به نام حضرت دوست

دانشکده مهندسی برق

درس ریاضی مهندسی

گزارش کار **matlab** ۳

نام و نام خانوادگی:

طاها انتصاری ۹۵۱۰۱۱۱۷

وحید احمدی ۹۵۱۰۹۰۸۳

قسمت اول :آشنایی با تبدیل فوریه

۱. برای هر یک از توابع زیر ابتدا با استفاده از fourier تبدیل فوریه آن تابع را بیابید و آن را رسم کنید.

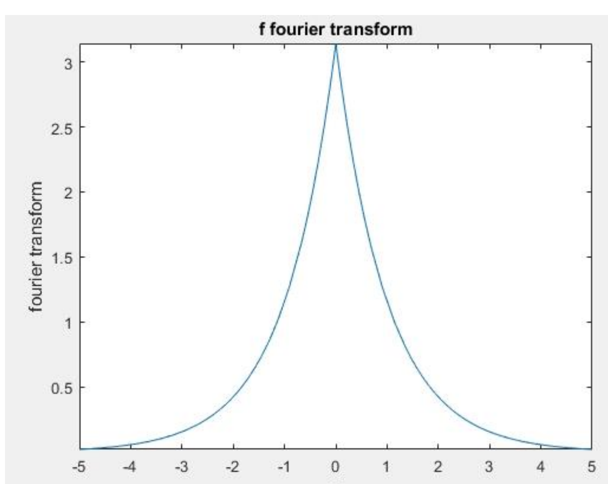
ابتدا با استفاده از توابع داده شده، ۴ تابع داده شده را تعریف می کنیم:

```
clear
syms x w;
f=@(x)(1/(1+x^2));
g=@(x)(exp(-abs(x)));
h=@(x)(x*exp(-x)*heaviside(x));
l=@(x)(x*exp(x)*heaviside(-x));
fft=eval(['@(w)' char(fourier(f(x)))]);
gft=eval(['@(w)' char(fourier(g(x)))]);
hft=eval(['@(w)' char(fourier(h(x)))]);
lft=eval(['@(w)' char(fourier(l(x)))]);
%fft=@(w)(fourier(f,x,w));
%gft=@(w)(fourier(g,x,w));
%hft=@(w)(fourier(h,x,w));
%lft=@(w)(fourier(l,x,w));

fprintf('f(x) fourier transform:\n');
pretty(fft(w));
fprintf('g(x) fourier transform:\n');
pretty(gft(w));
fprintf('h(x) fourier transform:\n');
pretty(hft(w));
fprintf('l(x) fourier transform:\n');
pretty(lft(w));
```

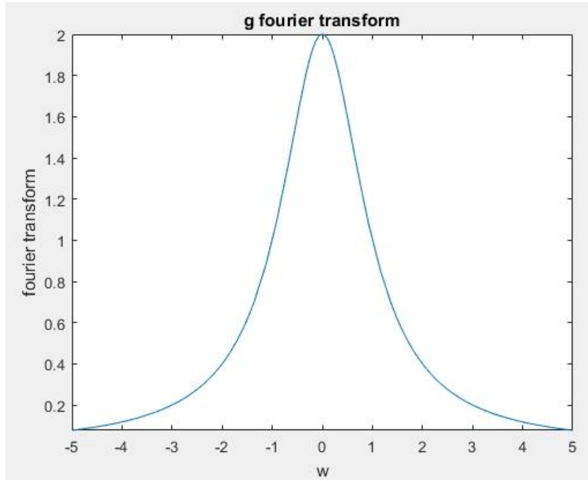
سپس کد دستوری را برای هر قسمت رسم می کنیم. به عنوان مثال برای اولی:

```
figure(1);
fplot(abs(fft(w)));
title('f fourier transform');
xlabel('w');
ylabel('fourier transform');
```

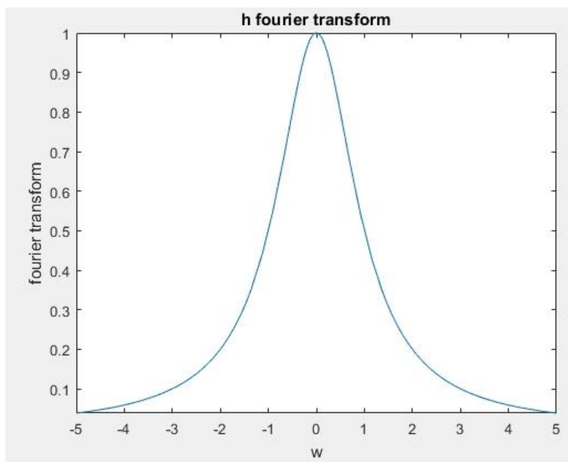
الف)



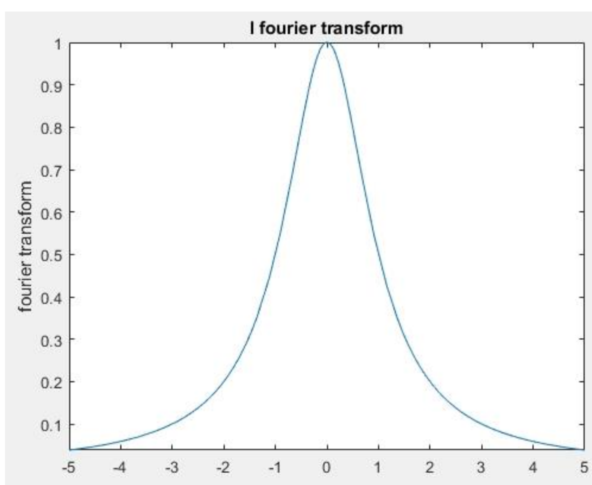
ب)


g fourier transform
fourier transform
w
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
-5
-4
-3
-2
-1
0
1
2
3
4
5

ج)


h fourier transform
fourier transform
w
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
-5
-4
-3
-2
-1
0
1
2
3
4
5

