偏微分方程的 MATLAB 求解精讲©

作者：dynamic

时间：2008.12.10

MATLAB 求解微分/偏微分方程，一直是一个头大的问题，两个字，“难过”，由于 MATLAB 对 LaTeX 的支持有限，所有方程必须化成 MATLAB 可接受的标准形式，不支持像其他三个数学软件那样直接傻瓜式输入，这个真把人给累坏了！

不抱怨了，还是言归正传，回归我们今天的主体吧！

MATLAB 提供了两种方法解决 PDE 问题，一是 pdepe()函数，它可以求解一般的 PDEs，据用较大的通用性，但只支持命令行形式调用。二是 PDE 工具箱，可以求解特殊 PDE 问题，PDEtool 有较大的局限性，比如只能求解二阶 PDE 问题，并且不能解决偏微分方程组，但是它提供了 GUI 界面，从繁杂的编程中解脱出来了，同时还可以通过 File->Save As 直接生成 M 代码

一、一般偏微分方程组(PDEs)的 MATLAB 求解

1、pdepe 函数说明

MATLAB 语言提供了 pdepe()函数，可以直接求解一般偏微分方程(组)，它的调用格式为

sol=pdepe(m,@pdefun,@pdeic,@pdebc,x,t)

【输入参数】

@pdefun：是 PDE 的问题描述函数，它必须换成下面的标准形式

$$c(x,t,\frac{\partial u}{\partial x})\frac{\partial u}{\partial t}=x^{-m}\frac{\partial}{\partial x}[x^m f(x,t,u,\frac{\partial u}{\partial x})]+s(x,t,u,\frac{\partial u}{\partial x}) \quad (式1)$$

这样，PDE 就可以编写下面的入口函数

[c,f,s]=pdefun(x,t,u,du)

m,x,t 就是对应于(式 1)中相关参数，du 是 u 的一阶导数，由给定的输入变量即可表示出出 c,f,s 这三个函数

@pdebc：是 PDE 的边界条件描述函数，必须先化为下面的形式

$$p(x,t,u)+q(x,t,u).*f(x,t,u,\frac{\partial u}{\partial x})=0$$

于是边值条件可以编写下面函数描述为

[pa,qa,pb,qb]=pdebc(x,t,u,du)

其中 a 表示下边界，b 表示下边界

@pdeic：是 PDE 的初值条件，必须化为下面的形式

$$u(x,t_0)=u_0$$

股我们使用下面的简单的函数来描述为

u0=pdeic(x)

m,x,t：就是对应于(式 1)中相关参数

【输出参数】

sol：是一个三维数组，sol(:,:,i)表示 u_i 的解，换句话说 u_k 对应 x(i)和 t(j)时的解为 sol(i,j,k)

通过 sol，我们可以使用 pdeval()直接计算某个点的函数值

2、实例讲解

试求解下面的偏微分

$$\begin{cases} \dfrac{\partial u_1}{\partial t} = 0.024\dfrac{\partial^2 u_1}{\partial x^2} - F(u_1 - u_2) \\ \dfrac{\partial u_2}{\partial t} = 0.17\dfrac{\partial^2 u_2}{\partial x^2} - F(u_1 - u_2) \end{cases}$$

其中，$F(x) = e^{5.73x} - e^{-11.46x}$，且满足初始条件 $u_1(x,0) = 1, u_2(x,0) = 0$ 及边界条件

$$\frac{\partial u_1}{\partial x}(0,t) = 0, u_2(0,t) = 0, u_1(1,t) = 1, \frac{\partial u_2}{\partial x}(1,t) = 0$$

【解】

（1）对照给出的偏微分方程，根据标注形式，则原方程可以改写为

$$\begin{bmatrix}1\\1\end{bmatrix}.*\frac{\partial}{\partial t}\begin{bmatrix}u_1\\u_2\end{bmatrix} = \frac{\partial}{\partial t}\begin{bmatrix}0.024\dfrac{\partial u_1}{\partial x}\\ 0.17\dfrac{\partial u_2}{\partial x}\end{bmatrix} + \begin{bmatrix}-F(u_1-u_2)\\ F(u_1-u_2)\end{bmatrix}$$

