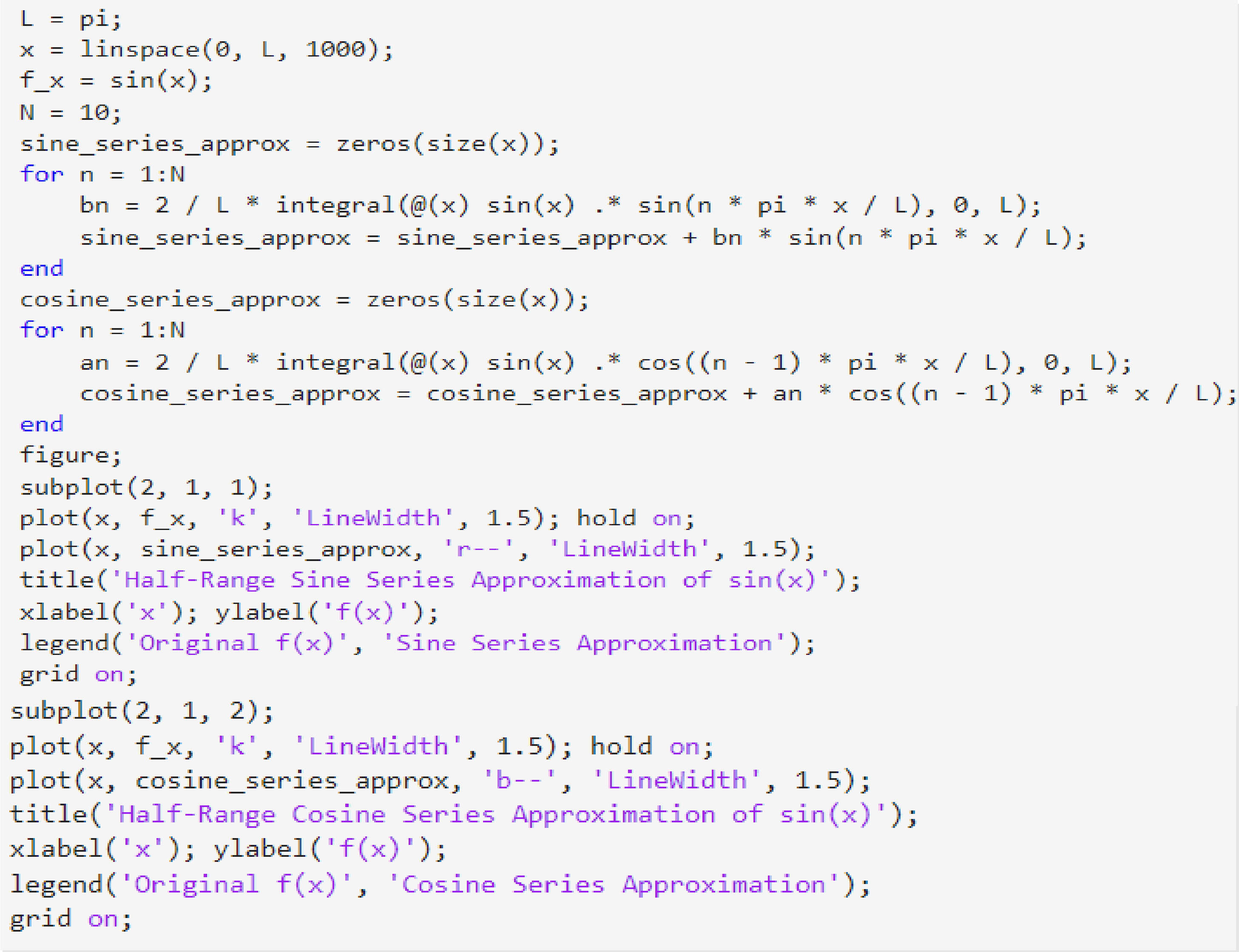
b) $\sin(x)\cos(x)$ **in the interval** $[1,3]$

```
syms x
a=-pi;
b=pi;
I=[a b];
l=(b-a)/2;
y = sin(x)*cos(x);
a0 = vpa((1/l)*int(y,x,a,b),3);
i = 25;
sum=0;
for n= 1:i
an = (1/l)*int(y*cos((n*pi*x)/l),x,I);
bn = (1/l)*int(y*sin((n*pi*x)/l),x,I);
sum = sum + an*cos((n*pi*x)/l) + bn*sin((n*pi*x)/l);
end
F=vpa((a0/2)+sum,3)
fplot(x,y,I)
grid on;
hold on;
fplot(x,F,I)
title('Fourier Series of a function')
legend('Function', 'Its Fourier Series')
hold off
```

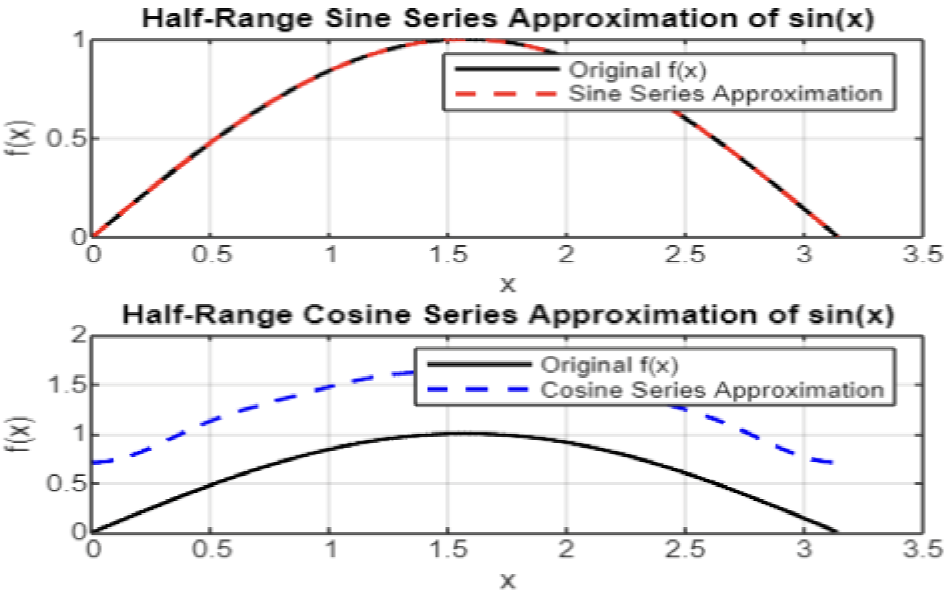
```
F=2.88e-4cos(69.1x)-0.00116cos(34.6x)-3.17e-4cos(66.0x)-2.64e-4cos(72.3x)-0.0162cos(9.42x)….
```



Extended activity (OE): **Write the code to obtain the half range sine and cosine series of the function sin(x) in (0,π)**



```matlab
L = pi;
x = linspace(0, L, 1000);
f_x = sin(x);
N = 10;
sine_series_approx = zeros(size(x));
for n = 1:N
    bn = 2 / L * integral(@(x) sin(x) .* sin(n * pi * x / L), 0, L);
    sine_series_approx = sine_series_approx + bn * sin(n * pi * x / L);
end
cosine_series_approx = zeros(size(x));
for n = 1:N
    an = 2 / L * integral(@(x) sin(x) .* cos((n - 1) * pi * x / L), 0, L);
    cosine_series_approx = cosine_series_approx + an * cos((n - 1) * pi * x / L);
end
figure;
subplot(2, 1, 1);
plot(x, f_x, 'k', 'LineWidth', 1.5); hold on;
plot(x, sine_series_approx, 'r--', 'LineWidth', 1.5);
title('Half-Range Sine Series Approximation of sin(x)');
xlabel('x'); ylabel('f(x)');
legend('Original f(x)', 'Sine Series Approximation');
grid on;
subplot(2, 1, 2);
plot(x, f_x, 'k', 'LineWidth', 1.5); hold on;
plot(x, cosine_series_approx, 'b--', 'LineWidth', 1.5);
title('Half-Range Cosine Series Approximation of sin(x)');
xlabel('x'); ylabel('f(x)');
legend('Original f(x)', 'Cosine Series Approximation');
grid on;
```



Additional questions by the course instructor(if any):

Total Maks Obtained in the activity (out of 10):	Signature of the faculty	Signature of the student

Activity 2: Harmonic Series

Explanation: We find the Harmonic series of a periodic function when the function itself is not known but for a finite number of functional values at some equidistant points are known as follows.

x	*x1*	*x2*	. . .	*xN*
f(x)=y	*y1*	*y2*	. . .	*yN*

For a function of period $2l$, the following is a very important observation:

$2l = \mathbf{Nxd}$. where **N - number of permissible values of x** and **d - common difference.**

Definition: The Harmonic series for a function whose values are given as above is given by:

$$F = \frac{a_0}{2} + \sum a_n \cos\left(\frac{n\pi x}{l}\right) + \sum b_n \sin\left(\frac{n\pi x}{l}\right)$$ **where,**

$$a_0 = \frac{2}{N}\Sigma y, \quad a_n = \frac{2}{N}\Sigma y \cos\left(\frac{n\pi x}{l}\right), \quad b_n = \frac{2}{N}\Sigma y \sin\left(\frac{n\pi x}{l}\right).$$