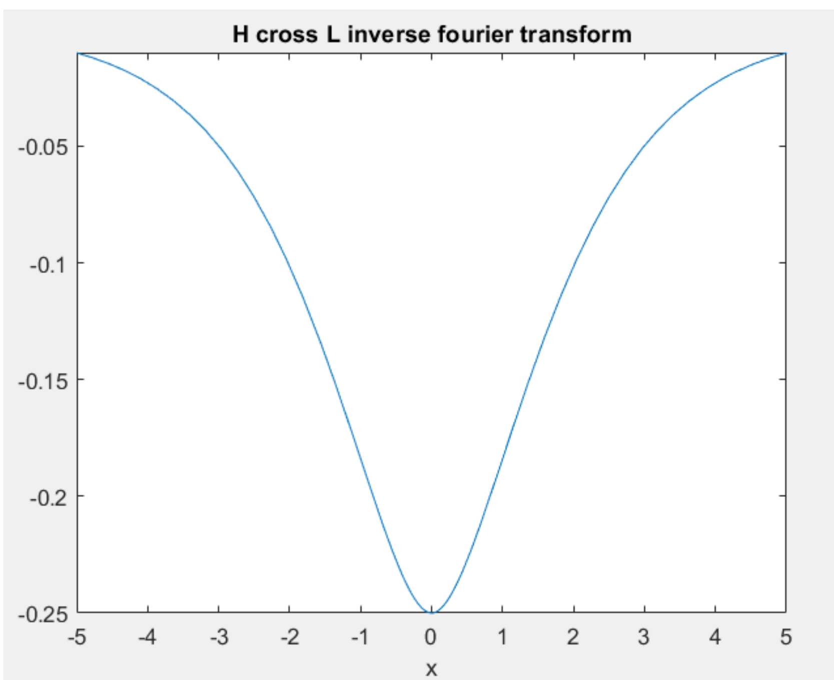
د)


l fourier transform
fourier transform
w
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
-5
-4
-3
-2
-1
0
1
2
3
4
5

۲. بااستفاده ازعباراتی که برای تبدیل فوریه هر یک از توابع بالابه دست آوردید. توابع گفته شده را محاسبه می کنیم سپس وارون تبدیل فوریه آن ها را رسم می کنیم.

با استفاده از کد دستوری زیر، توابع خواسته شده را رسم می کنیم:

```
FcrossG=eval(['@(w)' char(fft(w)*gft(w))]);
HcrossL=eval(['@(w)' char(hft(w)*lft(w))]);
fprintf('F(w)*G(w) :\n');
pretty(FcrossG(w));
fprintf('H(w)*L(w) :\n');
pretty(HcrossL(w));
FcrossGinvf=eval(['@(x)' char((ifourier(FcrossG(w))))]);
HcrossLinvf=eval(['@(x)' char((ifourier(HcrossL(w))))]);

fprintf('F cross G inverse fourier transform:\n');
pretty(FcrossGinvf(x));
fprintf('According to matlab''s documentation it is said
fprintf('H cross L inverse fourier transform:\n');
pretty(HcrossLinvf(x));
figure(5)
fplot(FcrossGinvf(x));
title('F cross G inverse fourier transform');
xlabel('x');
figure(6)
fplot(HcrossLinvf(x));
title('H cross L inverse fourier transform');
xlabel('x');
```

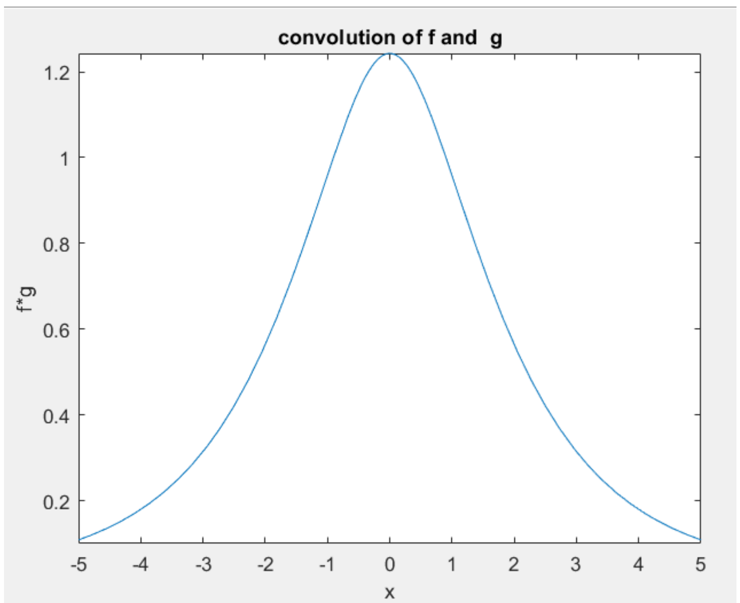


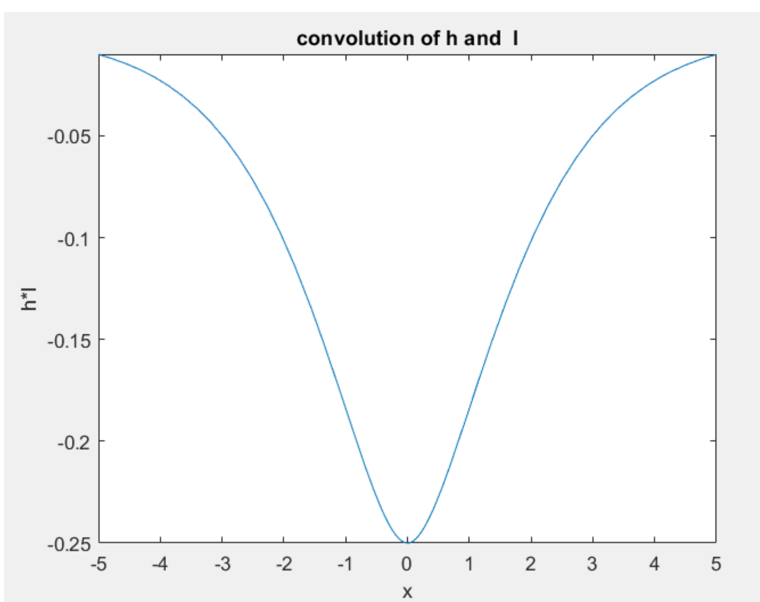
شکل برای تابع $F(j\omega) * G(j\omega)$، run می شود اما نمایش داده نمی شود.

۳. حال توابع این بخش را محاسبه می کنیم و آن ها را رسم می کنیم.