可见 m=0，且 $c = \begin{bmatrix}1\\1\end{bmatrix}, f = \begin{bmatrix}0.024\dfrac{\partial u_1}{\partial x}\\ 0.17\dfrac{\partial u_2}{\partial x}\end{bmatrix}, s = \begin{bmatrix}-F(u_1-u_2)\\ F(u_1-u_2)\end{bmatrix}$

```
%% 目标 PDE 函数
function [c,f,s]=pdefun (x,t,u,du)
c=[1;1];
f=[0.024*du(1);0.17*du(2)];
temp=u(1)-u(2);
s=[-1;1].*(exp(5.73*temp)-exp(-11.46*temp));
```

（2）边界条件改写为

下边界 $\begin{bmatrix}0\\u_2\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix}.*f = \begin{bmatrix}0\\0\end{bmatrix}$　上边界 $\begin{bmatrix}u_1-1\\0\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix}.*f = \begin{bmatrix}0\\0\end{bmatrix}$

```
%% 边界条件函数
function [pa,qa,pb,qb]=pdebc(xa,ua,xb,ub,t)
%a 表示下边界，b 表示上边界
pa=[0;ua(2)];
```

```
qa=[1;0];
pb=[ub(1)-1;0];
qb=[0;1];
```

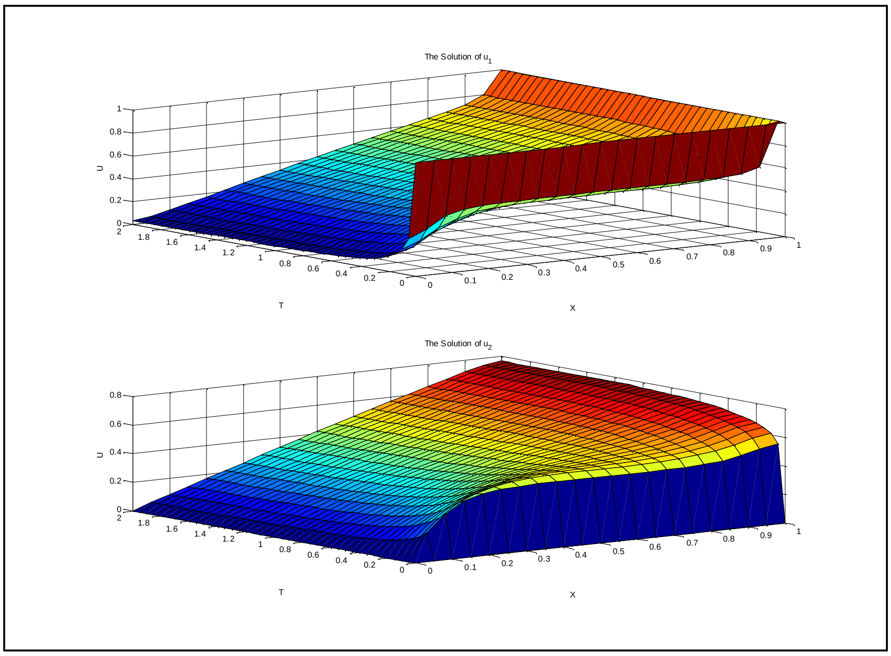
（3）初值条件改写为

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

```
%% 初值条件函数
function u0=pdeic(x)
u0=[1;0];
```