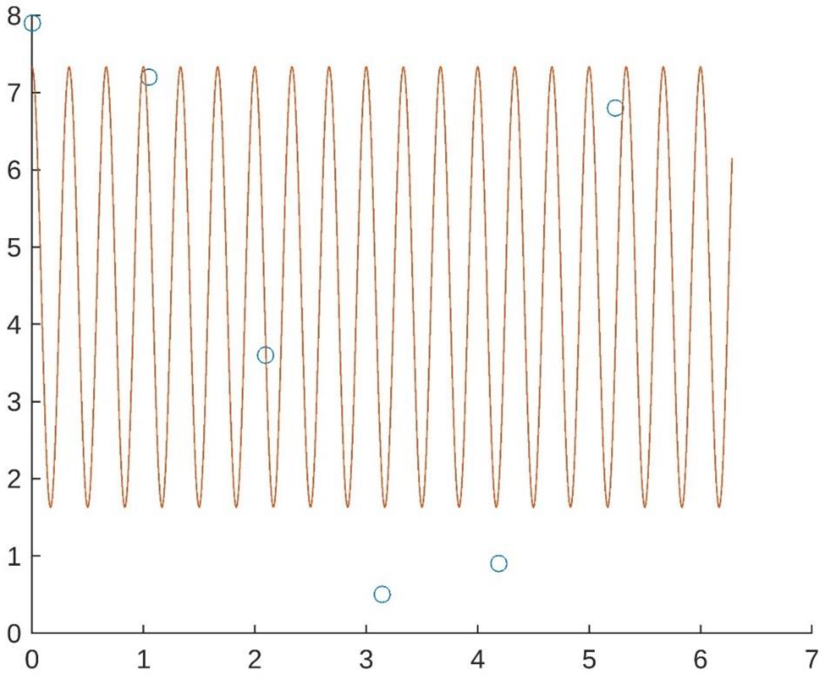
Question: Compute the harmonic series for

x	0	$\frac{\pi}{3}$	$\frac{2\pi}{3}$	π	$\frac{4\pi}{3}$	$\frac{5\pi}{3}$
y	7.9	7.2	3.6	0.5	0.9	6.8

```
clear
X=[0 pi/3 2*pi/3 pi 4*pi/3 5*pi/3]; % Only take the permissible values
y=[7.9 7.2 3.6 0.5 0.9 6.8];%Given data
h=pi/3; %Specify the common difference
N=length(X);
l=(N*h)/2;
a0 = (2/N)*sum(y); %Formula for a0
%Coefficients
for n=1:N
a(n)=(2/N)*sum(y.*cos((n*pi*X)/l)); %Formula for an
b(n)=(2/N)*sum(y.*sin((n*pi*X)/l)); %Formula for bn
end
%Series
syms x
for
n=1:N
F=a0/2+sum((a(n)*cos(n*pi*x)/l))+sum((b(n)*sin(n*pi*x)/l));
end
disp(F)
```

$$\frac{269\cos(6\pi x)}{30\pi}+\frac{4671179330349149\sin(6\pi x)}{1267650600228229401496703205376\,\pi}+\frac{269}{60}$$

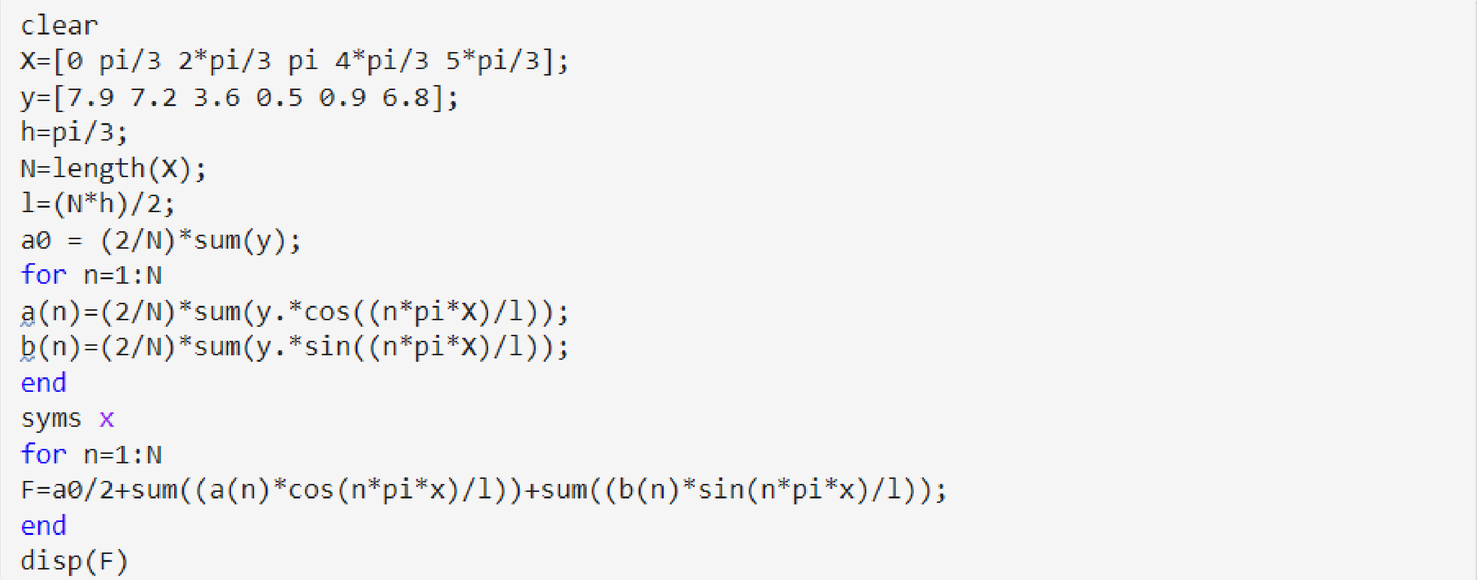
```
scatter(X,y)
hold on
fplot(x,F,[0,2*pi])
hold off
```



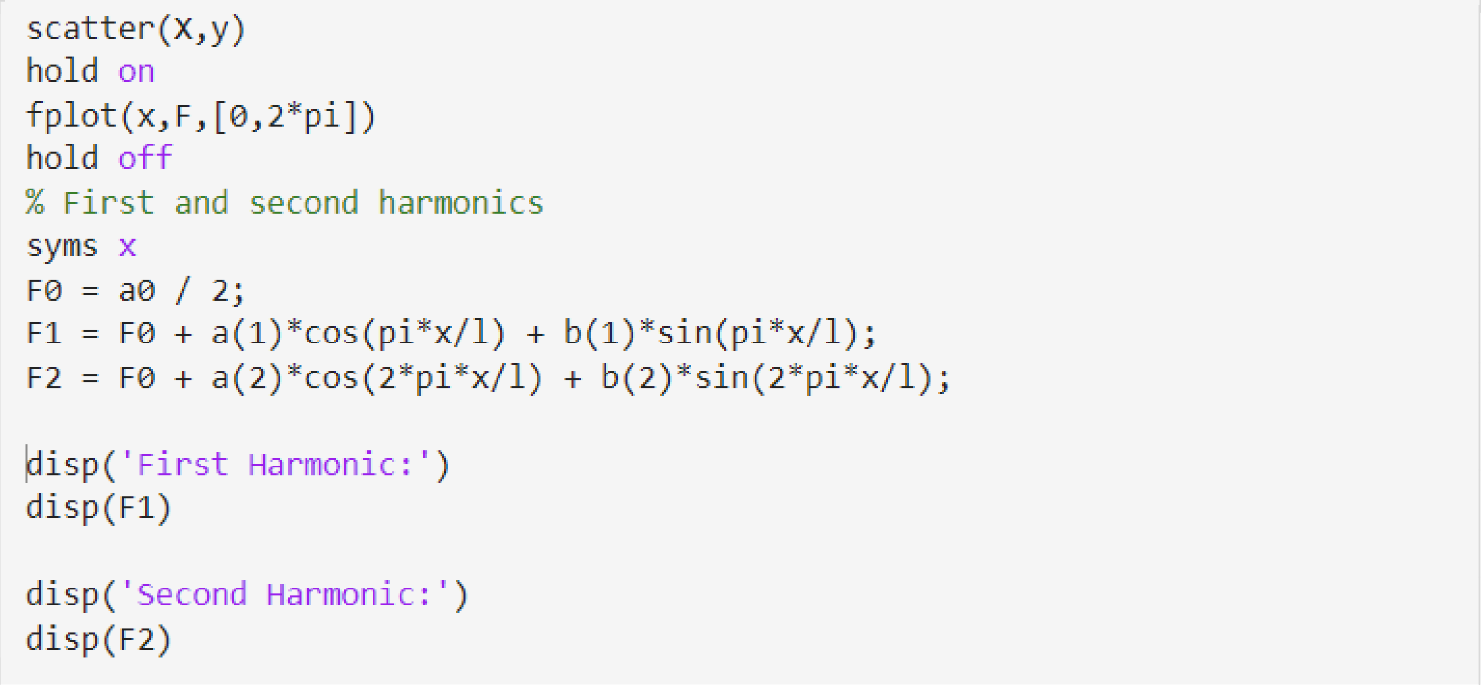
EXERCISES:

1. **Extract the first and second harmonics for the above function by writing the matlab code for it.**

Code



```
clear
X=[0 pi/3 2*pi/3 pi 4*pi/3 5*pi/3];
y=[7.9 7.2 3.6 0.5 0.9 6.8];
h=pi/3;
N=length(X);
l=(N*h)/2;
a0 = (2/N)*sum(y);
for n=1:N
a(n)=(2/N)*sum(y.*cos((n*pi*X)/l));
b(n)=(2/N)*sum(y.*sin((n*pi*X)/l));
end
syms x
for n=1:N
F=a0/2+sum((a(n)*cos(n*pi*x)/l))+sum((b(n)*sin(n*pi*x)/l));
end
disp(F)
```

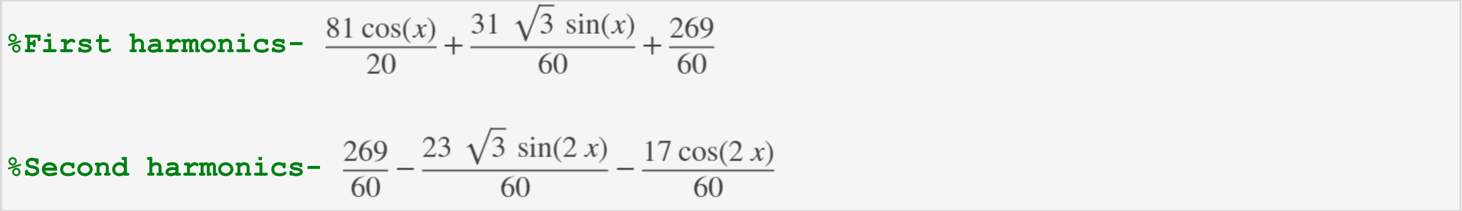


```
scatter(X,y)
hold on
fplot(x,F,[0,2*pi])
hold off
% First and second harmonics
syms x
F0 = a0 / 2;
F1 = F0 + a(1)*cos(pi*x/l) + b(1)*sin(pi*x/l);
F2 = F0 + a(2)*cos(2*pi*x/l) + b(2)*sin(2*pi*x/l);

disp('First Harmonic:')
disp(F1)

disp('Second Harmonic:')
disp(F2)
```

