

## 第二十章 偏微分方程的数值解

自然科学与工程技术中种种运动发展过程与平衡现象各自遵守一定的规律。这些规律的定量表述一般地呈现为关于含有未知函数及其导数的方程。我们将只含有未知多元函数及其偏导数的方程，称之为偏微分方程。

方程中出现的未知函数偏导数的最高阶数称为偏微分方程的阶。如果方程中对于未知函数和它的所有偏导数都是线性的，这样的方程称为线性偏微分方程，否则称它为非线性偏微分方程。

初始条件和边界条件称为定解条件，未附加定解条件的偏微分方程称为泛定方程。对于一个具体的问题，定解条件与泛定方程总是同时提出。定解条件与泛定方程作为一个整体，称为定解问题。

### §1 偏微分方程的定解问题

各种物理性质的定常（即不随时间变化）过程，都可用椭圆型方程来描述。其最典型、最简单的形式是泊松(Poisson)方程

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) \quad (1)$$

特别地，当  $f(x, y) \equiv 0$  时，即为拉普拉斯(Laplace)方程，又称为调和方程

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (2)$$

带有稳定热源或内部无热源的稳定温度场的温度分布，不可压缩流体的稳定无旋流动及静电场的电势等均满足这类方程。

Poisson 方程的第一边值问题为

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) & (x, y) \in \Omega \\ u(x, y)|_{(x, y) \in \Gamma} = \varphi(x, y) & \Gamma = \partial\Omega \end{cases} \quad (3)$$

其中  $\Omega$  为以  $\Gamma$  为边界的有界区域， $\Gamma$  为分段光滑曲线， $\Omega \cup \Gamma$  称为定解区域， $f(x, y), \varphi(x, y)$  分别为  $\Omega, \Gamma$  上的已知连续函数。

第二类和第三类边界条件可统一表示成

$$\left. \left( \frac{\partial u}{\partial n} + \alpha u \right) \right|_{(x, y) \in \Gamma} = \varphi(x, y) \quad (4)$$

其中  $n$  为边界  $\Gamma$  的外法线方向。当  $\alpha = 0$  时为第二类边界条件， $\alpha \neq 0$  时为第三类边界条件。

在研究热传导过程，气体扩散现象及电磁场的传播等随时间变化的非定常物理问题时，常常会遇到抛物型方程。其最简单的形式为一维热传导方程

$$\frac{\partial u}{\partial t} - a \frac{\partial^2 u}{\partial x^2} = 0 \quad (a > 0) \quad (5)$$

方程 (5) 可以有两种不同类型的定解问题：

初值问题（也称为 Cauchy 问题）

$$\begin{cases} \frac{\partial u}{\partial t} - a \frac{\partial^2 u}{\partial x^2} = 0 & t > 0, \quad -\infty < x < +\infty \\ u(x, 0) = \varphi(x) & -\infty < x < +\infty \end{cases} \quad (6)$$

初边值问题

$$\begin{cases} \frac{\partial u}{\partial t} - a \frac{\partial^2 u}{\partial x^2} = 0 & 0 < t < T, \quad 0 < x < l \\ u(x, 0) = \varphi(x) \\ u(0, t) = g_1(t), \quad u(l, t) = g_2(t), \quad 0 \leq t \leq T \end{cases} \quad (7)$$

其中  $\varphi(x), g_1(t), g_2(t)$  为已知函数, 且满足连接条件

$$\varphi(0) = g_1(0), \quad \varphi(l) = g_2(0)$$

问题 (7) 中的边界条件  $u(0, t) = g_1(t), u(l, t) = g_2(t)$  称为第一类边界条件。第二类和第三类边界条件为

$$\begin{cases} \left[ \frac{\partial u}{\partial x} - \lambda_1(t)u \right]_{x=0} = g_1(t), \quad 0 \leq t \leq T \\ \left[ \frac{\partial u}{\partial x} + \lambda_2(t)u \right]_{x=l} = g_2(t), \quad 0 \leq t \leq T \end{cases} \quad (8)$$

其中  $\lambda_1(t) \geq 0, \lambda_2(t) \geq 0$ 。当  $\lambda_1(t) = \lambda_2(t) \equiv 0$  时, 为第二类边界条件, 否则称为第三类边界条件。

