بخش ۱) حل معادله حرارت

بخش ۱.۵)

ابتدا باید با توجه به موارد گفته شده، معادله مورد نظر را پیاده سازی کنیم. طبق صورت پروژه معادله به فرم زیر است:

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} - \dfrac{1}{c^2}\dfrac{\partial u}{\partial t} = 0 \\ u(0,t) = 0 \\ u(2,t) = 50 \\ u(x,0) = 2e^x \end{cases} \quad , \quad \frac{1}{c^2} = 50$$

برای نمایش این معادله در متلب به شکل زیر عمل می‌کنیم:

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{c^2} * \frac{\partial u}{\partial t}$$

$$\frac{\partial^2 u}{\partial x^2} = 50 * \frac{\partial u}{\partial t}$$

ضریب c باید برابر ۵۰ باشد چون در واقع ضریب مشتق نسبت به زمان می‌باشد. با توجه به راهنمایی انجام شده می‌توان تغییرات را در تابع f هم اعمال کرد که در این صورت:

$$\frac{\partial^2 u}{\partial x^2} * \frac{1}{50} = \frac{\partial u}{\partial t}$$

$$f = \frac{\partial u}{\partial x}$$

واضح است که معادله ما همگن است. برای مشخص کردن اینکه معادله همگن است باید s برابر صفر باشد. بنابراین در کد متلب معادله به شکل زیر تعریف می‌شود:

فایل مربوط به این قسمت: Equation.m

```
Equation.m
function [c,f,s] = Equation(x,t,u,DuDx)
c=50;
f=DuDx;
s = 0;
end
```

حال به سراغ شرایط اولیه می‌رویم. برای تعریف شرایط اولیه طبق دستور داریم:

فایل مربوط به این قسمت: Initial_condition.m

```
Initial_condition.m
function u0= Initial_condition(x)
    u0=2*exp(x);
end
```

برای تعیین شرایط مرزی می‌توان به صورت زیر عمل کرد:

$$u(0,t) = 0 \; ul = 0$$

$$u(2,t) = 50 \; ur = 50$$

$$p\big(0,t,u(0,t)\big) = ul$$

$$p\big(2,t,u(2,t)\big) = ur - 50$$

چون شرایط مرزی دیگری وجود ندارد مقادیر ql و qr هردو برابر صفر خواهند بود. بنابراین شرایط مرزی در کد متلب به شکل زیر خواهد بود:

فایل مربوط به این قسمت: Border_Condition.m

```
Border_Condition.m
function [pl,ql,pr,qr]=Border_Condition(x1,ul,xr,ur,t)
pl=ul;
ql=0;
pr=ur-50;
qr=0;
end
```

برای حل معادله از تابع معرفی شده در دستور باید استفاده کنیم. در اینجا یک پارامتری به نام m تعریف می‌کنیم. در واقع این پارامتر نشان دهنده symmetry مسئله است. در مسئله ما با توجه به اینکه در فضای کارتزین کار می‌کنیم، مقدار m=0 را انتخاب می‌کنیم. بازه های زمان و مکان را مطابق دستور تعیین می‌کنیم. بنابراین فایل کد به شکل زیر خواهد بود:

فایل این قسمت:t_1.m و t_2.m و t_3.m

توجه: طبق هماهنگی انجام شده با چیف درس، بازه مربوط به x باید [0,2] باشد. هرچند مقدار اصلی در دستور پروژه به صورت کامنت در کد ها موجود است.

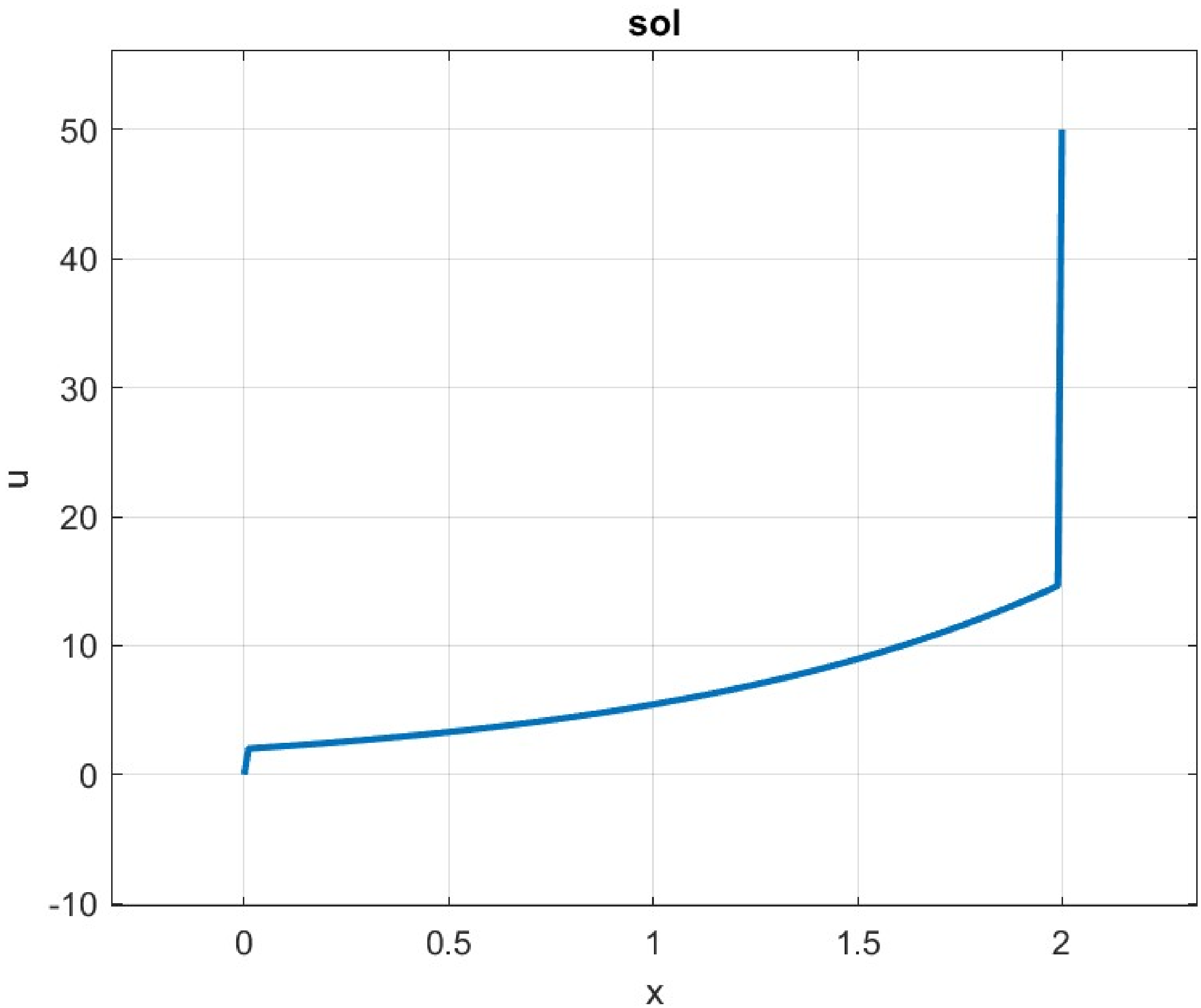
ابتدا برای t=0 بررسی می‌کنیم. کد به شکل زیر خواهد بود:

```
t_1.m
%x=linspace(0,1,200);
%t=linspace(0,10,201);
x=linspace(0,2,200);
t=linspace(0,10,201);
m=0;
sol=pdepe(m,@Equation,@Initial_condition,@Border_Condition,x,t);

u_at_t5=sol(t==0,:,1);
figure
plot(x,u_at_t5,'LineWidth',2)

xlabel('x')
ylabel('u')
title('sol')
grid on
```