（4）最后编写主调函数

```
clc
x=0:0.05:1;
t=0:0.05:2;
m=0;
sol=pdepe(m,@pdefun,@pdeic,@pdebc,x,t);
figure('numbertitle','off','name','PDE Demo——by Matlabsky')
subplot(211)
surf(x,t,sol(:,:,1))
title('The Solution of u_1')
xlabel('X')
ylabel('T')
zlabel('U')
subplot(212)
surf(x,t,sol(:,:,2))
title('The Solution of u_2')
xlabel('X')
ylabel('T')
zlabel('U')
```



二、PDEtool 求解特殊 PDE 问题

MATLAB 的偏微分工具箱(PDE toolbox)可以比较规范的求解各种常见的二阶偏微分方程，但是惋惜的是只能求解特殊二阶的 PDE 问题，并且不支持偏微分方程组！

PDE toolbox 支持命令行形式求解 PDE 问题，但是要记住那些命令以及调用形式真的很累人，还好 MATLAB 提供了 GUI 可视交互界面 pdetool，在 pdetool 中可以很方便的求解一个 PDE 问题，并且可以帮我们直接生成 M 代码(File->Save As)。

下面我们先了解下三个典型的二阶 PDE，然后介绍 pdetool，至于命令行我们就免了，真的很累人，如果的确需要的话，那就让 Matalb 直接生成就好了。

1、典型偏微分方程的描述

（1）椭圆型偏微分方程的一般形式为

$$-div(c\nabla u)+a*u=f(x,t)$$

即

$$-c*(\frac{\partial^2}{\partial {x_1}^2}+\frac{\partial^2}{\partial {x_2}^2}+\mathbf{L}+\frac{\partial^2}{\partial {x_n}^2})u+a*u=f(x,t)$$

其中c,a,f为给定的函数或者常数

（2）抛物线型偏微分方程的一般形式

$$d*\frac{\partial u}{\partial t}-div(c\nabla u)+a*u=f(x,t)$$

即

$$d*\frac{\partial u}{\partial t}-c*(\frac{\partial^2}{\partial {x_1}^2}+\frac{\partial^2}{\partial {x_2}^2}+\mathbf{L}+\frac{\partial^2}{\partial {x_n}^2})u+a*u=f(x,t)$$

其中d,c,a,f必须是常数

（3）双曲线型偏微分方程的一般形式

$$d*\frac{\partial^2 u}{\partial t^2}-div(c\nabla u)+a*u=f(x,t)$$

即

$$d*\frac{\partial^2 u}{\partial t^2}-c*(\frac{\partial^2}{\partial {x_1}^2}+\frac{\partial^2}{\partial {x_2}^2}+\cdots+\frac{\partial^2}{\partial {x_n}^2})u+a*u=f(x,t)$$