双曲型方程的最简单形式为一阶双曲型方程

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \quad (9)$$

物理中常见的一维振动与波动问题可用二阶波动方程

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2} \quad (10)$$

描述, 它是双曲型方程的典型形式。方程 (10) 的初值问题为

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2} & t > 0, \quad -\infty < x < +\infty \\ u(x, 0) = \varphi(x) & -\infty < x < +\infty \\ \left. \frac{\partial u}{\partial t} \right|_{t=0} = \psi(x) & -\infty < x < +\infty \end{cases} \quad (11)$$

边界条件一般也有三类, 最简单的初边值问题为

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2} & t > 0, 0 < x < l \\ u(x, 0) = \varphi(x), \quad \left. \frac{\partial u}{\partial t} \right|_{t=0} = \phi(x) & 0 \leq x \leq l \\ u(0, t) = g_1(t), \quad u(l, t) = g_2(t) & 0 \leq t \leq T \end{cases}$$

如果偏微分方程定解问题的解存在, 唯一且连续依赖于定解数据 (即出现在方程和定解条件中的已知函数), 则此定解问题是适定的。可以证明, 上面所举各种定解问题都是适定的。

## §2 偏微分方程的差分解法

差分方法又称为有限差分方法或网格法, 是求偏微分方程定解问题的数值解中应用最广泛的方法之一。它的基本思想是: 先对求解区域作网格剖分, 将自变量的连续变化区域用有限离散点 (网格点) 集代替; 将问题中出现的连续变量的函数用定义在网格点上离散变量的函数代替; 通过用网格点上函数的差商代替导数, 将含连续变量的偏微分方程定解问题化成只含有限个未知数的代数方程组 (称为差分格式)。如果差分格式有解, 且当网格无限变小时其解收敛于原微分方程定解问题的解, 则差分格式的解就作为原问题的近似解 (数值解)。因此, 用差分方法求偏微分方程定解问题一般需要解决以下问题:

- (i) 选取网格;
- (ii) 对微分方程及定解条件选择差分近似, 列出差分格式;
- (iii) 求解差分格式;
- (iv) 讨论差分格式解对于微分方程解的收敛性及误差估计。

下面我们只对偏微分方程的差分解法作一简要的介绍。

### 2.1 椭圆型方程第一边值问题的差分解法

以 Poisson 方程 (1) 为基本模型讨论第一边值问题的差分方法。

考虑 Poisson 方程的第一边值问题 (3)

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) & (x, y) \in \Omega \\ u(x, y)|_{(x, y) \in \Gamma} = \varphi(x, y) & \Gamma = \partial\Omega \end{cases}$$

取  $h, \tau$  分别为  $x$  方向和  $y$  方向的步长, 以两族平行线  $x = x_k = kh, y = y_j = j\tau$  ( $k, j = 0, \pm 1, \pm 2, \dots$ ) 将定解区域剖分成矩形网格。节点的全体记为  $R = \{(x_k, y_j) | x_k = kh, y_j = j\tau, i, j \text{ 为整数}\}$ 。定解区域内的节点称为内点, 记内点集  $R \cap \Omega$  为  $\Omega_{h\tau}$ 。边界  $\Gamma$  与网格线的交点称为边界点, 边界点全体记为  $\Gamma_{h\tau}$ 。与节点  $(x_k, y_j)$  沿  $x$  方向或  $y$  方向只差一个步长的点  $(x_{k\pm 1}, y_j)$  和  $(x_k, y_{j\pm 1})$  称为节点  $(x_k, y_j)$  的相邻节点。如果一个内点的四个相邻节点均属于  $\Omega \cup \Gamma$ , 称为正则内点, 正则内点的全体记为  $\Omega^{(1)}$ , 至少有一个相邻节点不属于  $\Omega \cup \Gamma$  的内点称为非正则内点, 非正则内点的全体记为  $\Omega^{(2)}$ 。我们的问题是要求出问题 (3) 在全体内点上的数值解。

为简便记, 记  $(k, j) = (x_k, y_j), u(k, j) = u(x_k, y_j), f_{k,j} = f(x_k, y_j)$ 。对正则内点