با استفاده از کد دستوری زیر، توابع خواسته شده را رسم می کنیم:

```
syms y
fconvg=eval(['@(x)' char(int(f(y)*g(x-y),y,-inf,inf))]);
fprintf('f convolve g is :\n');
pretty(fconvg(x))
hconvl=eval(['@(x)' char(int(h(y)*l(x-y),y,-inf,inf))]);
fprintf('h convolve l is :\n');
pretty(hconvl(x));
figure(8)
fplot(fconvg(x));
title('convolution of f and  g');
xlabel('x');
ylabel('f*g');
figure(9)
fplot(hconvl(x));
title('convolution of h and  l');
xlabel('x');
ylabel('h*l');
```





۴. نتایج دو قسمت قبل را با هم مقایسه می کنیم.

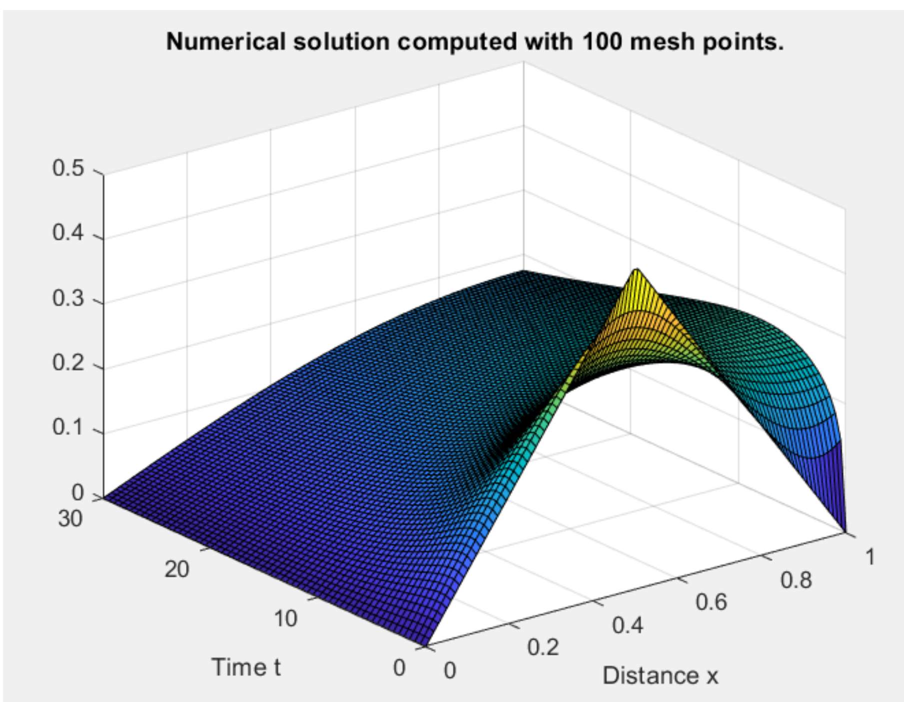
همانطور که از خواص تبدیل فوریه انتظار داریم، نتایج دو قسمت قبل یکسان است.

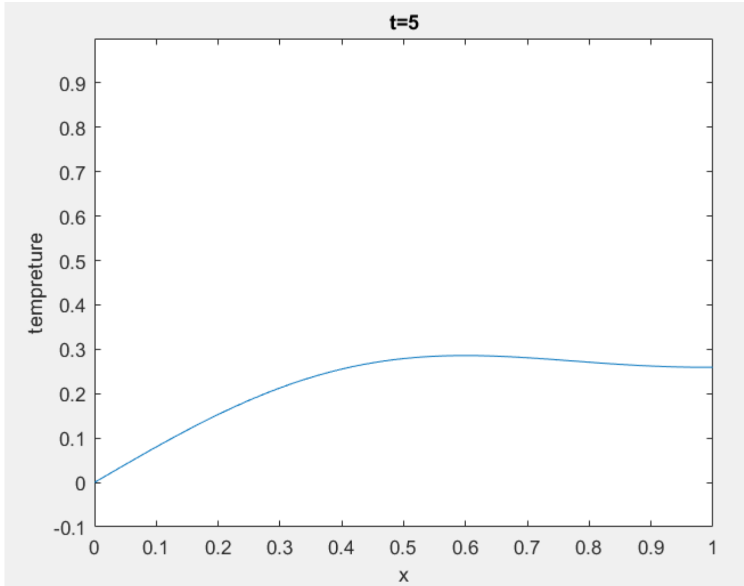
قسمت دوم: حل PDE در یک بعد (معادله حرارت)

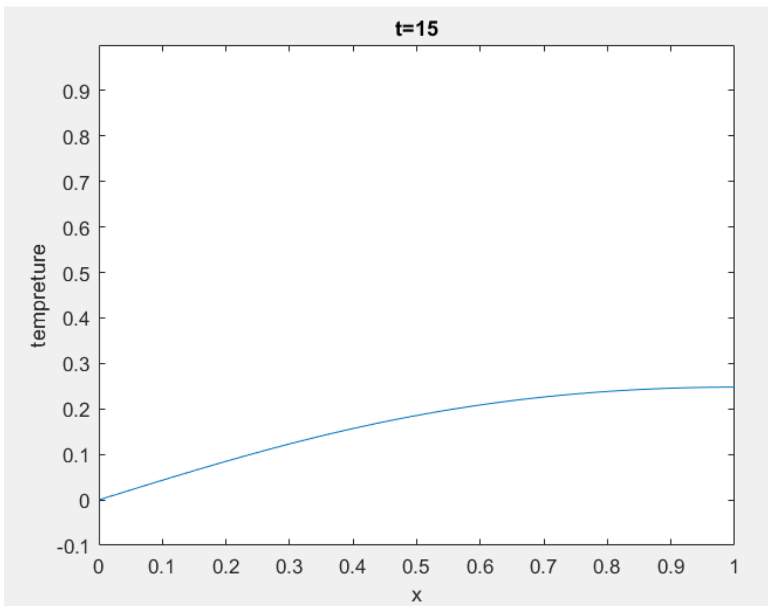
معادلات زیر معادله حرارت در یک میله به طول یک متر می باشند هر یک از معادلات حرارت زیر را با شرایط گفته شده با استفاده از تابع pdepe متلب حل کنید و نمودار تابع دمای میله را در لحظات گفته شده رسم کنید.