其中d,c,a,f必须是常数

（4）特征值型偏微分方程的一般形式，注意它是（1）的变形，不能算独立的一类

$$-div(c\nabla u)+a*u=\lambda*d*u$$

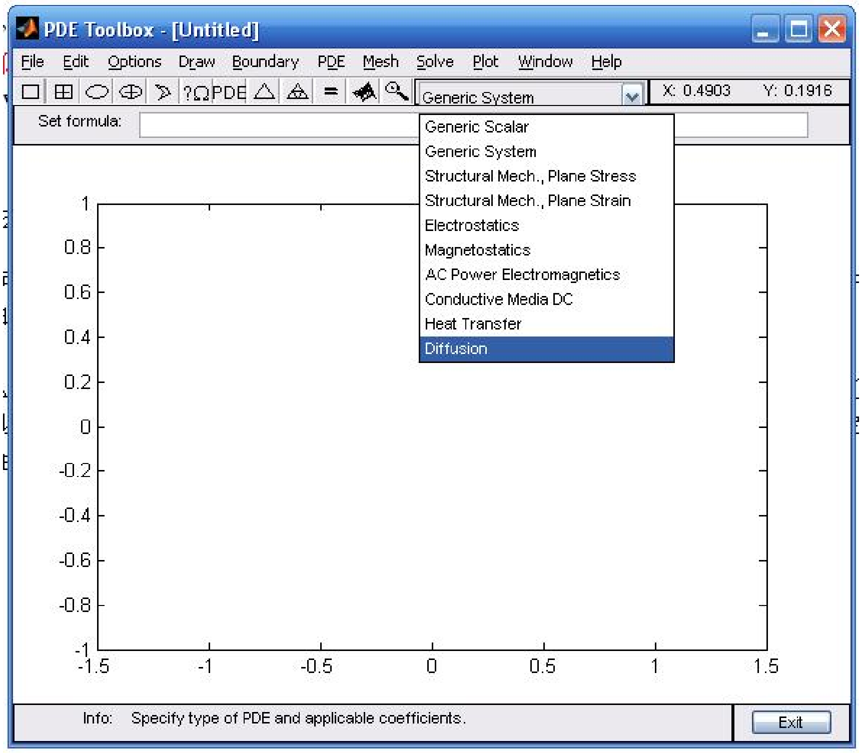
即

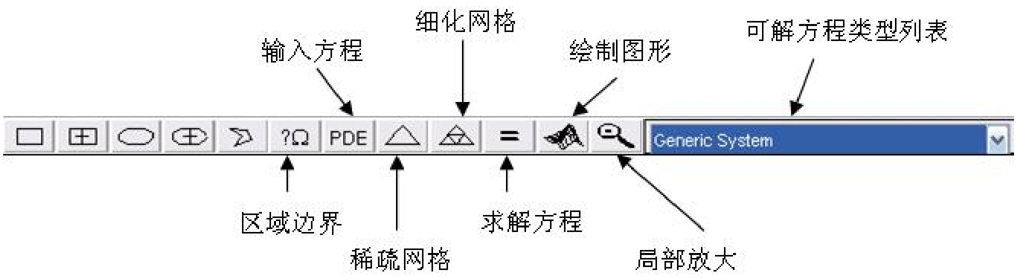
$$-c*(\frac{\partial^2}{\partial {x_1}^2}+\frac{\partial^2}{\partial {x_2}^2}+\cdots+\frac{\partial^2}{\partial {x_n}^2})u+a*u=\lambda*d*u$$

从上面可以看出，三类典型二阶偏微分方程的区别在于 u 对 t 的导数阶次。**椭圆型 PDEs 中，c、a、d 和 f 可以是给定的函数或者常数，但是其它两类必须都是常数**。

MATLAB 是采用有限元的方法求解各种 PDE。MATLAB 为我们提供一个 pdetool 的交互界面，可以求解二元偏微分 u(x1,x2)(注意只能求解二元)。

方程的参数由 a、c、d 和 f 确定，求解域由图形确定，求解域确定好后，需要对求解域进行栅格化(这个是自动)。





2、偏微分方程边界条件的描述

一般在 PDE 中边界条件包括 Dirichlet(狄利克莱)条件和 Neumann(纽曼)条件:

（1）Dirichlet 条件

一般描述为

$h(x,t,u,\frac{\partial u}{\partial x})*u|_{\partial\Omega}=r(x,t,u,\frac{\partial u}{\partial x})$，其中$\partial\Omega$表示求解域的边界

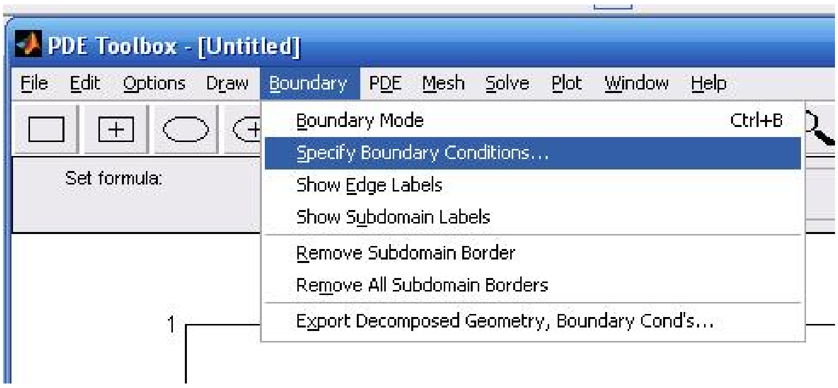
假设在边界上满足该方程，则只需给出 r 和 h 即可，它们可以是常数也可以是给定的函数