با اجرای این قطعه کد در متلب، خروجی به شکل زیر خواهد بود:



تحلیل: برای t=0 **باید طبق شرایط مرزی از صفر تابع شروع شود و در** x=2 **دما باید ۵۰ باشد چون در اینجا از قاعده** $2e^{x}$ **استفاده می کنی . پس از ۱۴.۷ باید به ۵۰ برسیم.**

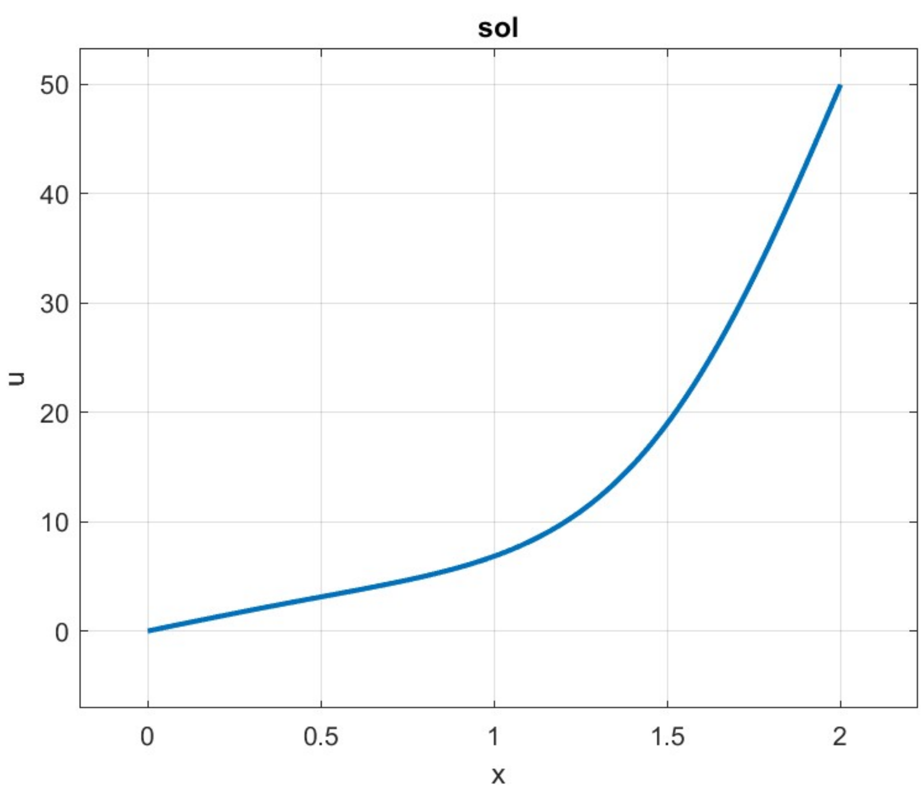
حال برای t=5 بررسی می‌کنیم. کد به شکل زیر خواهد بود:

```
%x=linspace(0,1,200);
%t=linspace(0,10,201);
x=linspace(0,2,200);
t=linspace(0,10,201);
m=0;
sol=pdepe(m,@Equation,@Initial_condition,@Border_Condition,x,t);

u_at_t5=sol(t==5,:,1);
figure
plot(x,u_at_t5,'LineWidth',2)

xlabel('x')
ylabel('u')
title('sol')
grid on
```

در صورت اجرای این کد خروجی به شکل زیر خواهد بود:



تحلیل: برای t=5 باید در x=0 دما برابر صفر باشد و در x=2 دما باید ۵۰ باشد.

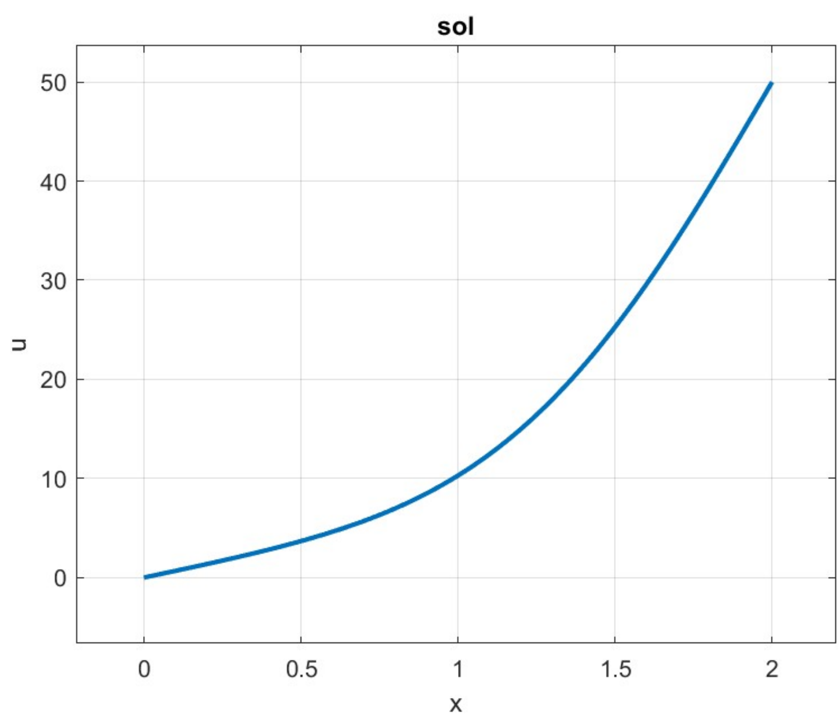
در نهایت برای t=10 بررسی می‌کنیم. کل به شکل زیر خواهد بود:

```
%x=linspace(0,1,200);
%t=linspace(0,10,201);
x=linspace(0,2,200);
t=linspace(0,10,201);
m=0;
sol=pdepe(m,@Equation,@Initial_condition,@Border_Condition,x,t);

u_at_t5=sol(t==10,:,1);
figure
plot(x,u_at_t5,'LineWidth',2)

xlabel('x')
ylabel('u')
title('sol')
grid on
```

با اجرای این کد خروجی به شکل زیر است:



تحلیل: برای t=10 هم باید این شرایط برقرار باشد در x=0 دما برابر صفر باشد و در x=2 دما باید ۵۰ باشد.

در این قسمت از توابع دیگری استفاده می‌کنیم برای نمایش دو بعدی تغییرات دمایی در هنگام تغییر زمان و مکان.