Output



%First harmonics- $\frac{81\cos(x)}{20} + \frac{31\sqrt{3}\sin(x)}{60} + \frac{269}{60}$

%Second harmonics- $\frac{269}{60} - \frac{23\sqrt{3}\sin(2x)}{60} - \frac{17\cos(2x)}{60}$

2. Solve for first and second harmonics manually and compare to verify with the above output.

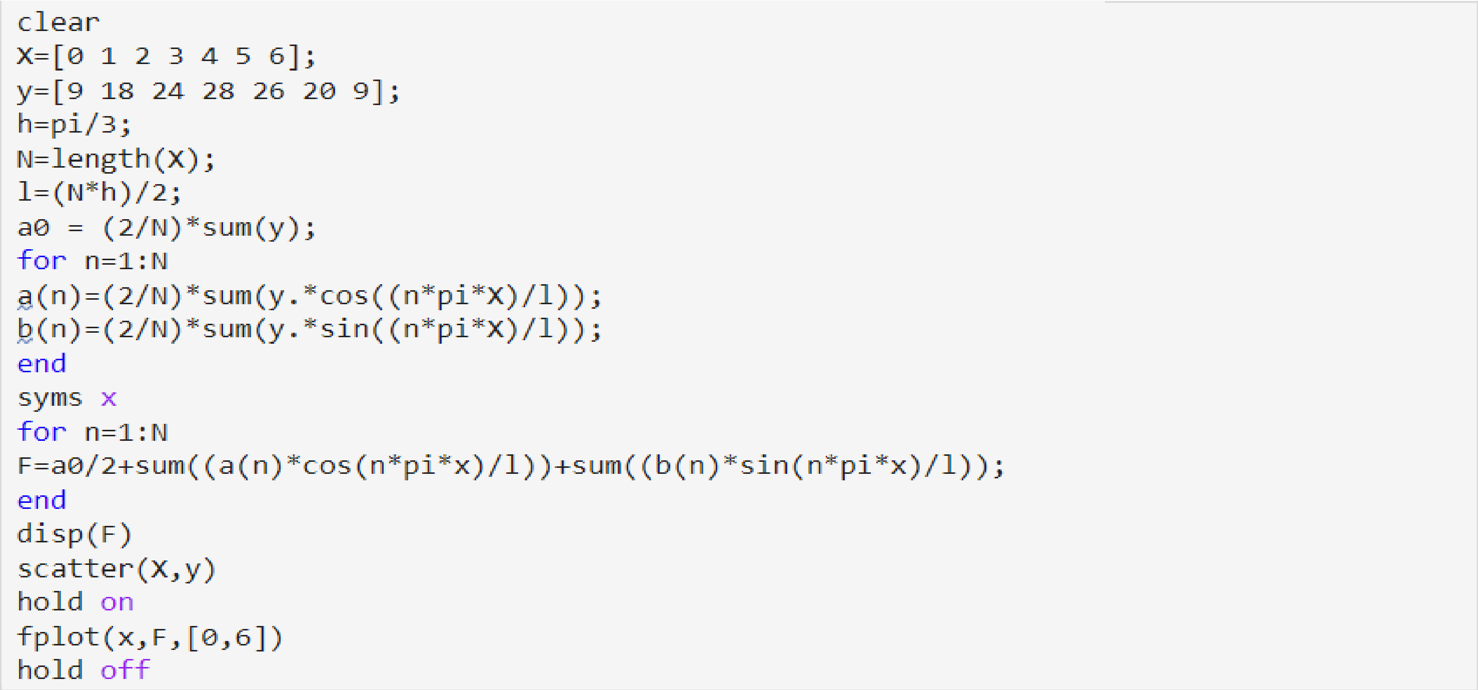
3. **What is the effect of N (Number of permissible values of x) on the accuracy of the Harmonic series? Explain.**

Increasing N (Number of permissible values of x) improves Fourier series accuracy by capturing finer details and higher harmonics, reducing approximation error and smoothing.

4. **Obtain the Harmonic series for the following data and hence paste the graphs.**

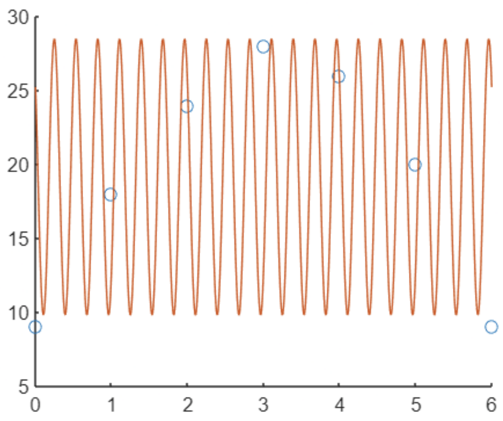
a)

x	0	1	2	3	4	5	6
y	9	18	24	28	26	20	9



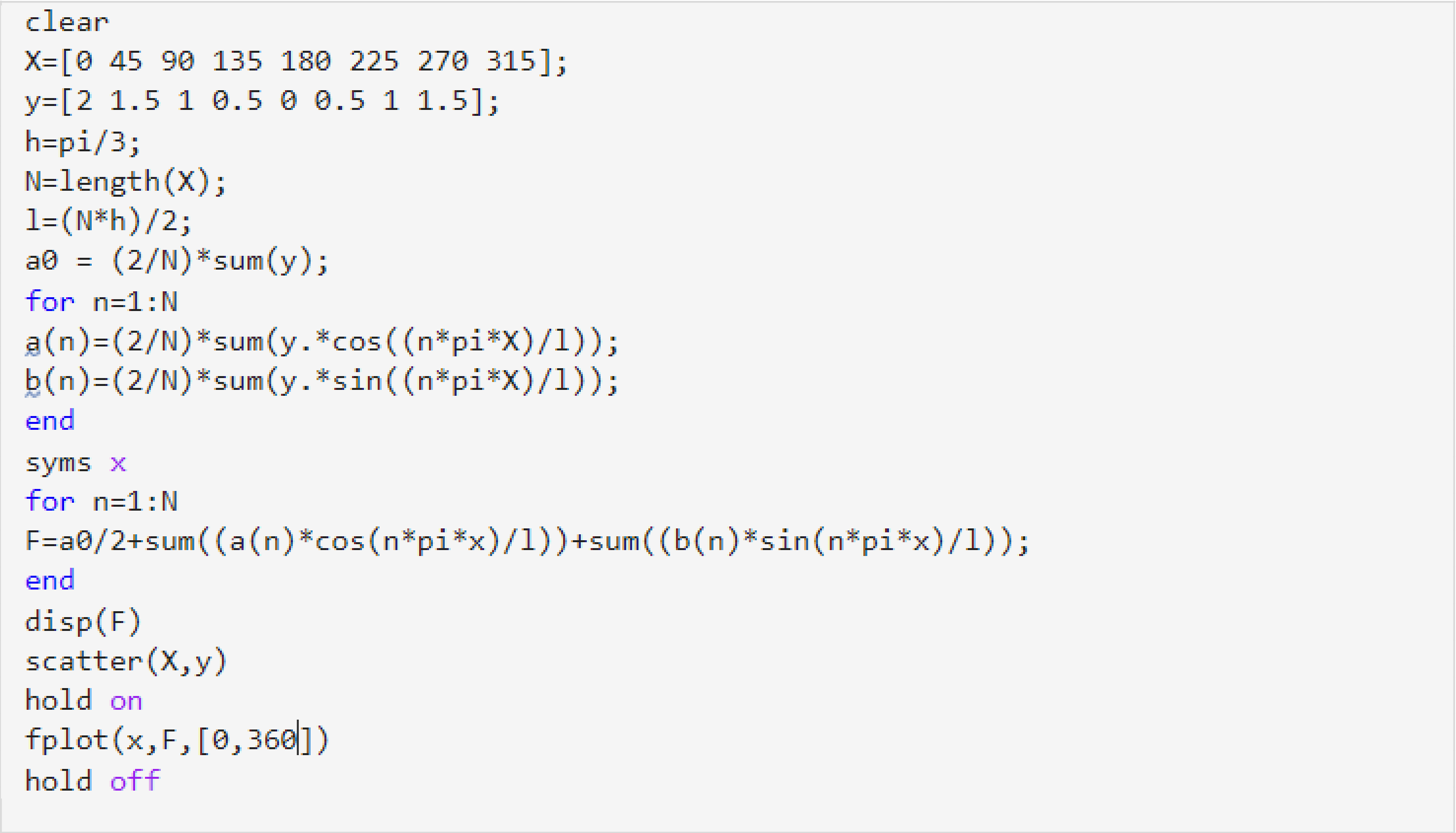
```
clear
X=[0 1 2 3 4 5 6];
y=[9 18 24 28 26 20 9];
h=pi/3;
N=length(X);
l=(N*h)/2;
a0 = (2/N)*sum(y);
for n=1:N
a(n)=(2/N)*sum(y.*cos((n*pi*X)/l));
b(n)=(2/N)*sum(y.*sin((n*pi*X)/l));
end
syms x
for n=1:N
F=a0/2+sum((a(n)*cos(n*pi*x)/l))+sum((b(n)*sin(n*pi*x)/l));
end
disp(F)
scatter(X,y)
hold on
fplot(x,F,[0,6])
hold off
```

$$\frac{18815340698337411\cos(7\pi x)}{985162418487296\pi} - \frac{196385992008753\sin(7\pi x)}{8796093022208\pi} + \frac{134}{7}$$