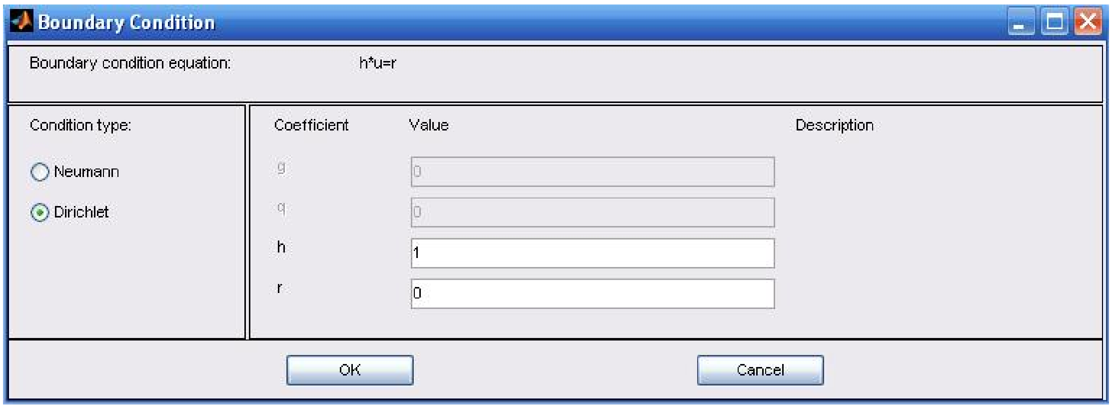
（2）Neumann 条件

一般描述为

$[\frac{\partial}{\partial n}(c\nabla u)+q*u]|_{\partial\Omega}=g$，其中$\frac{\partial u}{\partial n}$表示$u$的法向偏导数

通过下面的操作调出边界条件设置，注意在这之前一定要使用【区域边界】按钮制定边界





3、求解实例

试求解双曲线型偏微分方程

$$\frac{\partial^2 u}{\partial t^2}-\frac{\partial^2 u}{\partial x^2}-\frac{\partial^2 u}{\partial y^2}+2u=10$$