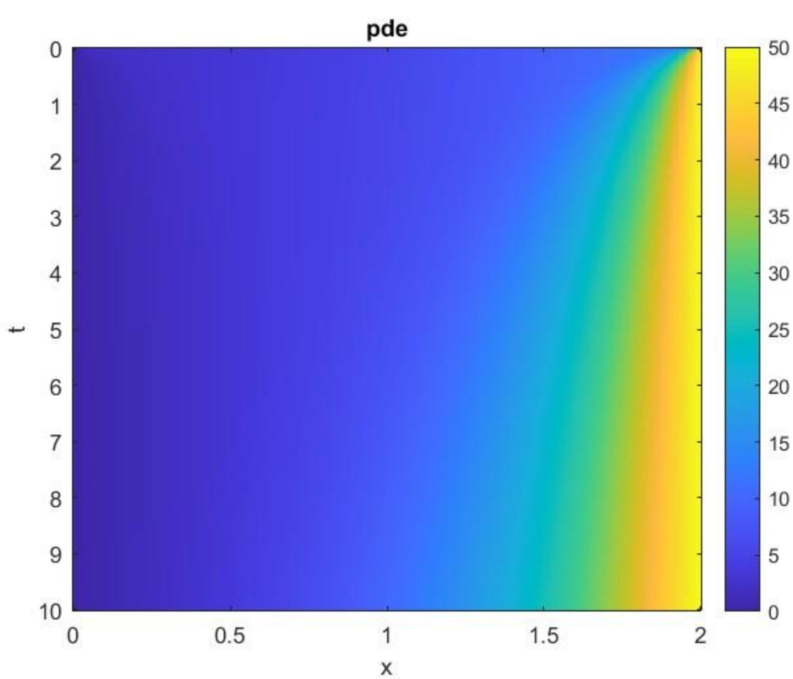
کد مربوط به این قسمت: colormap.m

کد به شکل زیر خواهد بود:

colormap.m

```
%x=linspace(0,1,200);
%t=linspace(0,10,201);
x = linspace(0, 2, 200);
t = linspace(0, 10, 201);
m=0;
sol=pdepe(m,@Equation,@Initial_condition,@Border_Condition,x,t);
u=sol(:,:,1);
figure
imagesc(x,t,u);
colorbar;
xlabel('x');
ylabel('t');
title('pde');
colormap('jet')
```

با اجرای این کد خروجی به شکل زیر خواهد بود:



تحلیل: در واقع این کد ابتدا معادله pde **را حل می‌کند و سپس جواب آن را به حالت** heatmap **نشان می‌دهد. در واقع رنگ هر قسمت نشان دهنده مقدار جواب معادله به ازای** x **و** t **مشخص است. با توجه به اینکه در تابع** colormap **از نوع** jet **استفاده کردیم که تحلیل رنگ ها در این حالت به شکل زیر خواهد بود:**

- **آبی تیره: کمترین مقدار**
- **آبی: کمی بیشتر از مینیمم**
- **فیروزه‌ای: مقادیر میانی**
- **سبز: بیشتر از مقادیر میانی**
- **زرد: مقادیر بالا نزدیک ماکزیمم**
- **نارنجی: مقادیر بیشتر از زرد**
- **قرمز: بیشترین مقدار**

حال تعداد تقسیم های x **و** t **را تغییر می‌دهیم و به ۱۰۰ و ۱۰۱ تقسیم می‌کنیم.**

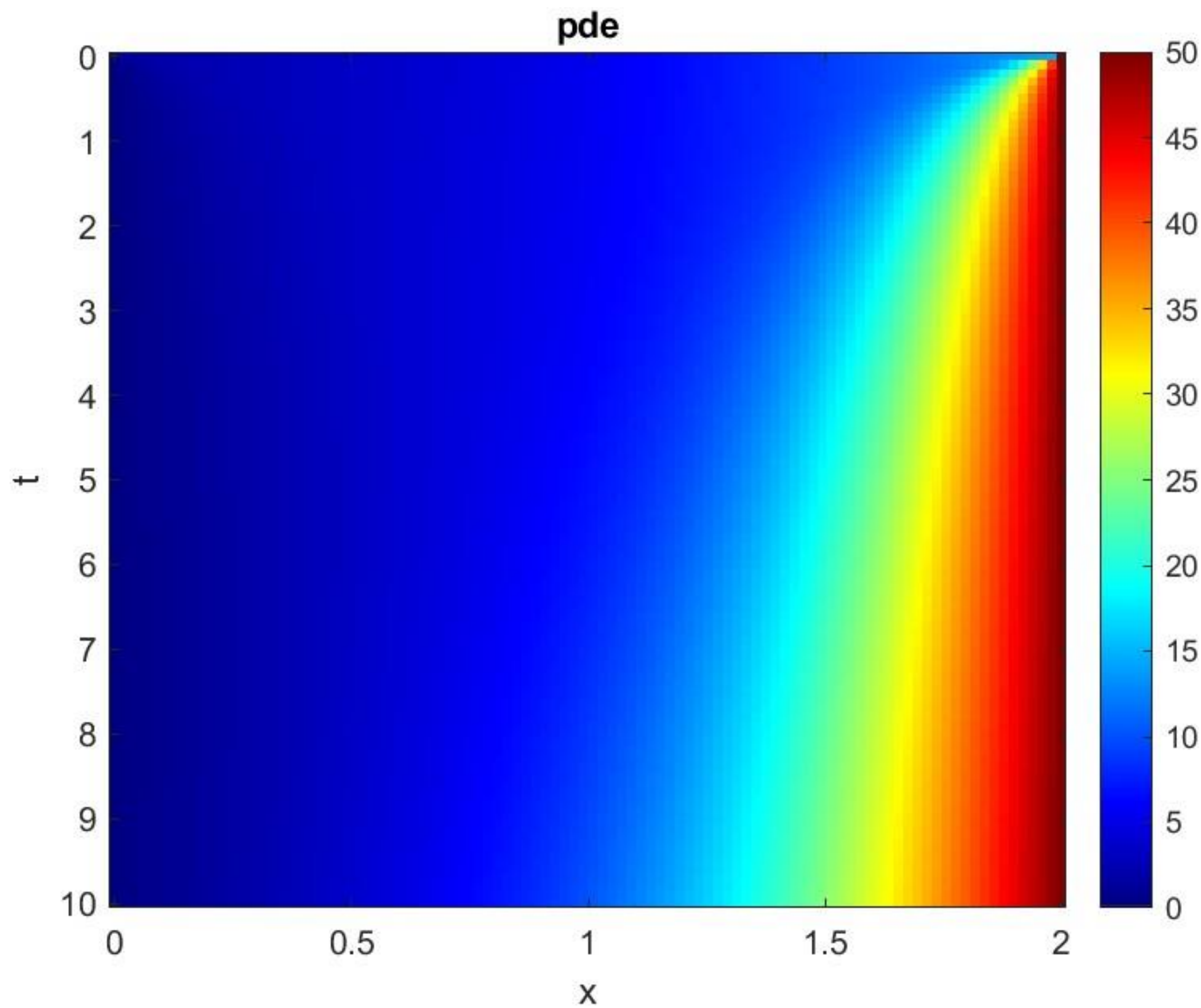
کد این قسمت: colormap_2.m

کد به شکل زیر خواهد بود:

colormap_2.m

```
x = linspace(0, 2, 100);
t = linspace(0, 10, 101);
m=0;
sol=pdepe(m,@Equation,@Initial_condition,@Border_Condition,x,t);
u=sol(:,:,1);
figure
imagesc(x,t,u);
colorbar;
xlabel('x');
ylabel('t');
title('pde');
colormap('jet')
```

با اجرای کد به خروجی زیر می‌رسیم:



حال تلاش می‌کنیم تا نمودار های سه بعدی رسم کنیم.