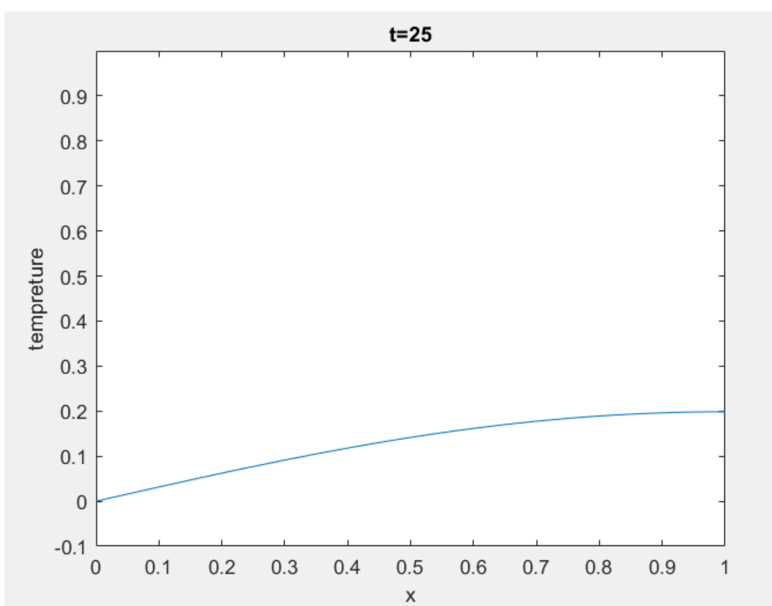
الف) کد دستوری زیر را اجرا می کنیم. برای ثانیه های ۱۵ و ۲۰ هم همانند کد زیر اجرا می کنیم:

```
x=linspace(0,1,100);
t=0:.5:30;
%t=linspace(0,30,100)
%t=[5,15,25];
u=pdepe(0,@equation,@initialcondition,@boundarycondition,x,t);
figure(10);
surf(x,t,u)
title('Numerical solution computed with 100 mesh points.')
xlabel('Distance x')
ylabel('Time t')

figure(11)
plot(x,u(find(t==5),:))
title('t=5');
xlabel('x');
ylabel('tempreture');
axis([0 1 -0.1 1])
```