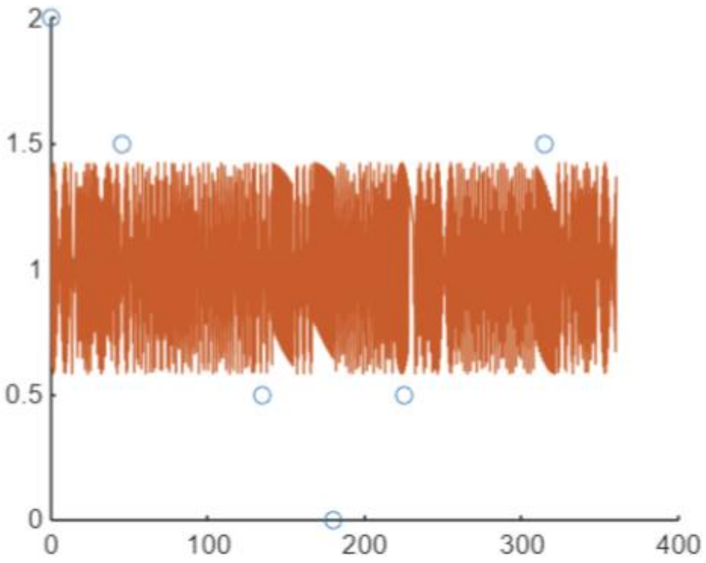
b)

x	0	45	90	135	180	225	270	315
y	2	1.5	1	0.5	0	0.5	1	1.5



```
clear
X=[0 45 90 135 180 225 270 315];
y=[2 1.5 1 0.5 0 0.5 1 1.5];
h=pi/3;
N=length(X);
l=(N*h)/2;
a0 = (2/N)*sum(y);
for n=1:N
a(n)=(2/N)*sum(y.*cos((n*pi*X)/l));
b(n)=(2/N)*sum(y.*sin((n*pi*X)/l));
end
syms x
for n=1:N
F=a0/2+sum((a(n)*cos(n*pi*x)/l))+sum((b(n)*sin(n*pi*x)/l));
end
disp(F)
scatter(X,y)
hold on
fplot(x,F,[0,360])
hold off
```

$$\frac{2591831838760815\cos(8\pi x)}{2251799813685248\pi}-\frac{1495494992712447\sin(8\pi x)}{2251799813685248\pi}+1$$



Additional questions by the course instructor (if any):

Total Maks Obtained in the activity (out of 10) :	**Signature of the faculty**	**Signature of the student**

Activity 3: Solving a difference equation using Z-transform.

Z-transform of a sequence u_n is: $Z_T(u_n) = \sum_{n=0}^{\infty} u_n z^{n}$.

`Example: Compute the Z-transform of n(1-(1/n)).`

```
clc
clear
syms n
f=n*(1-(1/n));
Z=ztrans(f) %ztrans() is the inbuilt function in maTlab to obtain the Z
transfom
```

```
Z =
```

$$\frac{z}{(z-1)^2} - \frac{z}{z-1}$$

`Example: Compute the inverse Z-transform of z/(z-1).`

```
clc clear
syms z
f=z/(z-
1);
un=iztrans(f) %iztrans() is the inbuilt function in maTlab to obtain the
inverse Z transfom
```

```
un = 1
```

Z transform can be used to solve difference equations(refer for info: Difreqns MathLibre).

Example: You can use the Z-transform to solve difference equations, such as the well-known "Rabbit Growth" problem. If a pair of rabbits matures in one year, and then produces another pair of rabbits every year, the rabbit population y_n at year n is described by this difference equation.

$y_{n+2} = y_{n+1} + y_n$ **with** $y_0 = 1, y_1 = 2.$

```
syms y(n) z
assume(n>=0 & in(n,"integer"))%Specifying that n should be an integer
greater than or equal to 0
f = y(n+2) - y(n+1) -
y(n);
fZT = ztrans(f,n,z);%Inbuilt function to find Z transform
syms yZT
fZT = subs(fZT,ztrans(y(n),n,z),yZT);
yZT = solve(fZT,yZT);
ySol = iztrans(yZT,z,n);%Inbuilt function to find inverse Z transform
ySol = simplify(ySol)
ySol =
```

$$2\,(-1)^{n/2}\,y(1)\cos\!\left(n\left(\frac{\pi}{2}+\mathrm{asin}\!\left(\frac{1}{2}\,\mathrm{i}\right)\right)\right)+\frac{2^{2-n}\,\sqrt{5}\,\sigma_1\,\left(\sqrt{5}+1\right)^{n-1}}{5}-\frac{2\,2^{1-n}\,\sqrt{5}\,\sigma_1\,\left(1-\sqrt{5}\right)^{n-1}}{5}$$

where

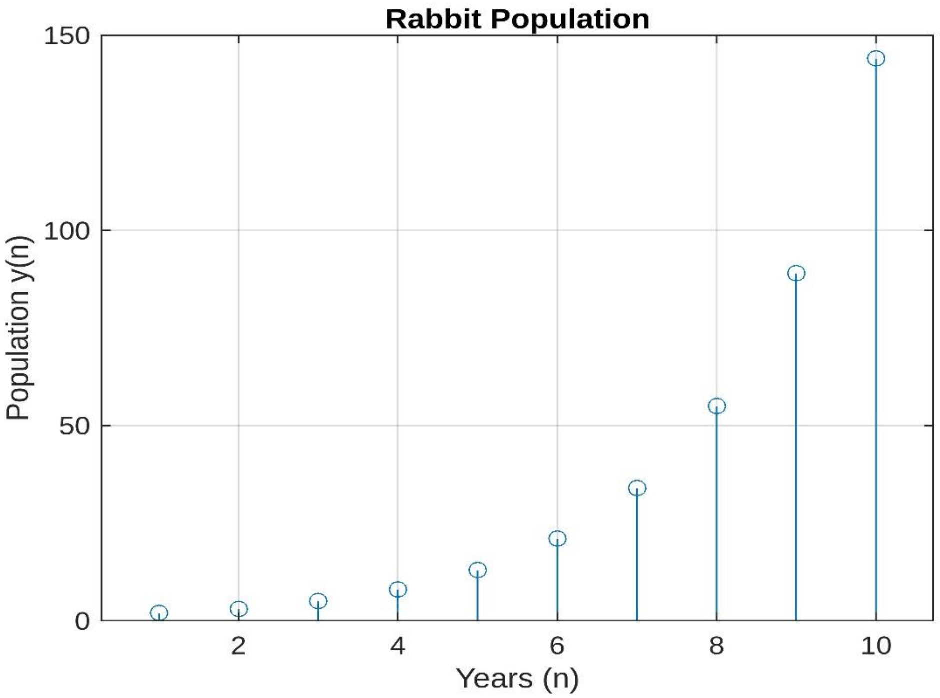
$$\sigma_1=\frac{y(0)}{2}-y(1)$$

```
ySol = subs(ySol,[y(0) y(1)],[1 2])
```

```
ySol =
```

$$4\,(-1)^{n/2}\cos\!\left(n\left(\frac{\pi}{2}+\mathrm{asin}\!\left(\frac{1}{2}\,\mathrm{i}\right)\right)\right)-\frac{3\,2^{2-n}\,\sqrt{5}\,\left(\sqrt{5}+1\right)^{n-1}}{10}+\frac{3\,2^{1-n}\,\sqrt{5}\,\left(1-\sqrt{5}\right)^{n-1}}{5}$$

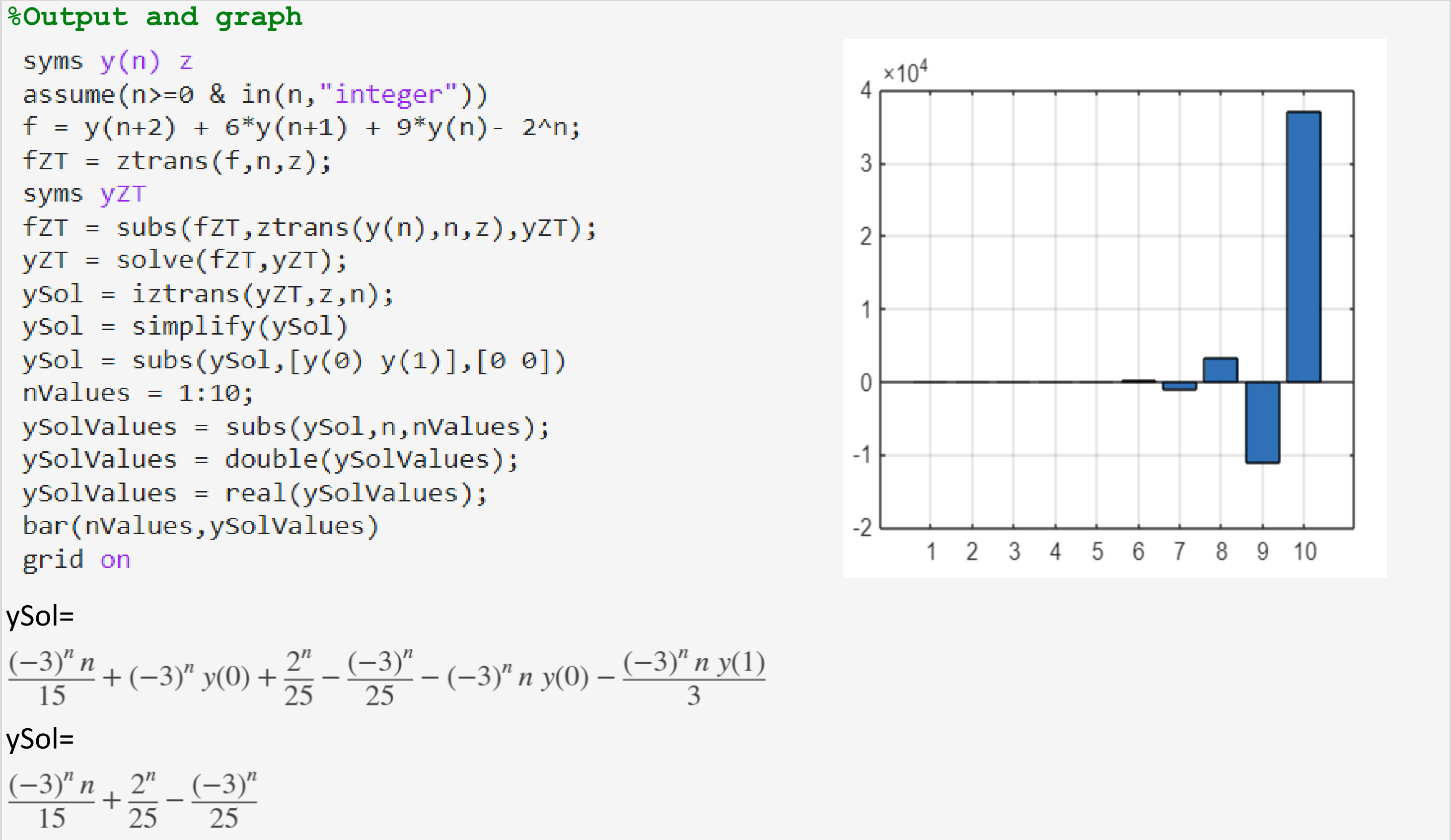
```
%Stem Plot of Rabbit Growth
nValues = 1:10;
ySolValues = subs(ySol,n,nValues);
ySolValues = double(ySolValues);
ySolValues = real(ySolValues);
stem(nValues,ySolValues)
title("Rabbit Population")
xlabel("Years (n)")
ylabel("Population y(n)")
grid on
```



EXERCISES:

Solve the following difference equations and get a suitable plot (may not be a stemplot) for the sequences obtained and paste the graphs.