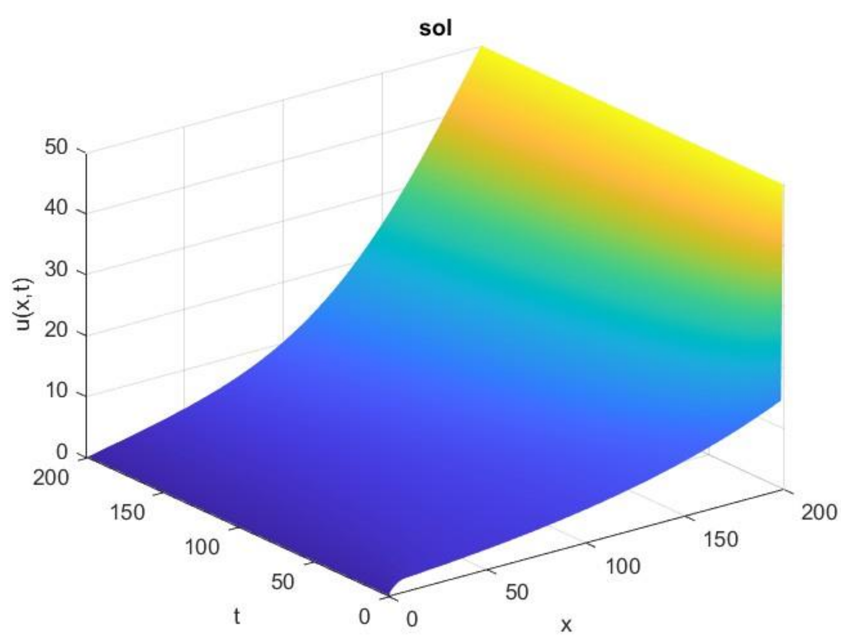
کد های این قسمت: surf_1.m و surf_2.m

ابتدا برای تقسیم بندی ۲۰۰ و ۲۰۱ بررسی می‌کنیم که کد به شکل زیر است:

surf_1.m | surf_2.m

```
%x=linspace(0,1,200);
%t=linspace(0,10,201);
x=linspace(0,2,200);
t=linspace(0,10,201);
m=0;
sol=pdepe(m,@Equation,@Initial_condition,@Border_Condition,x,t);
u=sol(:,:,1);
surf(sol)
xlabel('x')
ylabel('t')
zlabel('u(x,t)')
title('sol')
shading interp
```

با اجرای این کد داریم:



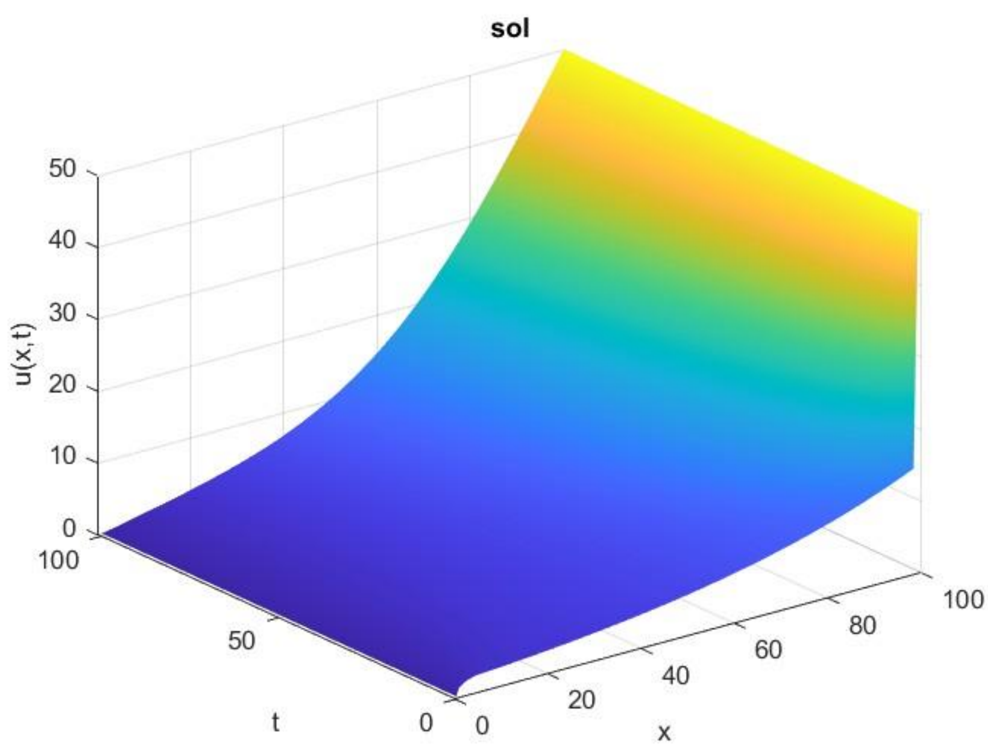
حال این مراحل را برای تقسیم بندی ۱۰۰ و ۱۰۱ هم انجام می‌دهیم که کد به شکل زیر خواهد بود:

surf_2.m

```
%x=linspace(0,1,100);
%t=linspace(0,10,101);
x=linspace(0,2,100);
t=linspace(0,10,101);
m=0;
sol=pdepe(m,@Equation,@Initial_condition,@Border_Condition,x,t);
u=sol(:,:,1);
surf(sol)
xlabel('x')
ylabel('t')
zlabel('u(x,t)')
title('sol')
shading interp

```

با اجرای این کد داریم:



بخش ۲) معادله هلمهولتز

بخش امتیازی:

به دست آوردن معادله هلمهولتز از معاده موج به صورت زیر قابل انجام است:

$$(\nabla^2 - \frac{1}{c^2}\frac{\delta^2}{\delta x^2})u(r,t) = 0$$

$$u(r,t) = A(r)T(t)$$

$$\frac{\nabla^2 A}{A} = \frac{1}{c^2 T}\frac{d^2 T}{dT^2}$$

$$\frac{\nabla^2 A}{A} = -k^2$$

$$\frac{1}{c^2 T}\frac{d^2 T}{dT^2} = -k^2$$

$$\nabla^2 A + k^2 A = (\Delta^2 + k^2)A = 0$$

برای بررسی حل این معادله، روش کلی را در مختصات کارتزین بیان می‌کنیم:

فرض کنیم یک دامنه معمولی داریم بنابراین می‌توانیم جواب معادله را به صورت حاصل ضرب یه تابع جداگانه بنویسیم که هر کدام به یکی از متغیر های فضای کارتزین وابسته است:

$$\psi(x,y,z) = X(x)Y(y)Z(z)$$

$$\frac{d^2X}{dx^2} + k_x^2 X = 0$$

$$\frac{d^2Y}{dy^2} + k_y^2 Y = 0$$

$$\frac{d^2Z}{dz^2} + k_z^2 Z = 0$$

$$k_x^2 + \ k_y^2 Y + \ k_z^2 Z = \ k^2$$

راه حل کلی این معادله به جواب زیر منجر خواهد شد:

$$X(x) = Acos(k_x x) + Bsin(k_x x)$$

$$Y(y) = Ccos\left(k_y y\right) + Dsin(k_y y)$$

$$Z(z) = Ecos(k_z z) + Fsin(k_z z)$$

$$\psi(x, y, z) = (Acos(k_x x) + Bsin(k_x x))(Ccos\left(k_y y\right) + Dsin\left(k_y y\right))(Ecos(k_z z) + Fsin(k_z z))$$

که مقادیر A,B,C,D,E,F به کمک شرایط مرزی قابل محاسبه خواهند بود.