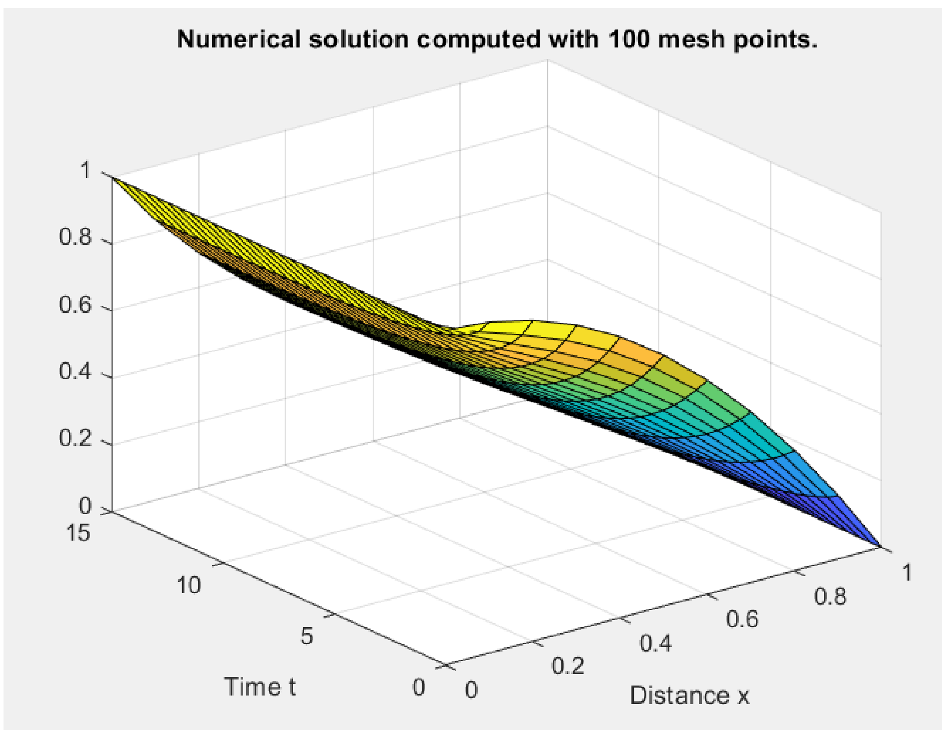

t=5
tempreture
x

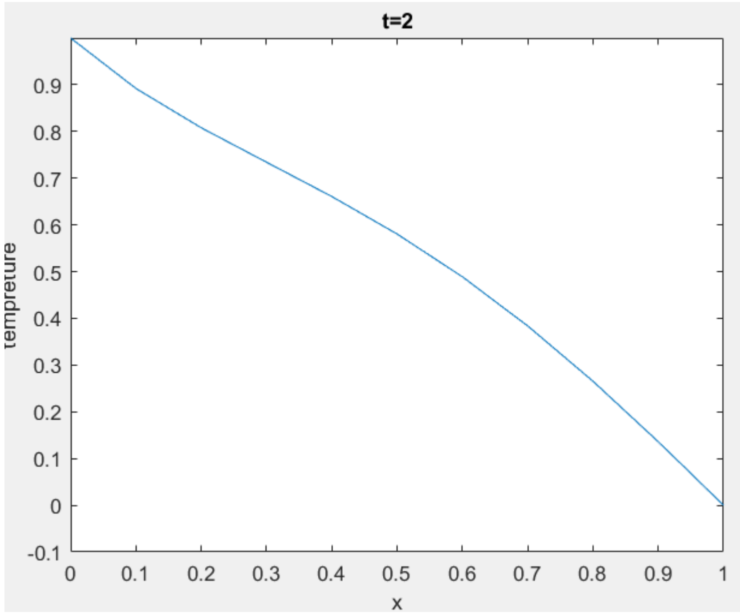

t=15
tempreture
x

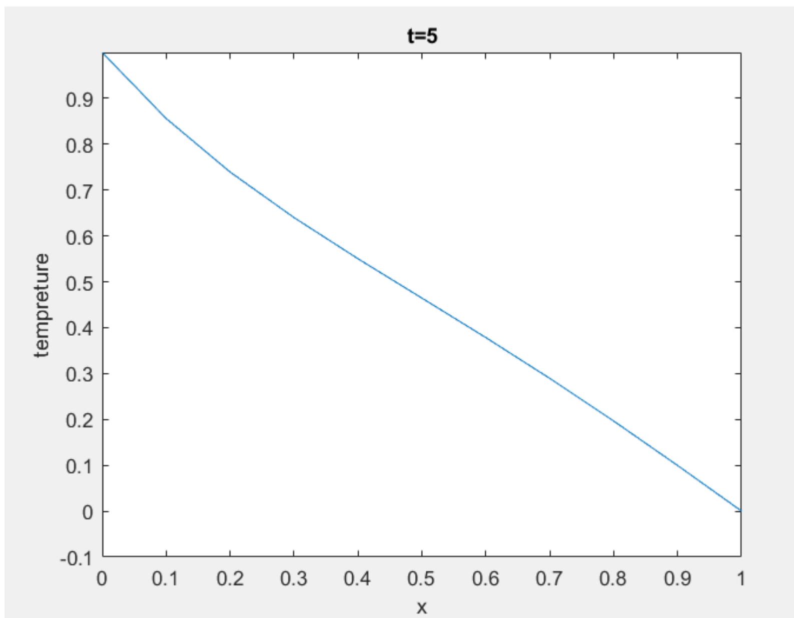

t=25
tempreture
x

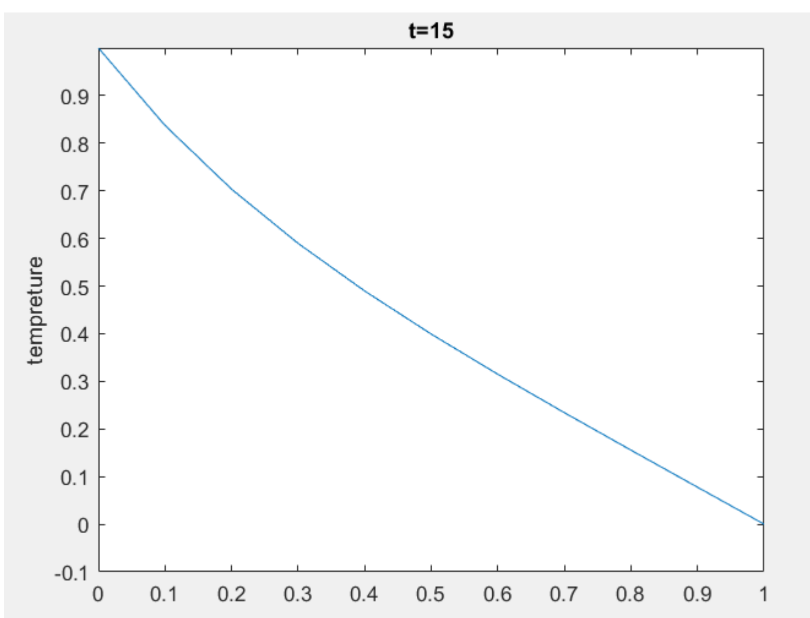
ب) کد دستوری زیر را اجرا می کنیم. برای ثانیه های ۵ و ۱۵ هم همانند کد زیر اجرا می کنیم:

```
x=linspace(0,1,11);
t=0:.5:15;
%t=linspace(0,30,100)
%t=[5,15,25];
u=pdepe(0,@equation2,@initialcondition2,@boundarycondition2,x,t);
figure(14);
surf(x,t,u)
title('Numerical solution computed with 100 mesh points.')
xlabel('Distance x')
ylabel('Time t')

figure(15)
plot(x,u(find(t==2),:))
title('t=2');
xlabel('x');
ylabel('tempreture');
axis([0 1 -0.1 1])
```




t=2
tempreture
x


t=5
tempreture
x


t=15
tempreture
x

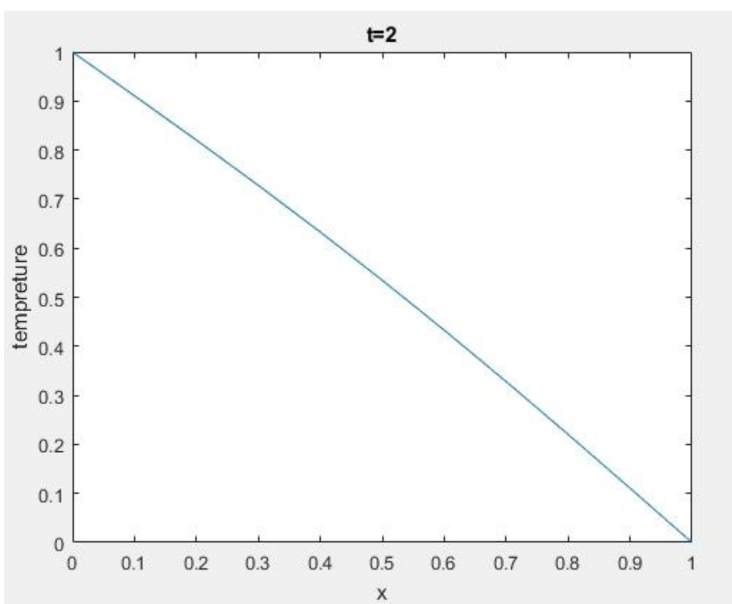
قسمت سوم: حل PDE در یک بعد به روش عددی

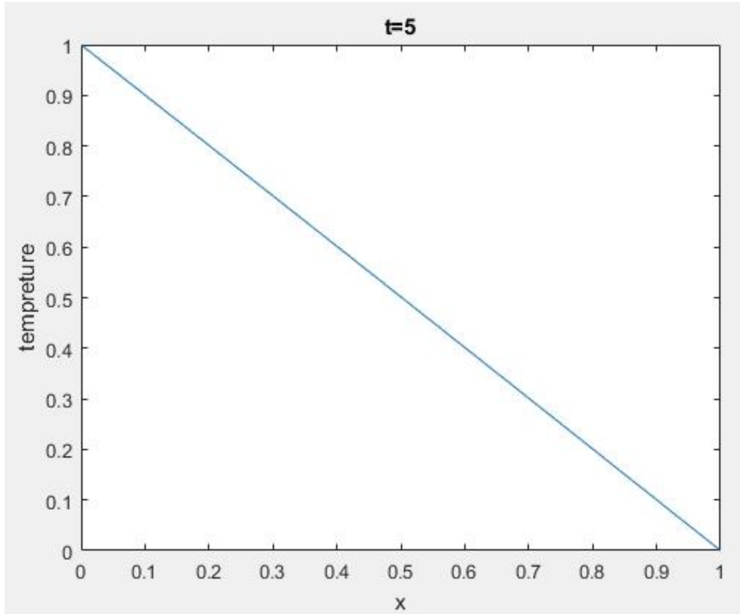
در این بخش می خواهیم معادلات PDE را به روش عددی حل کنیم. در این روش برای محاسبه مشتق های پاره ای همان گونه که در کلاس متلب توضیح داده شد، از روش اختلاف محدود استفاده می کنیم.