1. $y_{n+2}+6y_{n+1}+9y_n=2^n$ **with** $y_0=0, y_1=0.$



```
%Output and graph
 syms y(n) z
 assume(n>=0 & in(n,"integer"))
 f = y(n+2) + 6*y(n+1) + 9*y(n)- 2^n;
 fZT = ztrans(f,n,z);
 syms yZT
 fZT = subs(fZT,ztrans(y(n),n,z),yZT);
 yZT = solve(fZT,yZT);
 ySol = iztrans(yZT,z,n);
 ySol = simplify(ySol)
 ySol = subs(ySol,[y(0) y(1)],[0 0])
 nValues = 1:10;
 ySolValues = subs(ySol,n,nValues);
 ySolValues = double(ySolValues);
 ySolValues = real(ySolValues);
 bar(nValues,ySolValues)
 grid on
```

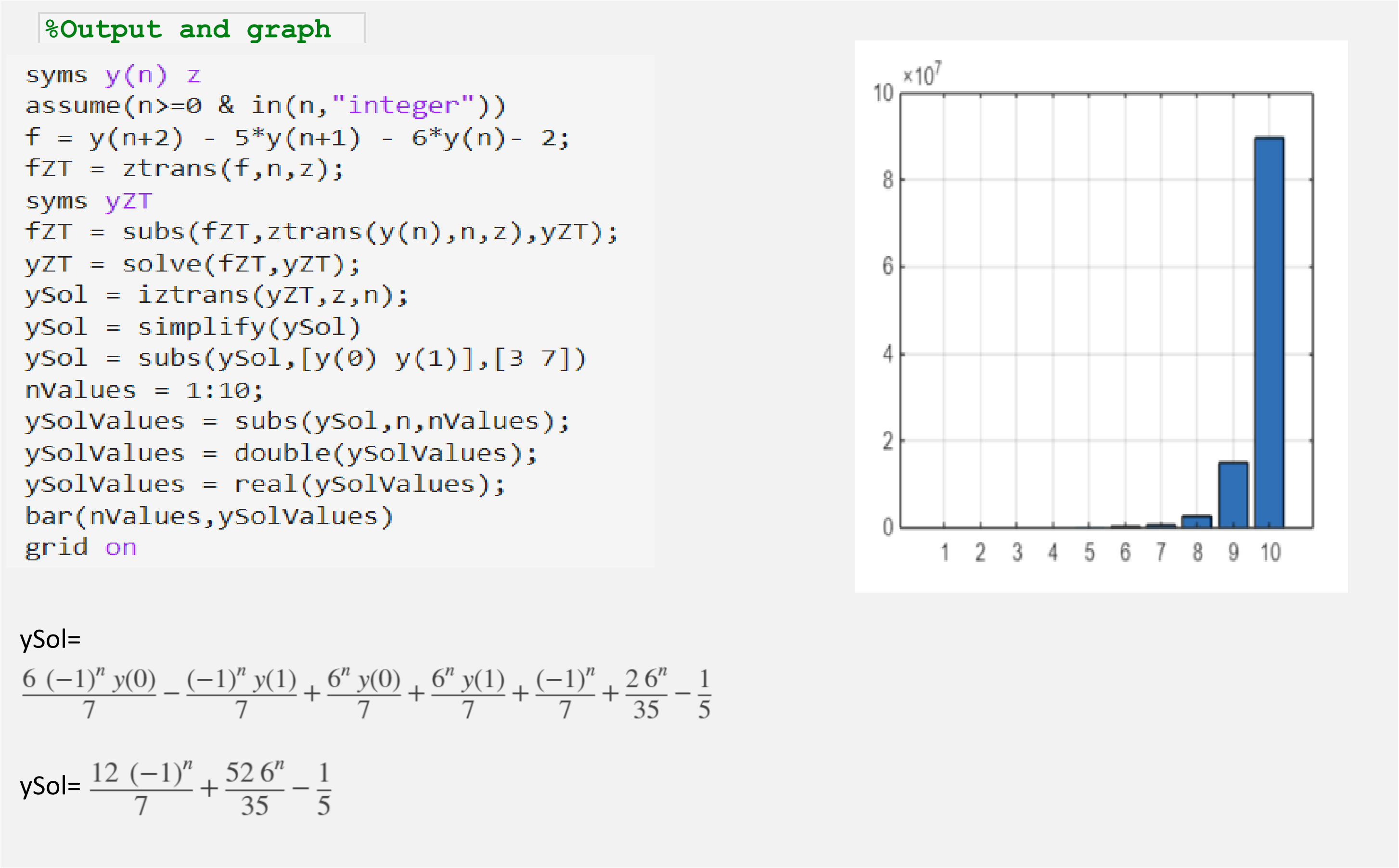
ySol=

$$\frac{(-3)^n\, n}{15}+(-3)^n\, y(0)+\frac{2^n}{25}-\frac{(-3)^n}{25}-(-3)^n\, n\, y(0)-\frac{(-3)^n\, n\, y(1)}{3}$$

ySol=

$$\frac{(-3)^n\, n}{15}+\frac{2^n}{25}-\frac{(-3)^n}{25}$$

2. $y_{n+2}=5y_{n+1}+6y_n=2$ **with** $y_0=3, y_1=7.$



```
%Output and graph
syms y(n) z
assume(n>=0 & in(n,"integer"))
f = y(n+2) - 5*y(n+1) - 6*y(n)- 2;
fZT = ztrans(f,n,z);
syms yZT
fZT = subs(fZT,ztrans(y(n),n,z),yZT);
yZT = solve(fZT,yZT);
ySol = iztrans(yZT,z,n);
ySol = simplify(ySol)
ySol = subs(ySol,[y(0) y(1)],[3 7])
nValues = 1:10;
ySolValues = subs(ySol,n,nValues);
ySolValues = double(ySolValues);
ySolValues = real(ySolValues);
bar(nValues,ySolValues)
grid on
```

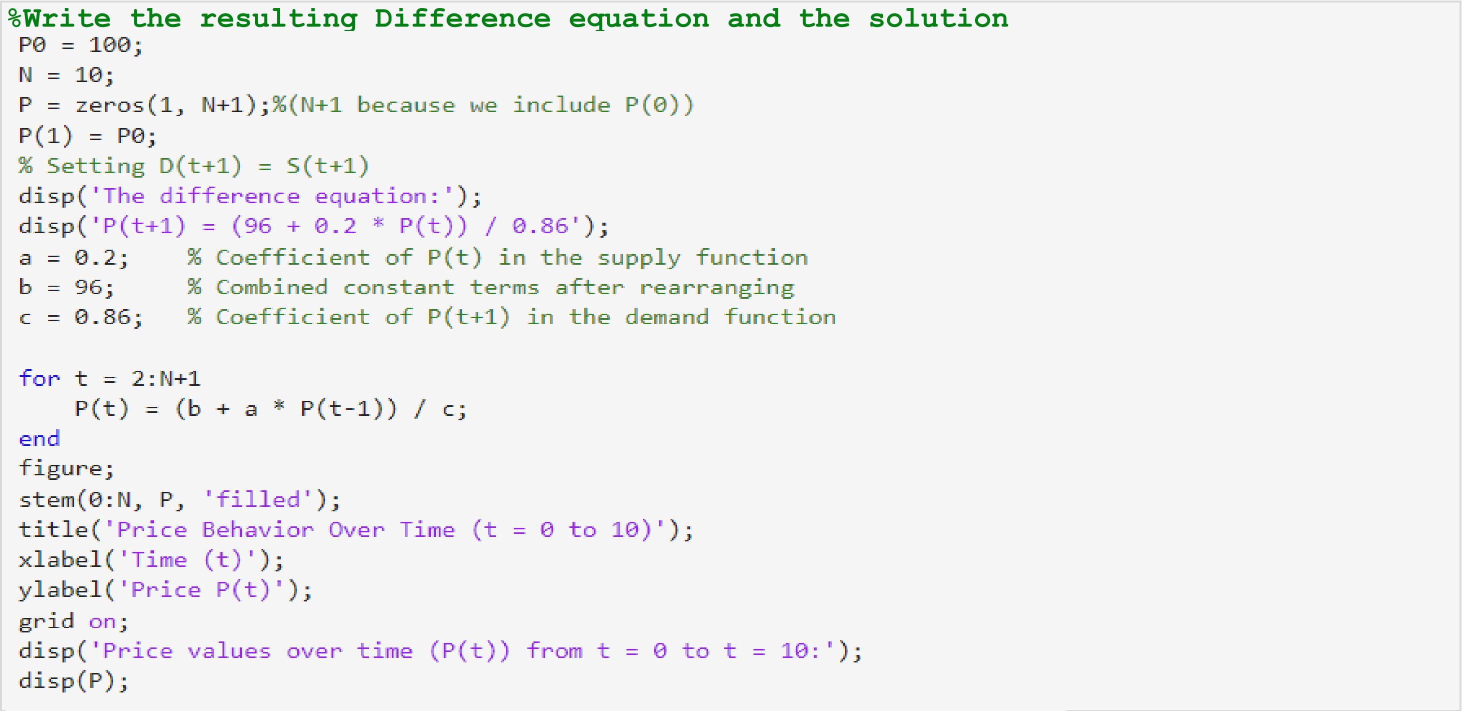
ySol=

$$\frac{6\,(-1)^n\, y(0)}{7}-\frac{(-1)^n\, y(1)}{7}+\frac{6^n\, y(0)}{7}+\frac{6^n\, y(1)}{7}+\frac{(-1)^n}{7}+\frac{2\, 6^n}{35}-\frac{1}{5}$$

ySol= $\frac{12\,(-1)^n}{7}+\frac{52\, 6^n}{35}-\frac{1}{5}$

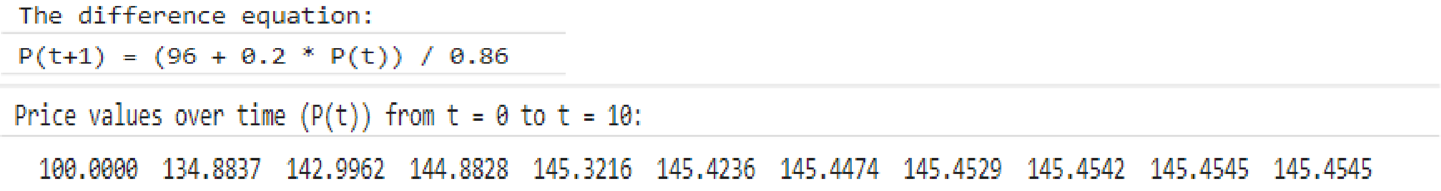
An economic application: **We want to investigate the behaviour of price in a market with the demand and supply functions as follows.**