بنابراین می‌توان گفت روشی که برای حل این معادله استفاده می‌کنیم به دامنه، شرایط مرزی و فرم های خاص آن بستگی دارد. به طور کلی می‌توان سه راه برای حل این معادله در نظر گرفت:

۱. **جداسازی متغیر ها: مانند راه حل کلی که در بالا مطرح شد که در نهایت منجر به حل چند ODE می‌شود.**
۲. **تبدیل فوریه: با استفاده از تبدیل فوریه می‌توان معادله را به گونه تبدیل کرد که دامنه آن دارای تناوب باشد. بنابراین با به دست آوردن تناوب آن مثلا w معادله را حل می‌کنیم و سپس به فرم معمولی بر می‌گردانیم**
۳. **تابع گرین**

انواع شرایط مرزی معادله هلمهولتز:

۱. **دیریکله: در این حالت مقدار تابع** ψ **را در شرایط** S **بر روی دامنه** Ω **تعریف می‌کنیم که** ψ|S=f**. در این عبارت** f **همان تابع داده شده روی** s **است**

۲. **نیومن: مشتق معمولی** ψ **را در شرایط مرزی** s **بیان می‌کند.**

$$\frac{\delta\psi}{\delta n}(in\ S) = g$$

۳. **رابین یا ترکیبی: در واقع ترکیبی از نیومن و دیریکله است.**

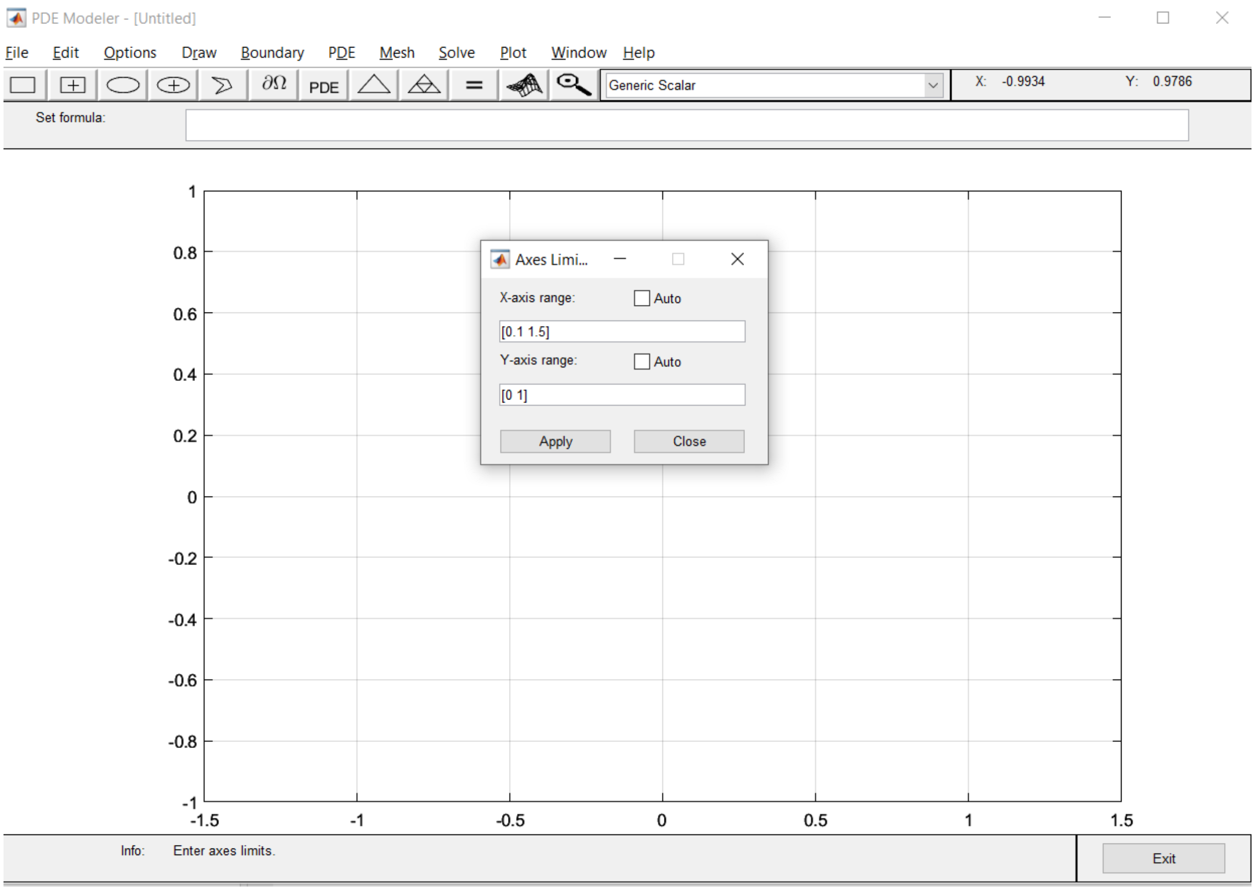
$$\alpha\psi + \beta\frac{\delta\psi}{\delta n} = h$$

که در آن α **و** β **توابع داده شده روی** s **هستند.**

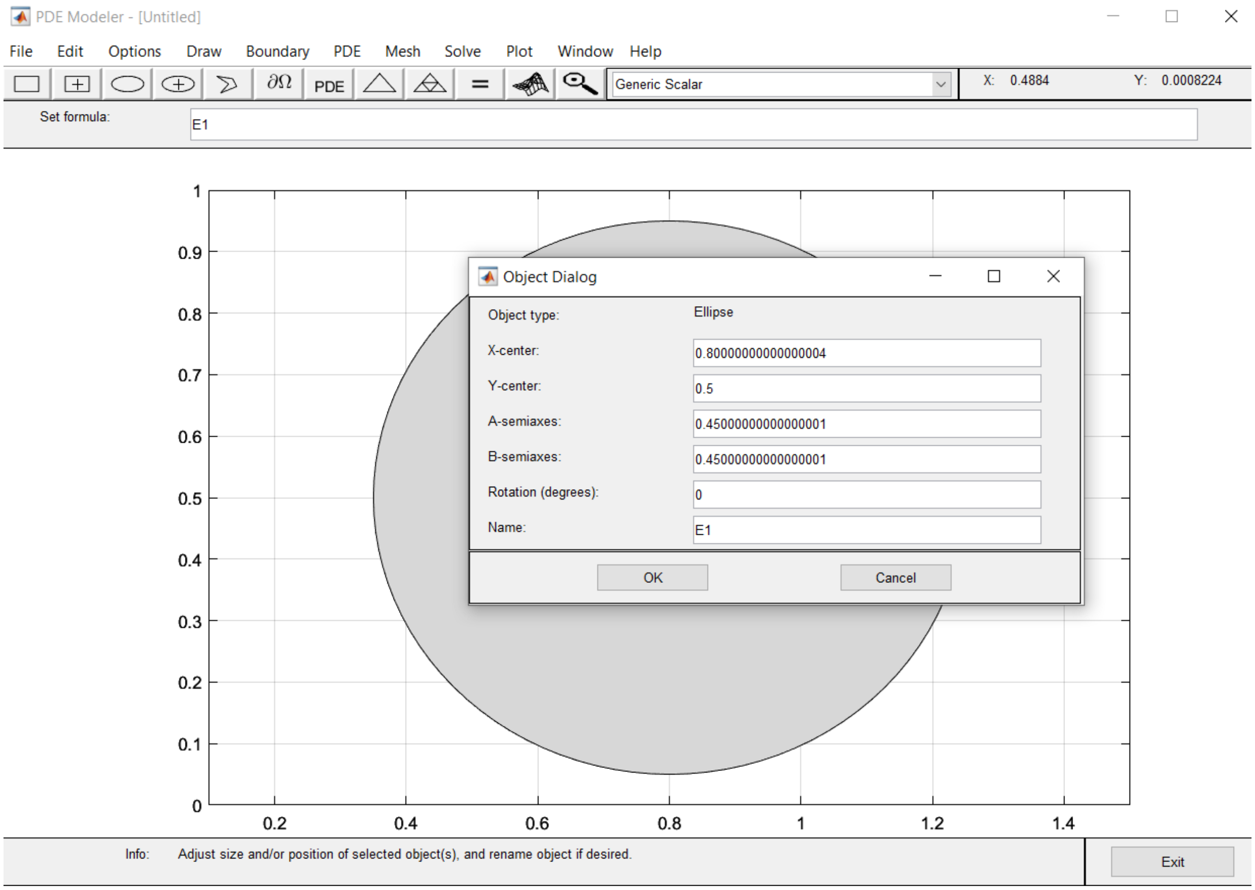
بخش ۲.۱) مراحل حل معادله:

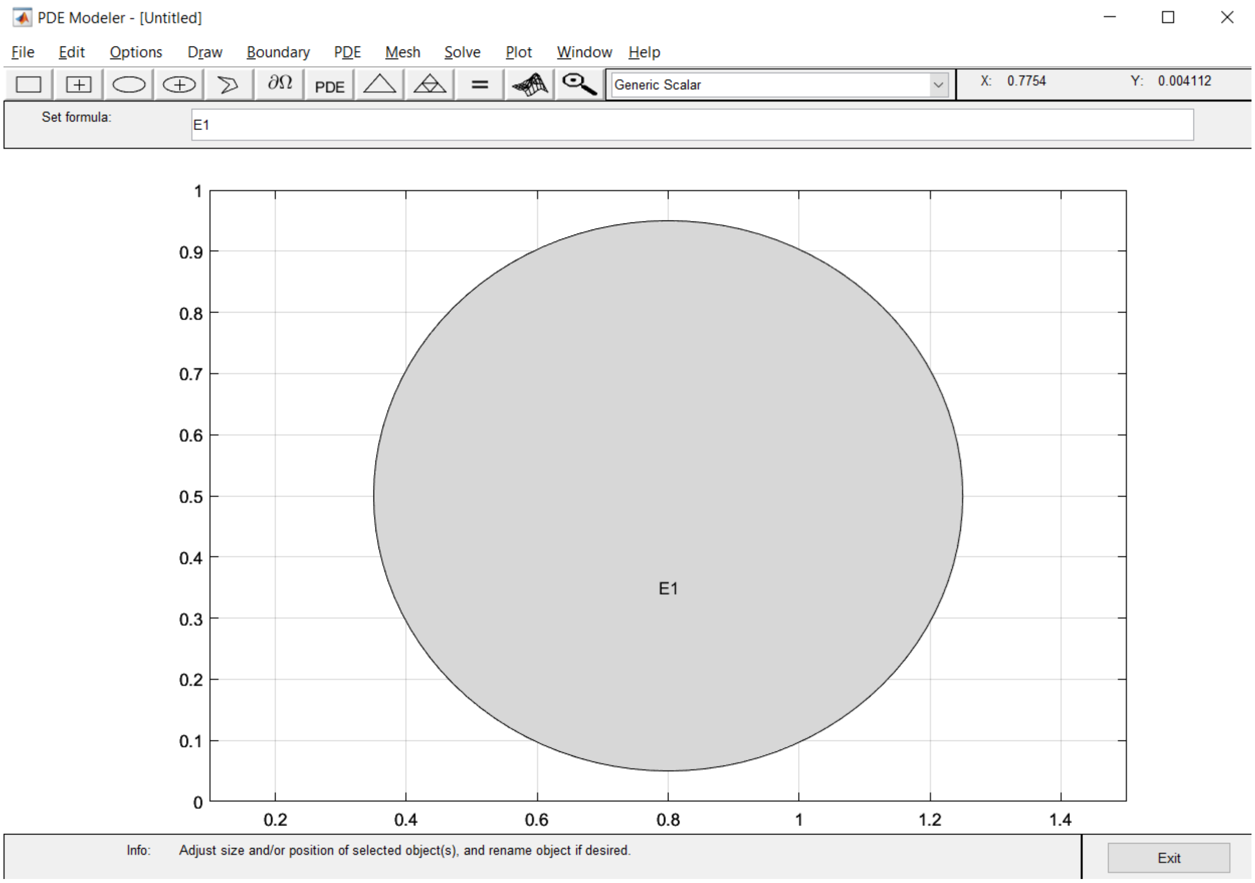
ابتدا طبق دستور در قسمت command window **در متلب عبارت** pdeModeler **را تایپ می‌کنیم. با زدن اینتر پنجره جدیدی باز می‌شود.**

سپس وارد options **می‌شویم. تنظیمات لازم را اعمال می‌کنیم. مثلا** grid **را روشن می‌کنیم و محدوده** x **و** y **را از** Axes limits **انتخاب می‌کنیم.**