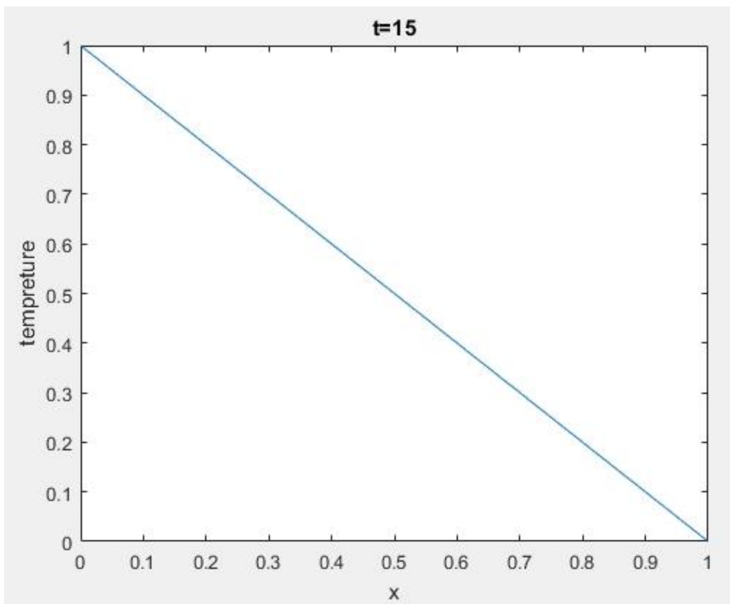
معادله ب قسمت قبل را در نظر می گیریم.

۱. ابتدا این معادله را با روش عددی حل کنید و منحنی دما برای نقاط مختلف میله در زمان های گفته شده رسم کنید .(قرار دهید $\Delta z = ۰.۰۵$ و $\Delta z = ۰.۱$)

```
dz = 0.1;                % each depth step is 0.1 meter
Nz = ceil(1 / dz) ;      % Choose the number of depth steps
Nt = 300;                % Choose the number of time steps
dt = 15/Nt;              % Length of each time step in seconds
K = 1/10;
z = linspace(0,1,Nz+1);
T = ones(Nz+1,Nt+1);     % Create temperature matrix with Nz+1 rows, and Nt+1 columns
time = [0:dt:dt*Nt];
T(1,:) = 1;
T(end,:) = 0;
T(:,1) = 1 - z.^2  ;
for i=2:Nt+1
        depth_2D = (T(1:end-2,i-1)-2*T(2:end-1,i-1)+T(3:end,i-1))/dz^2;
        time_1D = K*depth_2D ;
        T(2:end-1,i) = time_1D*dt + T(2:end-1,i-1);
end
figure(18)
plot(z,T(:,find(time==2)))
title('t=2');
xlabel('x');
ylabel('tempreture');
```