求解域 s 为

$$s1: x^2+y^2\le 9$$

$$s2: \frac{x^2}{4}+\frac{y^2}{16}\le 1$$

$$s=(s1\cup s2)-(s1\cap s2)$$

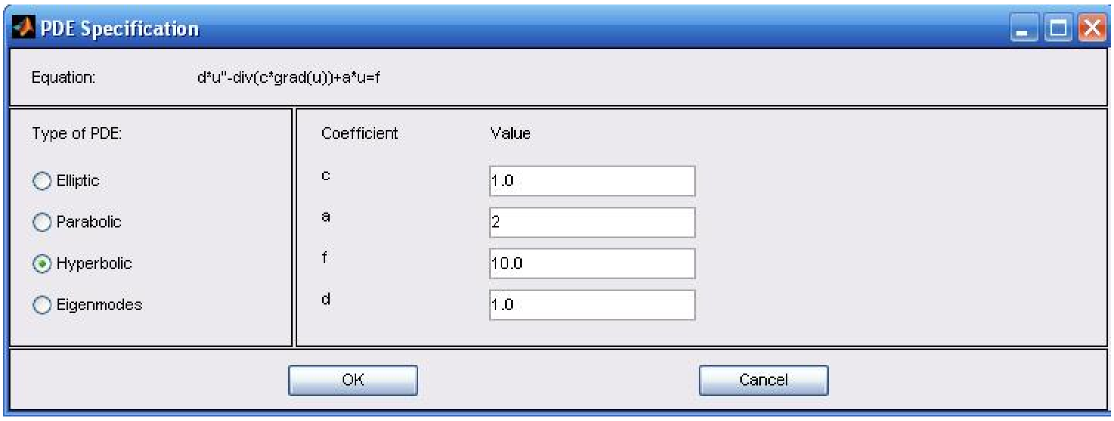
边界条件为

构成求解域的边界值都为 5

【解】

由给定的 PDE，可以的 d=1,c=1,a=2,f=10，再说一次，对于抛物线和双曲线型偏微分方程 4 个系数必须是常数，否则 MATLAB 无能为力

step1:点击工具栏的【PDE】按钮，如下输入 PDE 的参数，注意选择 Hyperbolic