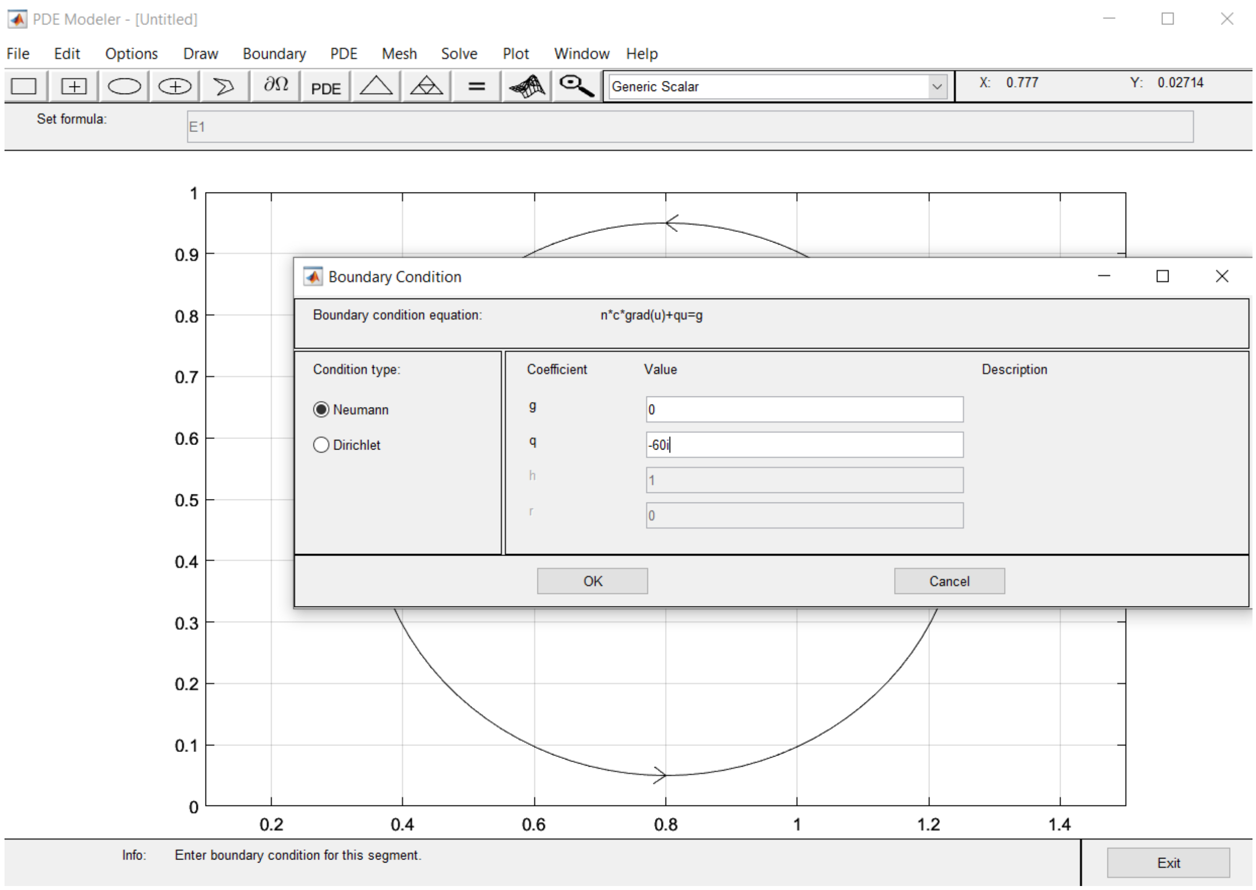


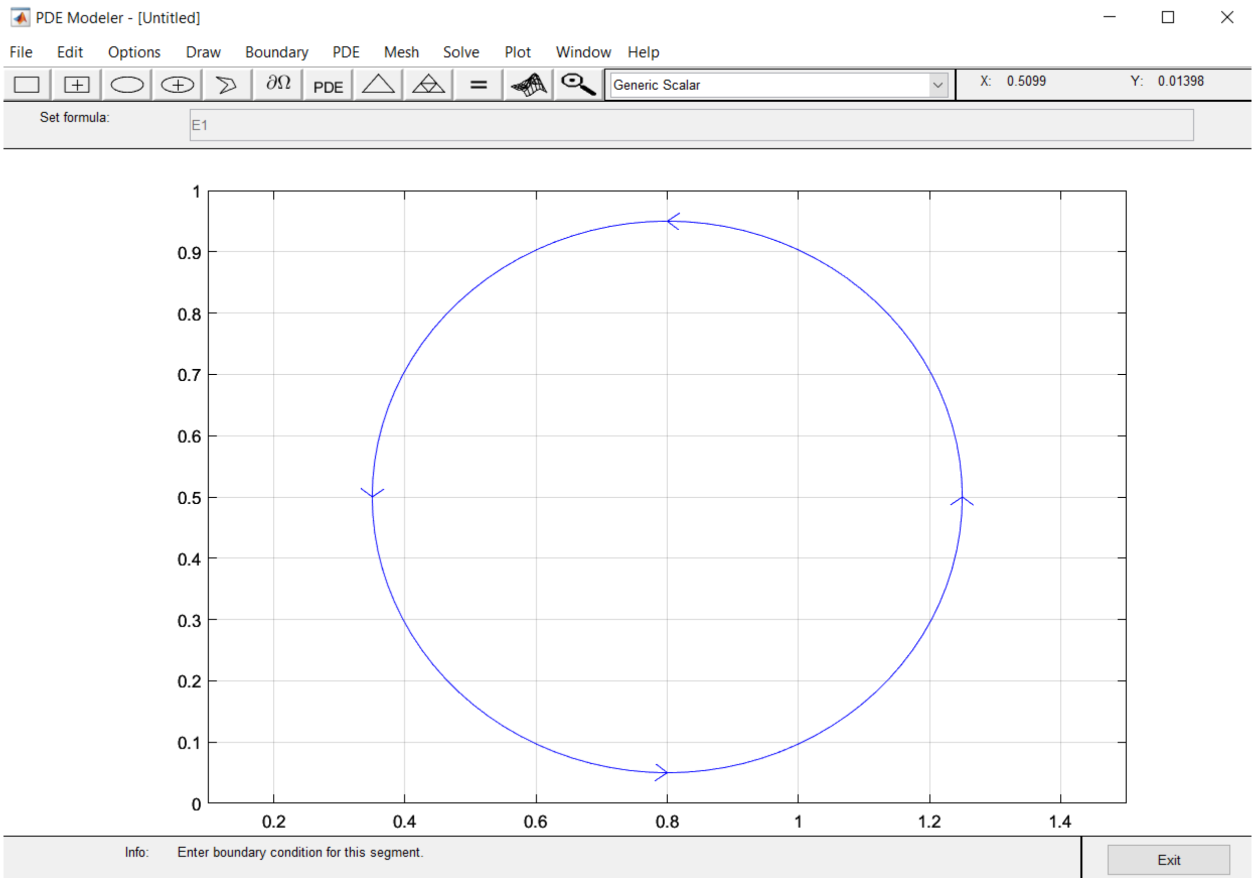
از منوی Draw وارد draw mode می‌شویم و دایره رسم می‌کنیم. با دوبار کلیک روی دایره مشخصات آن را طبق دستور تعیین می‌کنیم.


PDE Modeler - [Untitled]
File Edit Options Draw Boundary PDE Mesh Solve Plot Window Help
PDE
Generic Scalar
X: 0.4884 Y: 0.0008224
Set formula:
E1
Object Dialog
Object type:
Ellipse
X-center:
0.80000000000000004
Y-center:
0.5
A-semiaxes:
0.45000000000000001
B-semiaxes:
0.45000000000000001
Rotation (degrees):
0
Name:
E1
OK
Cancel
Info:
Adjust size and/or position of selected object(s), and rename object if desired.
Exit