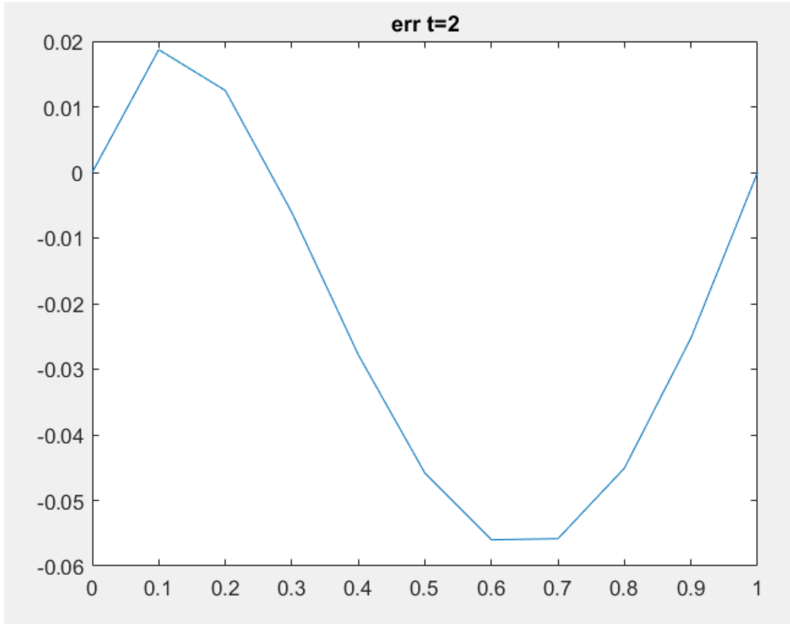


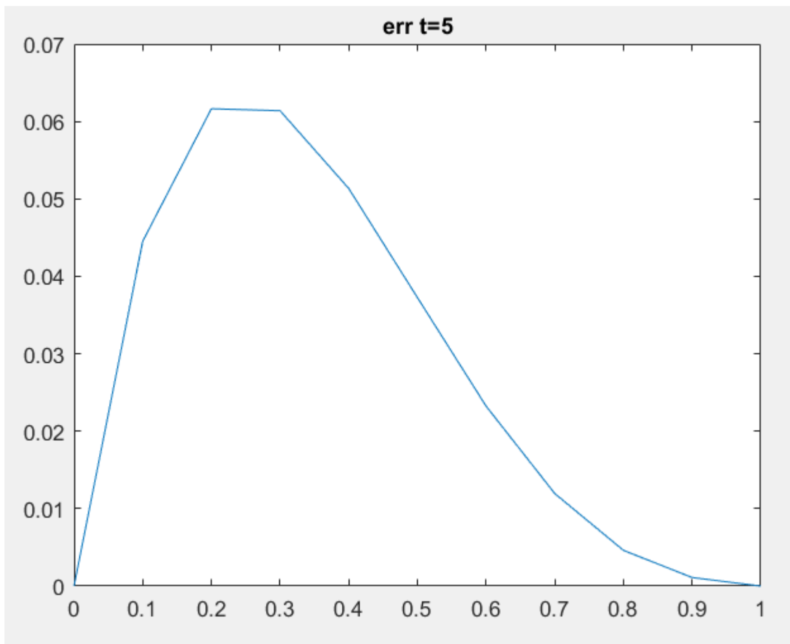


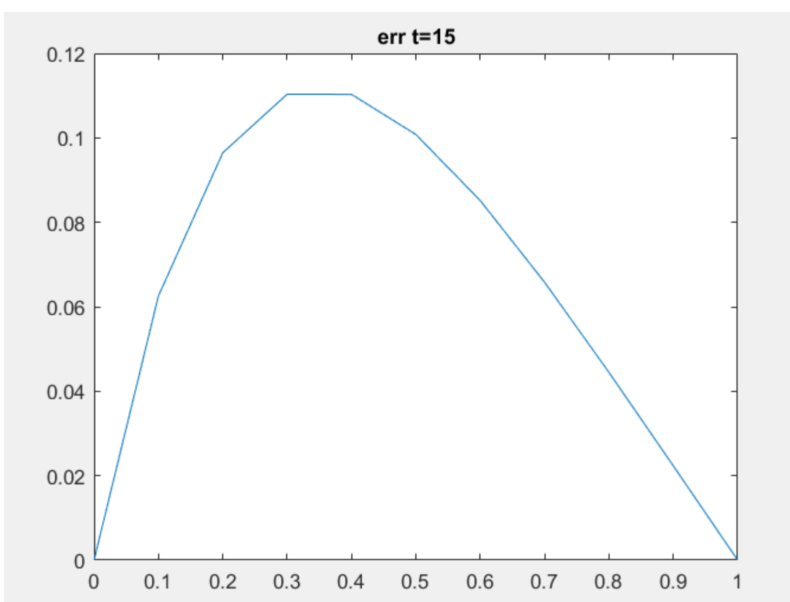


۲. حال نمودار اندازه تفاضل بین مقدار دقیق به دست آمده در قسمت قبل و مقدار به دست آمده از روش عددی را برای نقاط مختلف میله برای نقاط مختلف میله رسم کنید.

```
figure(21)
plot(z,T(:,find(time==2))-(u(find(t==2),:))')
title('err t=2');
figure(22)
plot(z,T(:,find(time==5))-u(find(t==5),:)')
title('err t=5');
figure(23)
plot(z,T(:,find(time==15))-u(find(t==15),:)')
title('err t=15');
```


err t=2
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1


err t=5
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1


err t=15
0.12
0.1
0.08
0.06
0.04
0.02
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

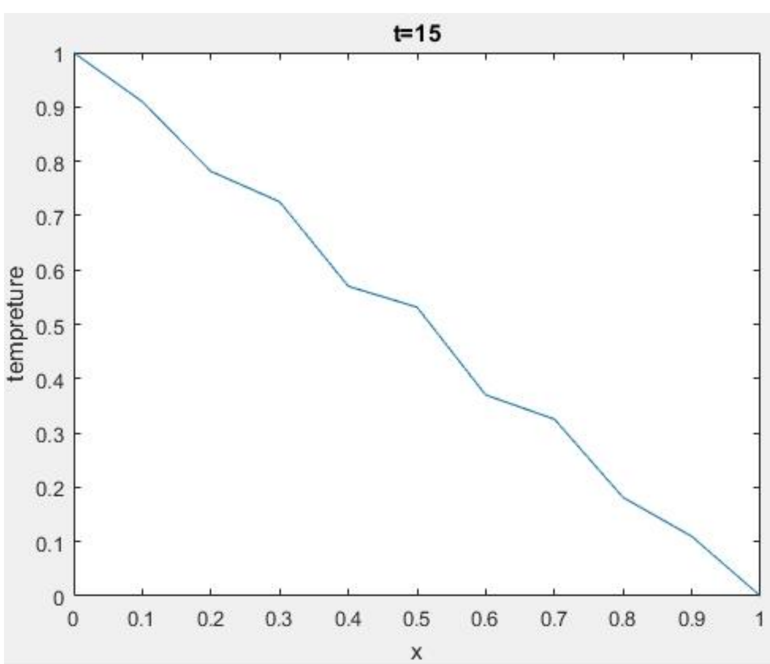
۳. حال نمودار اندازه تفاضل بین مقدار دقیق به دست آمده در قسمت قبل و مقدار به دست آمده از روش عددی را برای نقاط مختلف میله برای نقاط مختلف میله رسم کنید.

```
dz = 0.1;              % each depth step is 0.1 meter
Nz = ceil(1 / dz);     % Choose the number of depth steps
Nt = 290;              % Choose the number of time steps
dt = 15/Nt;            % Length of each time step in seconds
K = 1/10;
z = linspace(0,1,Nz+1);

d=K*dt/dz^2

T = ones(Nz+1,Nt+1);  % Create temperature matrix with Nz+1 rows, and Nt+1 columns

time = [0:dt:dt*Nt];
T(1,:) = 1;
T(end,:) = 0;
T(:,1) = 1 - z.^2  ;

for i=2:Nt+1
        depth_2D = (T(1:end-2,i-1)-2*T(2:end-1,i-1)+T(3:end,i-1))/dz^2;
        time_1D = K*depth_2D ;
        T(2:end-1,i) = time_1D*dt + T(2:end-1,i-1);
end
```

با اجرای کد بالا $\boldsymbol{d = 0.5172}$ بدست می آید.