$D_{t+1} = 86 - 0.8P_{t+1}$.**and** $S_{t+1} = -10 + 0.2P_t$. **Assuming that market is cleared** (ie, D_t=S_t)**. Form the difference equation and hence obtain the price at any point of time *'t',* ie,** P_t **if the price initially is** $P_0 = 100$.



```
%Write the resulting Difference equation and the solution
 P0 = 100;
 N = 10;
 P = zeros(1, N+1);%(N+1 because we include P(0))
 P(1) = P0;
 % Setting D(t+1) = S(t+1)
 disp('The difference equation:');
 disp('P(t+1) = (96 + 0.2 * P(t)) / 0.86');
 a = 0.2;    % Coefficient of P(t) in the supply function
 b = 96;     % Combined constant terms after rearranging
 c = 0.86;   % Coefficient of P(t+1) in the demand function

 for t = 2:N+1
     P(t) = (b + a * P(t-1)) / c;
 end
 figure;
 stem(0:N, P, 'filled');
 title('Price Behavior Over Time (t = 0 to 10)');
 xlabel('Time (t)');
 ylabel('Price P(t)');
 grid on;
 disp('Price values over time (P(t)) from t = 0 to t = 10:');
 disp(P);
```

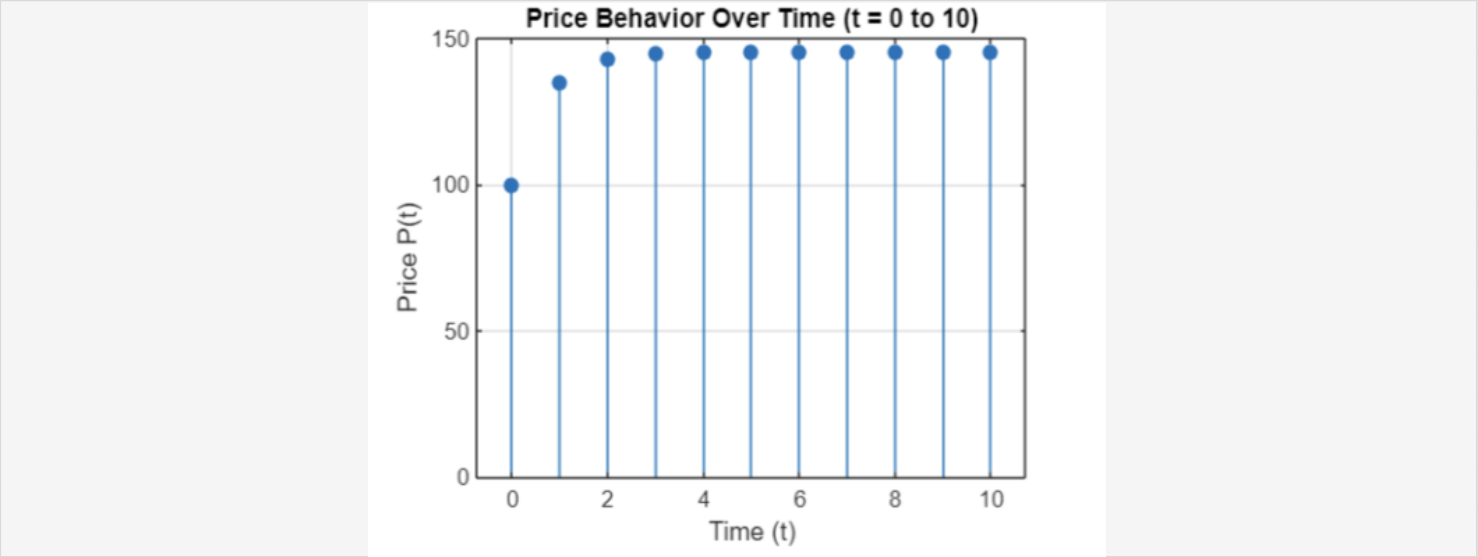


```
The difference equation:
P(t+1) = (96 + 0.2 * P(t)) / 0.86

Price values over time (P(t)) from t = 0 to t = 10:

  100.0000  134.8837  142.9962  144.8828  145.3216  145.4236  145.4474  145.4529  145.4542  145.4545  145.4545
```

Obtain the stemplot for time t=0 to 10 as explained in the first example. Give suitable name for the graph. Paste the graph.

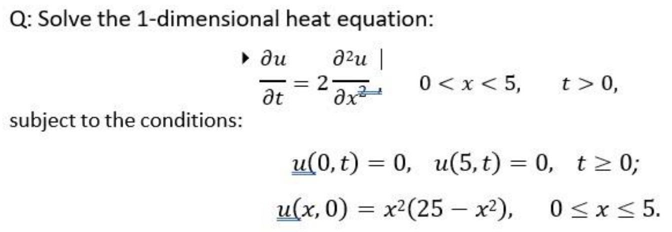


Total Maks Obtained in the activity (out of 10) :	Signature of the faculty	Signature of the student

Activity 4: Numerical Solution of one dimensional heat equation.

One dimensional heat equation is given by $u_t = c^2 u_{xx}$, with initial condition on time and boundary conditions on the space variable *x.* Bender-Schmidt method uses the following formula to find the approximate values of *u* at the *j+1* th time level using the value at *j*th time level.

$u_{i,j+1} = \lambda u_{i-1,j} + (1-2\lambda)u_{i,j} + \lambda\lambda u_{i+1,j}$. **where** $\lambda = \frac{kc^2}{h^2}$ **, k and h are the step lengths for t and respectively.**



Q: Solve the 1-dimensional heat equation:

$$\frac{\partial u}{\partial t} = 2\frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < 5, \qquad t > 0,$$

subject to the conditions:

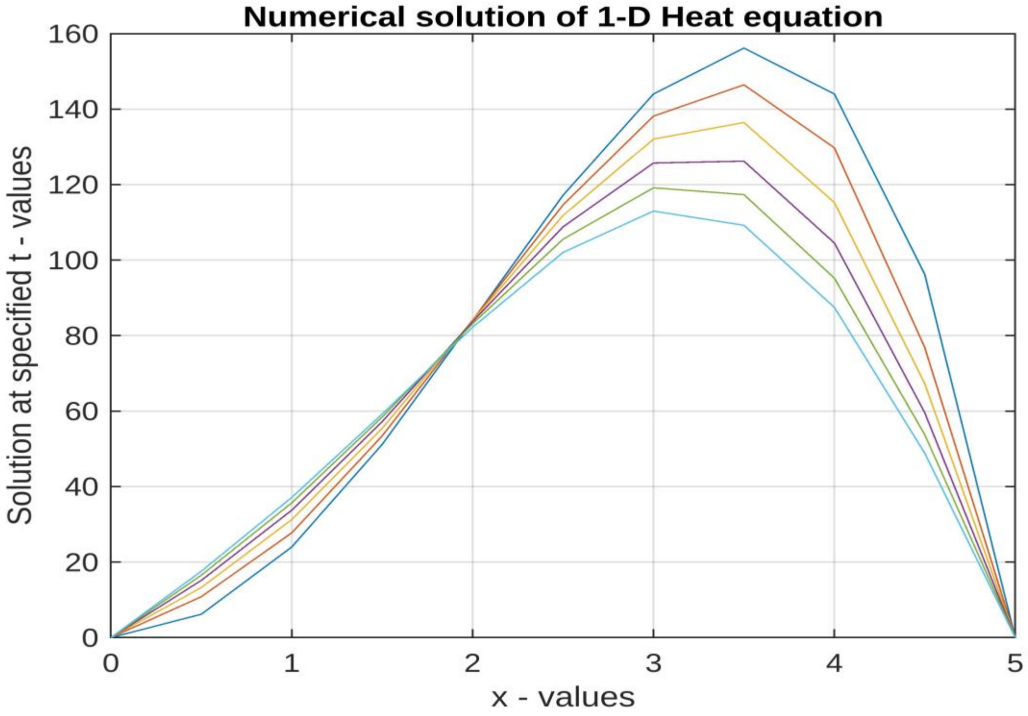
$$u(0,t) = 0, \quad u(5,t) = 0, \quad t \geq 0;$$

$$u(x,0) = x^2(25 - x^2), \qquad 0 \leq x \leq 5.$$

```
x0=0; xm = 5; tn = 0.25; h = 0.5; k=0.05; c2=2;
f = @(x) x.^2.*(25-x.^2);
[x,t,u] = FT_Heat(x0,xm,tn,h,k,c2,f); fprintf('Numerical solution of
the given equation is : \n'); disp(u);
```

```
Numerical solution of the given equation is :
         0         0         0         0         0         0
    6.1875   10.8375   13.2675   15.1575   16.5267   17.5844
24.0000    27.7500   31.2600   33.7380   35.6976   37.1225
51.1875    53.4375   55.4475   57.2175   58.4307   59.2292
84.0000    84.1500   84.0600   83.7300   83.1600   82.2233
117.1875  114.6375  111.8475  108.8175  105.5475  102.0375
  144.0000  138.1500  132.0600  125.7300  119.1600  112.9913
  156.1875  146.4375  136.4475  126.2175  117.3507  109.2212
  144.0000  129.7500  115.2600  104.5380   95.2176   87.4745
   96.1875   76.8375   67.2675   59.5575   53.7267   48.8324
         0         0         0         0         0         0
```

```
plot(x,u); grid on;
xlabel('x - values');
ylabel('Solution at specified t - values'); title('Numerical solution of 1-D
Heat equation');
```