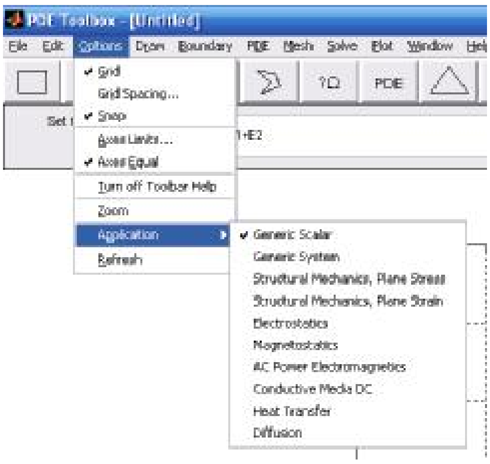


step2:绘制求解域

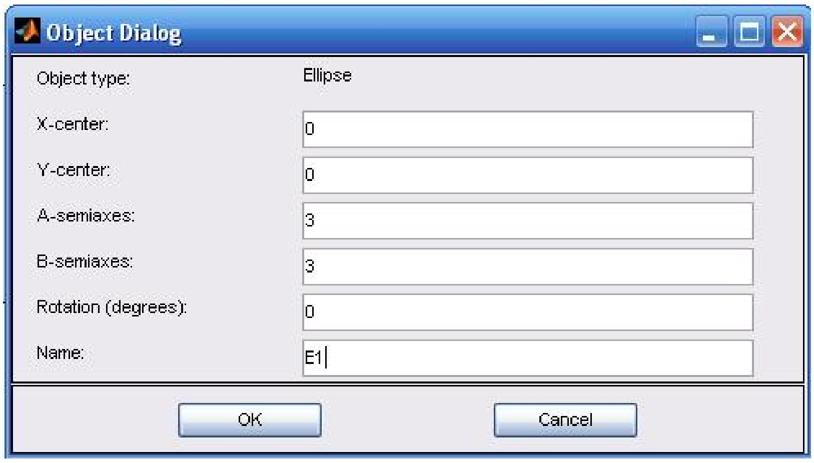
对坐标轴的操作可以在【Options】主菜单中操作，包括设置网格、坐标系范围等

(1)【Options】->Axis Limits 设置如下

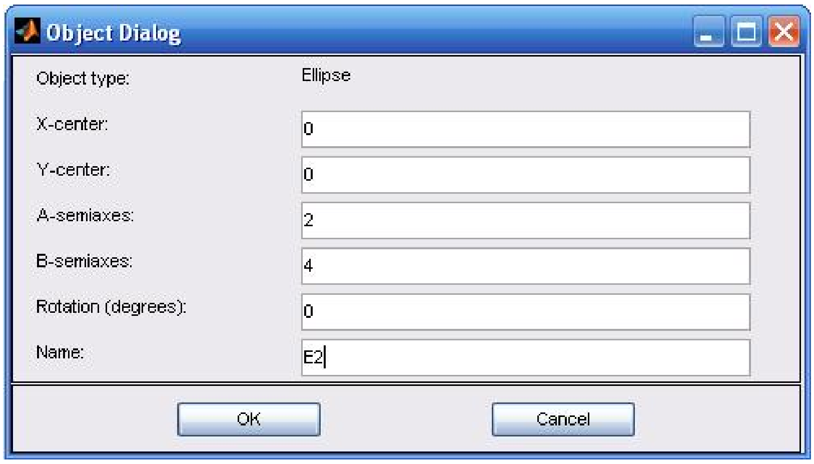
其它设置如下



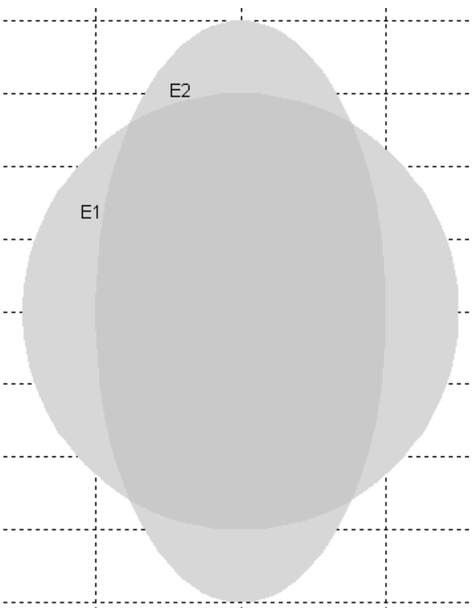
(2)点击工具栏上的第三个按钮【绘制椭圆】，任意绘制一个椭圆，双击椭圆，设置如下