حال با استفاده از کد دستوری زیر، نمودار دما را در ثانیه ۱۵ ام رسم می کنیم:

```
figure(20)
plot(z,T(:,find(time==15)))
title('t=15');
xlabel('x');
ylabel('tempreture');
```



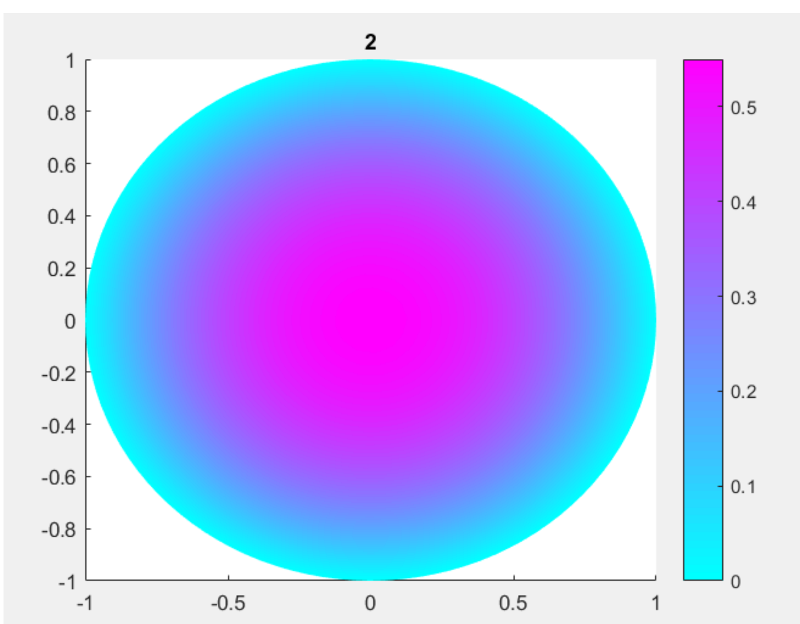
قسمت چهارم: حل PDE در دو بعد

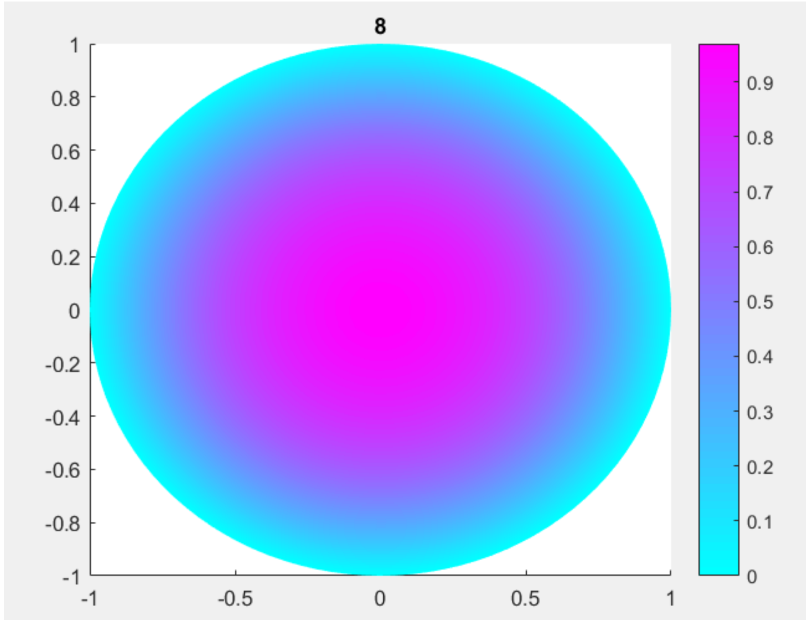
در زیر معادله یک پوسته مرتعش دایره ای به شعاع یک آمده است . با استفاده از متلب معادله داده شده را حل کنید.

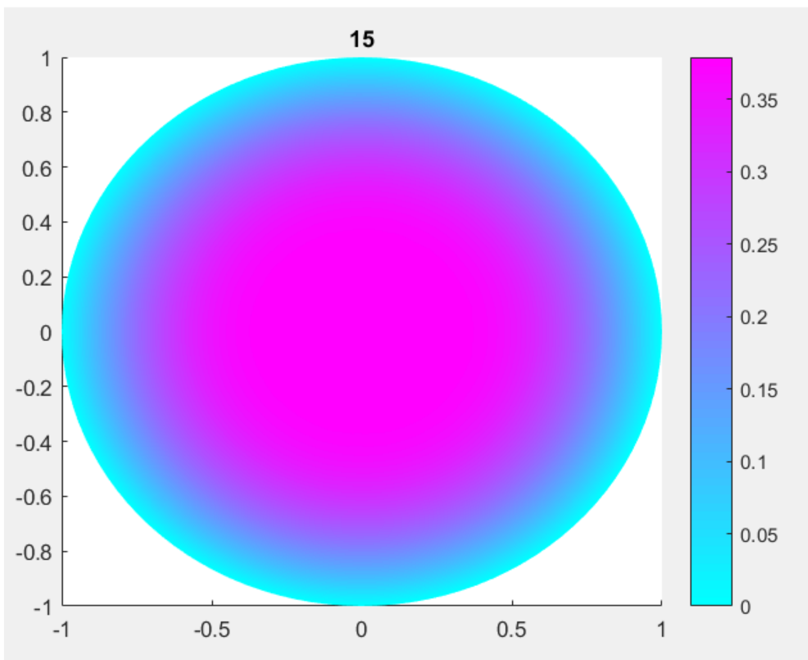
کد دستوری زیر را اجرا می کنیم:

```
clear
%syms locations
%defining geometry
gd = [1;0;0;1];
ns = [83;81;49];
sf = 'SQ1';
dl = decsg(gd,sf,ns);
model = createpde(1);
pg = geometryFromEdges(model,dl);
pdegplot(model,'EdgeLabels','on');
axis([-1.2 1.2 -1.2 1.2])

% Boundary condition
applyBoundaryCondition(model,'dirichlet','edge',1,'u',0);
applyBoundaryCondition(model,'dirichlet','edge',2,'u',0);
applyBoundaryCondition(model,'dirichlet','edge',3,'u',0);
applyBoundaryCondition(model,'dirichlet','edge',4,'u',0);

% initial condition
%ut0 = @(locations) 0;
u0  = @(locations) 1-(locations.x).^2-(locations.y).^2;

% solving PDE
%generateMesh(model,'Hmax',0.05);
generateMesh(model);
setInitialConditions(model,u0,0);
specifyCoefficients(model,'m',1,'d',0,'c',1,'a',0,'f',2);
tlist = 0:0.5:15;
results = solvepde(model,tlist);
u = results.NodalSolution;
```

حال ۳ شکل خواسته شده را به ترتیب می کشیم:




8
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0


15
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0