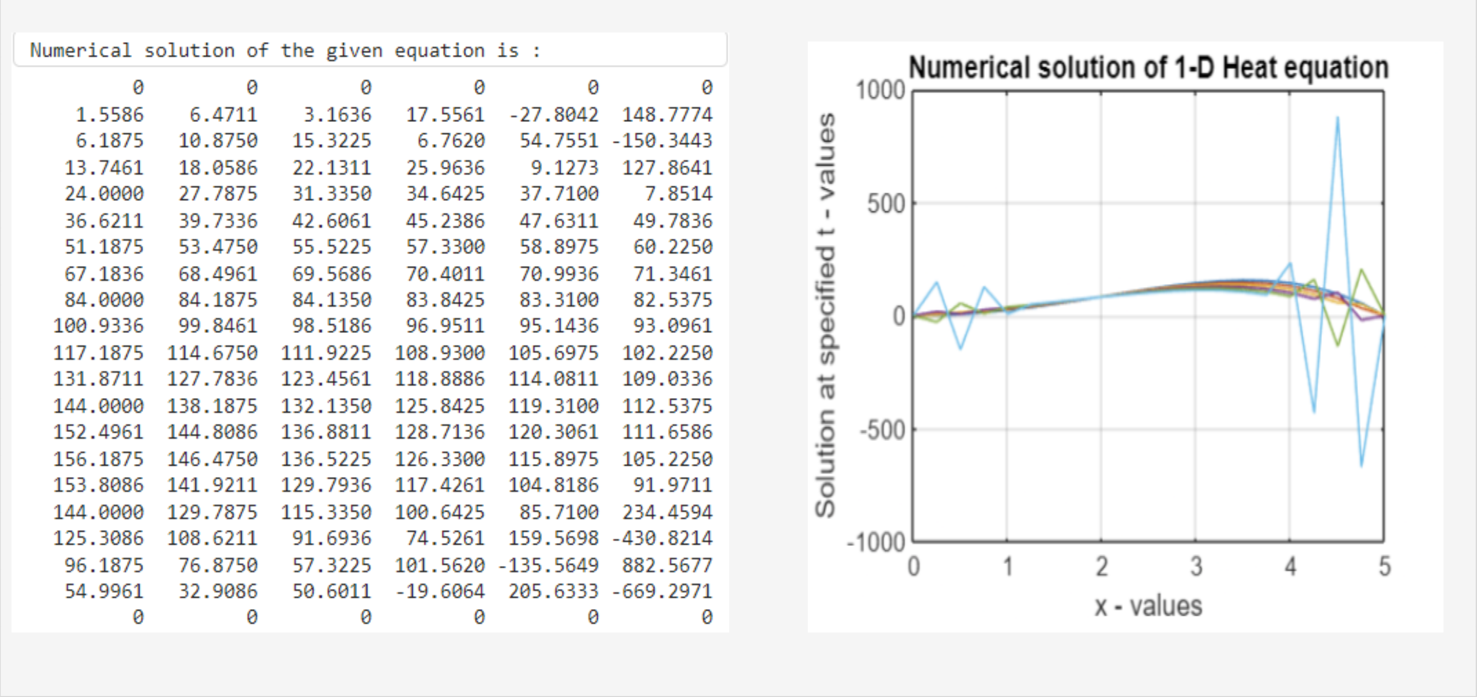


```
%Defining the function FTCS_Heat
function [x,t,u] = FT_Heat(x0,xm,tn,h,k,c2,f)
lambda=c2*k/h^2;
x=x0:h:xm; n=length(x); t=0:k:tn; m=length(t); u=zeros(n,m); u(:,1)=f(x);
for j=1:m-1
for i=2:n-1
u(i,j+1)= lambda*u(i-1,j)+(1-2*lambda)*u(i,j)+lambda*u(i+1,j);
end
end
end
```

EXERCISE:

What is the effect of decreasing the values of *h* and *k* further on the accuracy of the solution? Observe it by taking h=0.25, k=0.05 for the same PDE above. Obtain the output and paste the graph and write your conclusion.

Decreasing the values of h (spatial step) and k (time step) in the numerical solution of the 1-D heat equation improves accuracy by providing finer resolution in both space and time. With a smaller h, more spatial points are considered, capturing detailed changes in the heat distribution, while a smaller k allows better tracking of the time evolution of the solution, reducing numerical errors. However, this comes at the cost of increased computational effort, as the number of calculations grows significantly. The improved resolution generally results in a smoother and more accurate approximation of the heat equation's behavior.



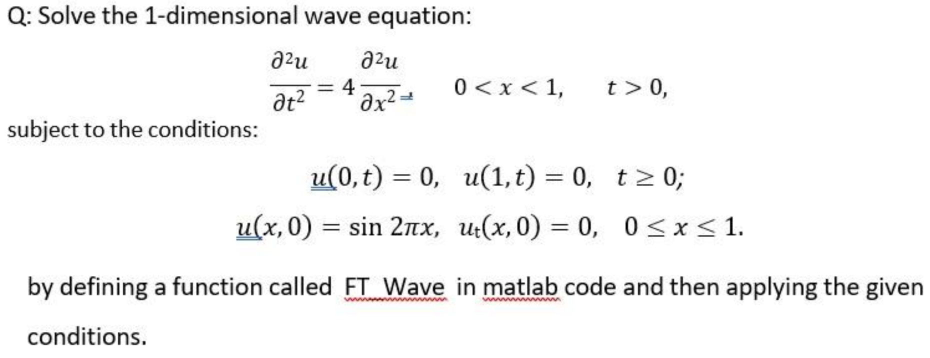
```
Numerical solution of the given equation is :
         0         0         0         0         0         0
    1.5586    6.4711    3.1636   17.5561  -27.8042  148.7774
    6.1875   10.8750   15.3225    6.7620   54.7551 -150.3443
   13.7461   18.0586   22.1311   25.9636    9.1273  127.8641
   24.0000   27.7875   31.3350   34.6425   37.7100    7.8514
   36.6211   39.7336   42.6061   45.2386   47.6311   49.7836
   51.1875   53.4750   55.5225   57.3300   58.8975   60.2250
   67.1836   68.4961   69.5686   70.4011   70.9936   71.3461
   84.0000   84.1875   84.1350   83.8425   83.3100   82.5375
  100.9336   99.8461   98.5186   96.9511   95.1436   93.0961
  117.1875  114.6750  111.9225  108.9300  105.6975  102.2250
  131.8711  127.7836  123.4561  118.8886  114.0811  109.0336
  144.0000  138.1875  132.1350  125.8425  119.3100  112.5375
  152.4961  144.8086  136.8811  128.7136  120.3061  111.6586
  156.1875  146.4750  136.5225  126.3300  115.8975  105.2250
  153.8086  141.9211  129.7936  117.4261  104.8186   91.9711
  144.0000  129.7875  115.3350  100.6425   85.7100  234.4594
  125.3086  108.6211   91.6936   74.5261  159.5698 -430.8214
   96.1875   76.8750   57.3225  101.5620 -135.5649  882.5677
   54.9961   32.9086   50.6011  -19.6064  205.6333 -669.2971
         0         0         0         0         0         0
```

Numerical solution of 1-D Heat equation
Solution at specified t - values
x - values
1000
500
0
-500
-1000
0
1
2
3
4
5

Total Maks Obtained in the activity (out of 10) :	**Signature of the faculty**	**Signature of the student**

Activity 5: Numerical Solution of one dimensional wave equation.

One dimensional wave equation is given by $u_{tt} = c^2 u_{xx}$, with initial condition on time and boundary conditions on the space variable *x.* Bender-Schmidt method uses the following formula to find the approximate values of *u* at the *j+1* th time level using the value at *j*th time level. Let us take an example.



Q: Solve the 1-dimensional wave equation:

$$\frac{\partial^2 u}{\partial t^2} = 4\frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < 1, \qquad t > 0,$$

subject to the conditions:

$$u(0,t) = 0, \quad u(1,t) = 0, \quad t \geq 0;$$

$$u(x,0) = \sin 2\pi x, \quad u_t(x,0) = 0, \quad 0 \leq x \leq 1.$$

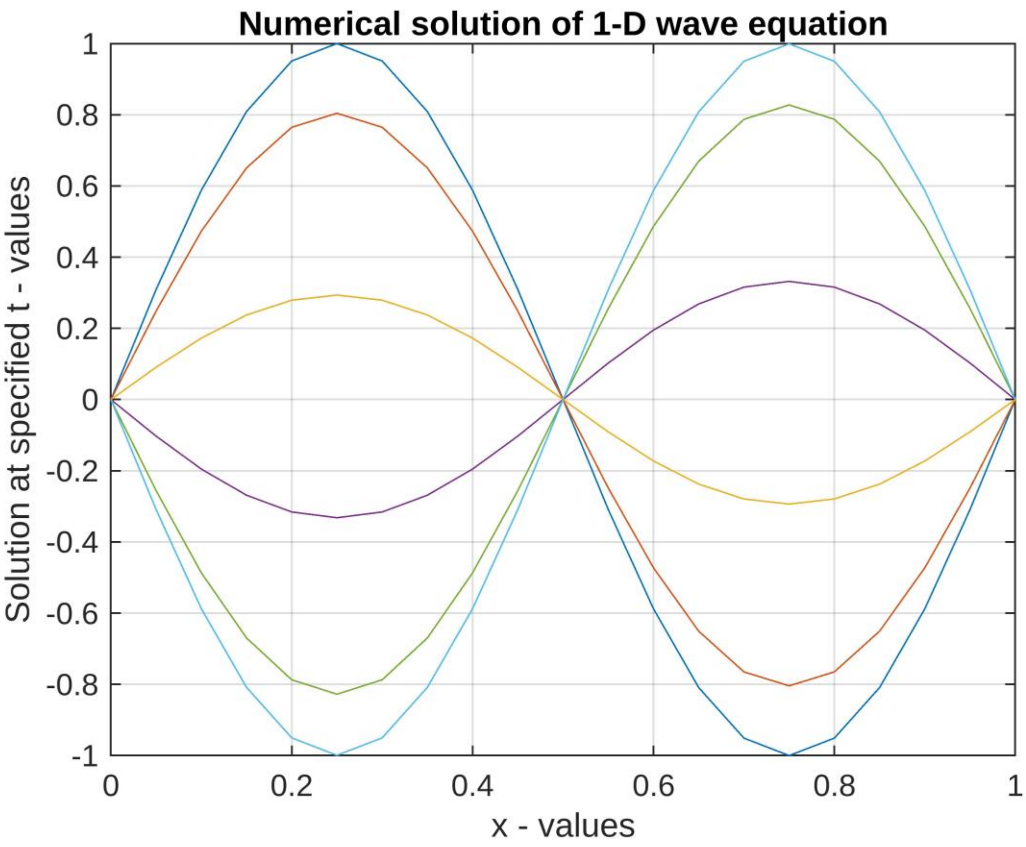
by defining a function called FT_Wave in matlab code and then applying the given conditions.