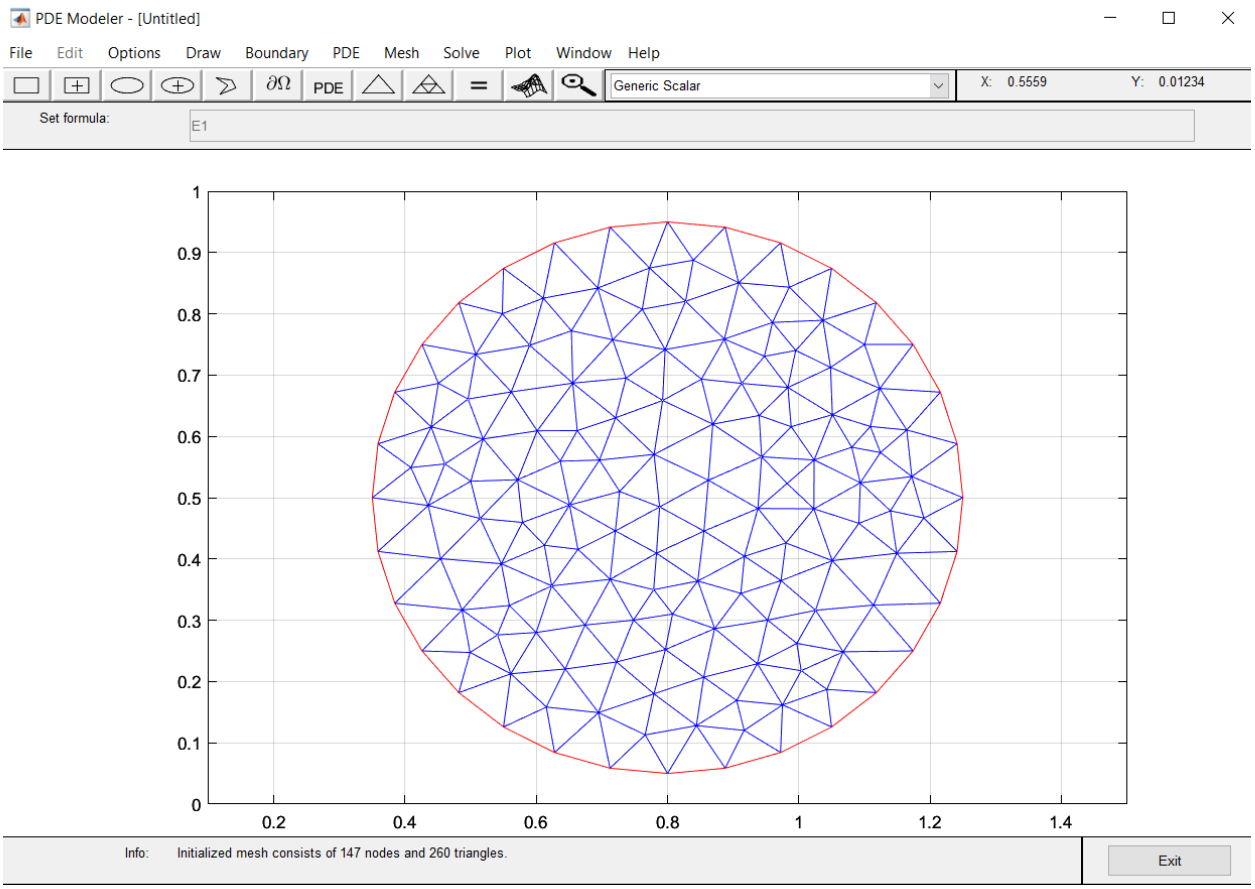


از منوی boundary **مقادیر دستور پروژه را اعمال می‌کنیم**

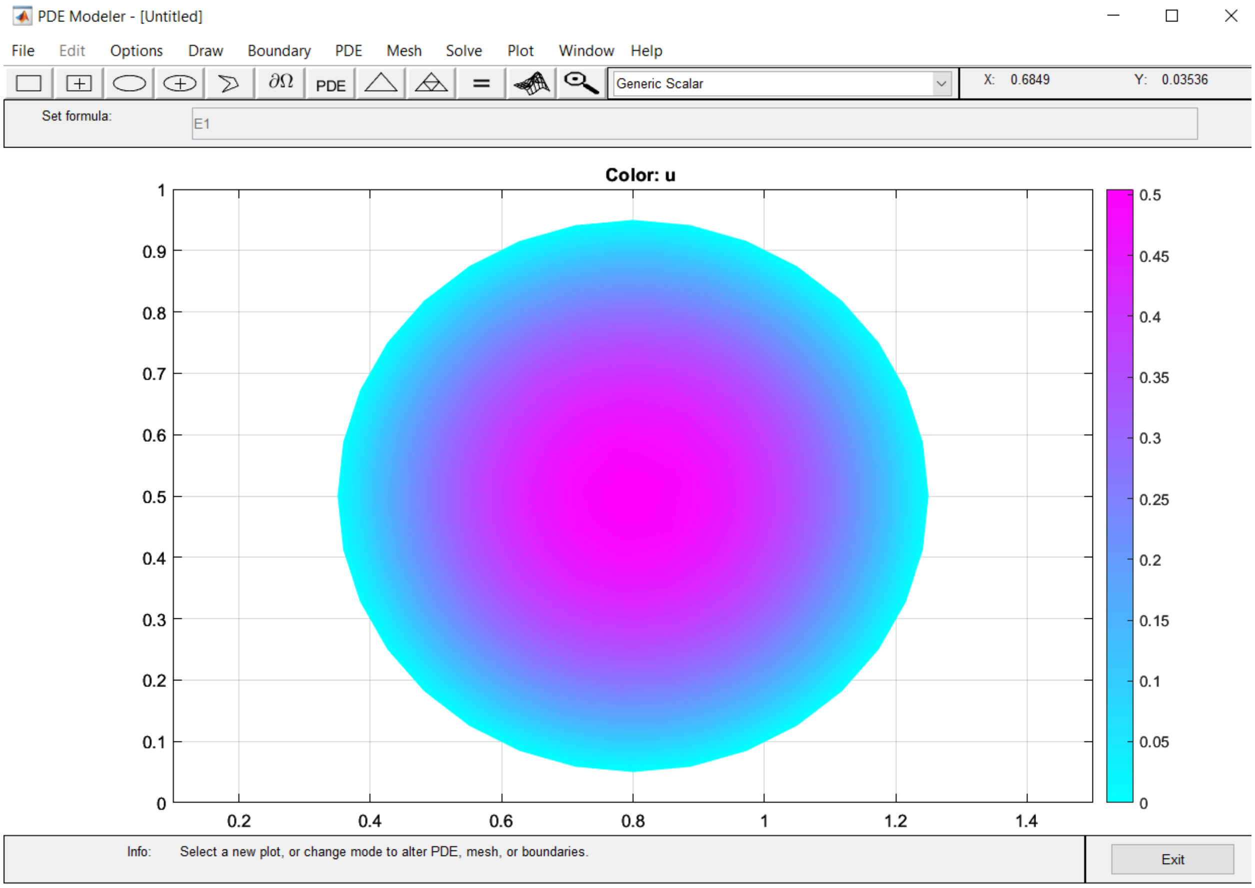

PDE Modeler - [Untitled]
File Edit Options Draw Boundary PDE Mesh Solve Plot Window Help
Generic Scalar
X: 0.777
Y: 0.02714
Set formula:
E1
Boundary Condition
Boundary condition equation:
n*c*grad(u)+qu=g
Condition type:
Neumann
Dirichlet
Coefficient
Value
Description
g
0
q
-60i
h
1
r
0
OK
Cancel
Info:
Enter boundary condition for this segment.
Exit



وارد منوی mesh می‌شویم و با زدن initial mesh دایره تغییر می‌کند



سپس از منوی solve گزینه solve pde را انتخاب می‌کنیم



تحلیل: تغییر رنگ از مرکز به بیرون نشان دهنده تغییر دامنه u است. این تفاوت رنگ تفاوت دامنه ها را نشان می دهد. در واقع در این نوع می توان گفت موج نوسان می کند و منتشر نمی شود. مرکز دارای حداکثر مقدار است که در واقع می تواند نشان دهنده ارتعاش باشد که مرکز بالاترین دامنه را دارد و کم کم دامنه به سمت بیرون کاهش می یابد.