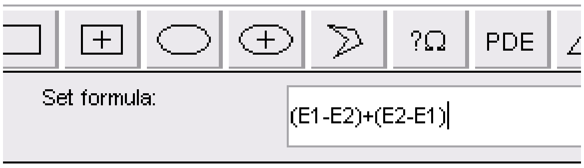
重复上面的操作，参数如下



于是得到



(3)在 set formula 中如下输入，“+”表示求并集，“-”表示求差集，注意没有直接求交接的操作符

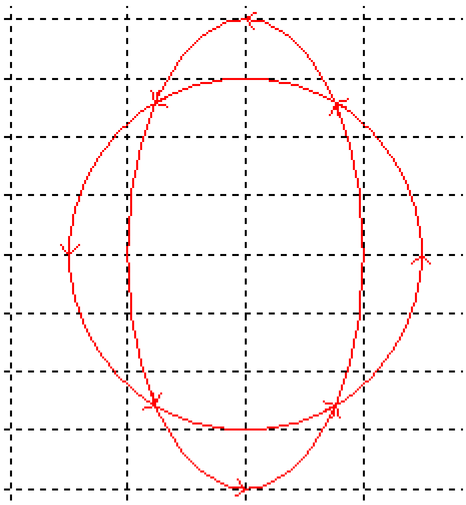


step3:边界条件和初值条件

初值条件可以通过【Solve】->【Parameters…】设置

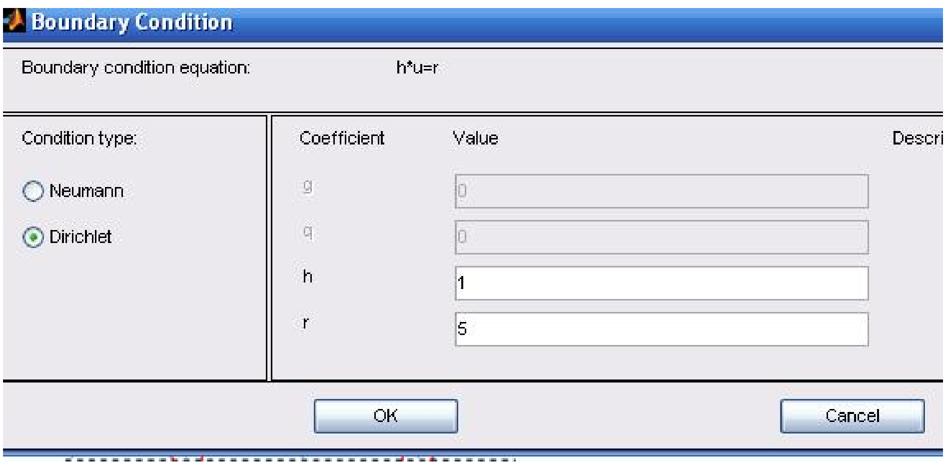
边值条件设置如下

(1)点击工具栏的第 6 个按钮【区域边界】，显示如下



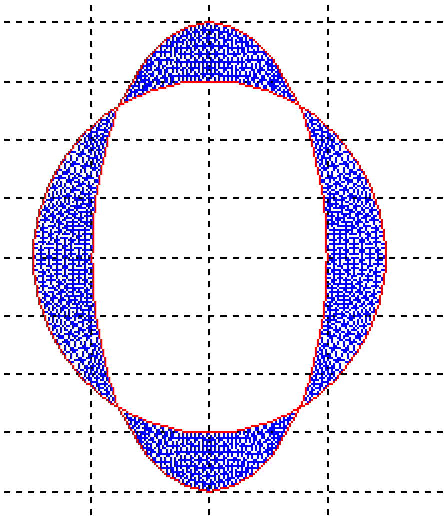
(2)【Boundary】->【Remove All Subdomain Borders】移除所有子域的边界，将得到所有子域合并成一个求解域

(3) 【Boundary】->【Secify Boundary Conditons…】设置边界如下，注意我们这里只有 Dirichlet 条件



step4:生成使用有限元方法求解方程所需的栅格

点击工具栏的第 8/9 个按钮，对求解域生成栅格，多次点击可以在原来基础上继续细化栅格，直到自己觉得满意为止，当然可以通过【Mesh】主菜单进行精确控制

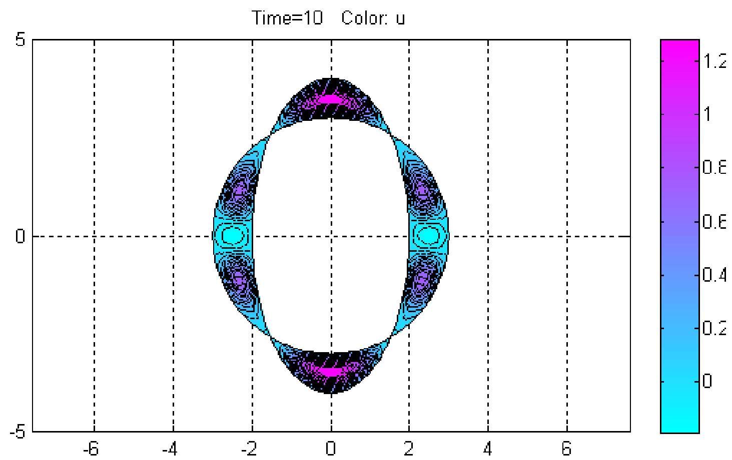


step5:求解方程

点解工具栏的第 10 个按钮“=”【求解方程】

step6:求解结果绘图

点击第 11 个按钮【绘制图形】，里面的选项很丰富，可以绘制等高线等好多，甚至播放动画，具体大家可以自己慢慢摸索



动画播放设置：

(1)【Solve】->【Parameters】设置合适的时间向量 Time

(2)【Plot】->【Parameters】选中【Animation】，点击后面的【Options】，设置播放速度和次数，比如 6fps 表示每秒 6 帧

(3)【Plot】->【Export Movie…】输入动画保存的变量名，比如 M

(4)在 Command Windows 中直接输入 movie(M)即可播放

(5)使用 movie2ve(M,’demo.avi’)命令可以将动画保存为 avi 文件