Code:

```
x0=0; xn=1; t0=0; tm=0.5; h=0.05; k=0.1; c=1;
f= @(x) sin(2*pi*x); g = @(x) 0;
[x,t,u] = FT_wave(t0,tm,x0,xn,h,k,c,f,g); fprintf('Numerical solution
of the given equation is : \n'); disp(u);
```

```
Numerical solution of the given equation is :
0          0          0          0          0          0
    0.3090     0.2485     0.0907    -0.1026    -0.2558    -0.3088
    0.5878     0.4727     0.1725    -0.1952    -0.4865    -0.5873
    0.8090     0.6506     0.2375    -0.2686    -0.6696    -0.8084
0.9511      0.7649      0.2792     -0.3158     -0.7871     -0.9503
1.0000      0.8042      0.2936     -0.3321     -0.8276     -0.9992
0.9511      0.7649      0.2792     -0.3158     -0.7871     -0.9503
0.8090      0.6506      0.2375     -0.2686     -0.6696     -0.8084
0.5878      0.4727      0.1725     -0.1952     -0.4865     -0.5873
0.3090      0.2485      0.0907     -0.1026     -0.2558     -0.3088
0.0000    -0.0000     0.0000    -0.0000     0.0000    -0.0000     -
0.3090    -0.2485    -0.0907     0.1026     0.2558     0.3088
   -0.5878    -0.4727    -0.1725     0.1952     0.4865     0.5873
   -0.8090    -0.6506    -0.2375     0.2686     0.6696     0.8084
   -0.9511    -0.7649    -0.2792     0.3158     0.7871     0.9503
   -1.0000    -0.8042    -0.2936     0.3321     0.8276     0.9992
   -0.9511    -0.7649    -0.2792     0.3158     0.7871     0.9503
   -0.8090    -0.6506    -0.2375     0.2686     0.6696     0.8084
   -0.5878    -0.4727    -0.1725     0.1952     0.4865     0.5873
   -0.3090    -0.2485    -0.0907     0.1026     0.2558     0.3088
   -0.0000          0          0          0          0          0
```

```
plot(x,u); grid on;
xlabel('x - values');
ylabel('Solution at specified t - values'); title('Numerical solution of 1-D
wave equation');
```

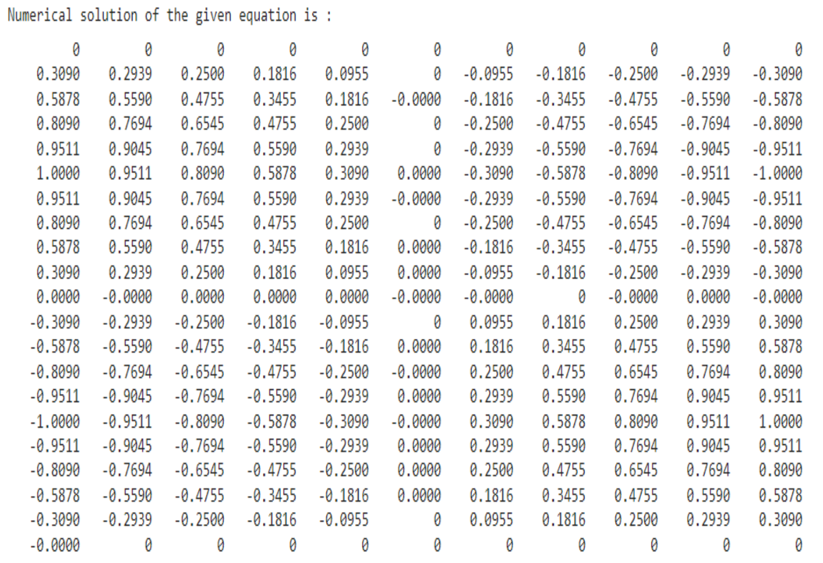


```
%Defining the function FT_Wave
function [x,t,u] = FT_wave(t0,tm,x0,xn,h,k,c,f,g)
r = c*k/h;
x=x0:h:xn; t=t0:k:tm; n=length(x); m=length(t); u=zeros(n,m); u(:,1)=f(x);
for i=2:n-1
u(i,2)=(1-r^2)*u(i,1)+0.5*(r^2*(u(i-1,1)+u(i+1,1)))+k*g(x(i));
end
for j=2:m-
1 for
i=2:n-1
u(i,j+1)=2*(1-r^2)*u(i,j)+r^2*(u(i-1,j)+u(i+1,j))-u(i,j-1);
end
end
end
```

EXERCISE:

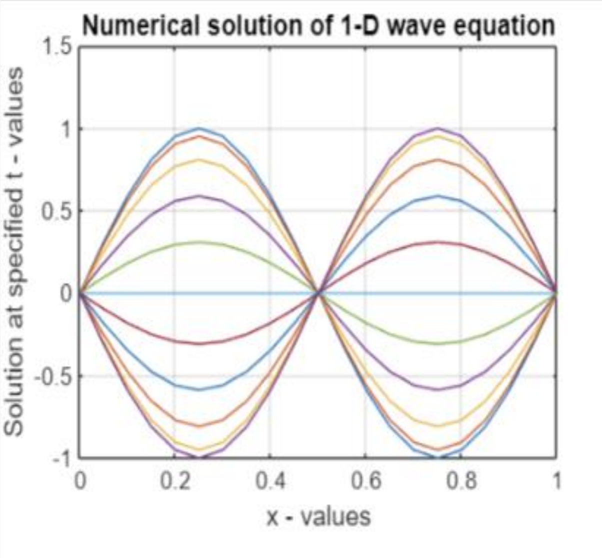
What is the effect of decreasing the values of *h* and *k* further on the accuracy of the solution? Observe it by taking h=0.05, k=0.05 for the same PDE above. Obtain the output and paste the graph and write your conclusion.

Decreasing h and k in the numerical solution of the 1-D wave equation increases the solution's accuracy by providing finer spatial and temporal resolution. This allows the wave's behavior to be captured in more detail, reducing errors in the approximation. However, it also increases the computational cost. When using h = 0.05 and k = 0.05, the wave solution becomes smoother and more precise, especially in regions of rapid change. This leads to better accuracy at the expense of more calculations.



```
Numerical solution of the given equation is :

         0         0         0         0         0         0         0         0         0         0         0
    0.3090    0.2939    0.2500    0.1816    0.0955         0   -0.0955   -0.1816   -0.2500   -0.2939   -0.3090
    0.5878    0.5590    0.4755    0.3455    0.1816   -0.0000   -0.1816   -0.3455   -0.4755   -0.5590   -0.5878
    0.8090    0.7694    0.6545    0.4755    0.2500         0   -0.2500   -0.4755   -0.6545   -0.7694   -0.8090
    0.9511    0.9045    0.7694    0.5590    0.2939         0   -0.2939   -0.5590   -0.7694   -0.9045   -0.9511
    1.0000    0.9511    0.8090    0.5878    0.3090    0.0000   -0.3090   -0.5878   -0.8090   -0.9511   -1.0000
    0.9511    0.9045    0.7694    0.5590    0.2939   -0.0000   -0.2939   -0.5590   -0.7694   -0.9045   -0.9511
    0.8090    0.7694    0.6545    0.4755    0.2500         0   -0.2500   -0.4755   -0.6545   -0.7694   -0.8090
    0.5878    0.5590    0.4755    0.3455    0.1816    0.0000   -0.1816   -0.3455   -0.4755   -0.5590   -0.5878
    0.3090    0.2939    0.2500    0.1816    0.0955    0.0000   -0.0955   -0.1816   -0.2500   -0.2939   -0.3090
    0.0000   -0.0000    0.0000    0.0000    0.0000   -0.0000   -0.0000         0   -0.0000    0.0000   -0.0000
   -0.3090   -0.2939   -0.2500   -0.1816   -0.0955         0    0.0955    0.1816    0.2500    0.2939    0.3090
   -0.5878   -0.5590   -0.4755   -0.3455   -0.1816    0.0000    0.1816    0.3455    0.4755    0.5590    0.5878
   -0.8090   -0.7694   -0.6545   -0.4755   -0.2500   -0.0000    0.2500    0.4755    0.6545    0.7694    0.8090
   -0.9511   -0.9045   -0.7694   -0.5590   -0.2939    0.0000    0.2939    0.5590    0.7694    0.9045    0.9511
   -1.0000   -0.9511   -0.8090   -0.5878   -0.3090   -0.0000    0.3090    0.5878    0.8090    0.9511    1.0000
   -0.9511   -0.9045   -0.7694   -0.5590   -0.2939    0.0000    0.2939    0.5590    0.7694    0.9045    0.9511
   -0.8090   -0.7694   -0.6545   -0.4755   -0.2500    0.0000    0.2500    0.4755    0.6545    0.7694    0.8090
   -0.5878   -0.5590   -0.4755   -0.3455   -0.1816    0.0000    0.1816    0.3455    0.4755    0.5590    0.5878
   -0.3090   -0.2939   -0.2500   -0.1816   -0.0955         0    0.0955    0.1816    0.2500    0.2939    0.3090
   -0.0000         0         0         0         0         0         0         0         0         0         0
```



Total Maks Obtained in the activity (out of 10) :	**Signature of the faculty**	**Signature of the student**

Final Assessment

Total Maks Obtained in the activity (out of 10) :	Signature of the faculty	Signature of the student

Final Total Maks Obtained in the ABA (out of 10) :	Signature of the faculty	Signature of the HOD