$(k, j) \in \Omega^{(1)}$ , 由二阶中心差商公式

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2}\bigg|_{(k,j)} &= \frac{u(k+1, j) - 2u(k, j) + u(k-1, j)}{h^2} + O(h^2) \\ \frac{\partial^2 u}{\partial y^2}\bigg|_{(k,j)} &= \frac{u(k, j+1) - 2u(k, j) + u(k, j-1)}{\tau^2} + O(\tau^2)\end{aligned}$$

Poisson 方程 (1) 在点  $(k, j)$  处可表示为

$$\begin{aligned}\frac{u(k+1, j) - 2u(k, j) + u(k-1, j)}{h^2} + \frac{u(k, j+1) - 2u(k, j) + u(k, j-1)}{\tau^2} \\ = f_{k,j} + O(h^2 + \tau^2)\end{aligned}\quad (12)$$

在式 (12) 中略去  $O(h^2 + \tau^2)$ , 即得与方程 (1) 相近似的差分方程

$$\frac{u_{k+1,j} - 2u_{k,j} + u_{k-1,j}}{h^2} + \frac{u_{k,j+1} - 2u_{k,j} + u_{k,j-1}}{\tau^2} = f_{k,j}\quad (13)$$

式 (13) 中方程的个数等于正则内点的个数, 而未知数  $u_{k,j}$  则除了包含正则内点处解  $u$  的近似值, 还包含一些非正则内点处  $u$  的近似值, 因而方程个数少于未知数个数。在非正则内点处 Poisson 方程的差分近似不能按式 (13) 给出, 需要利用边界条件得到。边界条件的处理可以有各种方案, 下面介绍较简单的两种。

(1) 直接转移。用最接近非正则内点的边界点上的  $u$  值作为该点上  $u$  值的近似, 这就是边界条件的直接转移。例如, 点  $P(k, j)$  为非正则内点, 其最接近的边界点为  $Q$  点, 则有:

$$u_{k,j} = u(Q) = \varphi(Q), \quad (k, j) \in \Omega^{(2)}$$

(ii) 线性插值。这种方案是通过用同一条网格线上与点  $P$  相邻的边界点  $R$  与内点  $T$  作线性插值得到非正则内点  $P(k, j)$  处  $u$  值的近似。由点  $R$  与  $T$  的线性插值确定  $u(P)$  的近似值为

$$u_{k,j} = \frac{h}{h+d} \varphi(R) + \frac{d}{h+d} u(T)$$

其中  $d = |RP|$ ,  $h = |PT|$ , 其截断误差为  $O(h^2)$ 。

由式 (13) 所给出的差分格式称为五点菱形格式, 实际计算时经常取  $h = \tau$ , 此时五点菱形格式可化为

$$\frac{1}{h^2} (u_{k+1,j} + u_{k-1,j} + u_{k,j+1} + u_{k,j-1} - 4u_{k,j}) = f_{k,j}\quad (14)$$

简记为

$$\frac{1}{h^2} \diamond u_{k,j} = f_{k,j}\quad (15)$$

其中  $\diamond u_{k,j} = u_{k+1,j} + u_{k-1,j} + u_{k,j+1} + u_{k,j-1} - 4u_{k,j}$ 。

求解差分方程组最常用的方法是同步迭代法, 同步迭代法是最简单的迭代方式。除边界节点外, 区域内节点的初始值是任意取定的。

例 1 用五点菱形格式求解 Laplace 方程第一边值问题

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 & (x, y) \in \Omega \\ u(x, y)|_{(x,y) \in \Gamma} = \lg[(1+x)^2 + y^2] & \Gamma = \partial\Omega \end{cases}$$

其中  $\Omega = \{(x, y) | 0 \leq x, y \leq 1\}$ 。取  $h = \tau = \frac{1}{3}$ 。

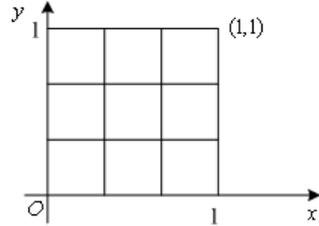


图1 网格划分图

解 节点编号为  $(k, j)$ ,  $k = 0, 1, 2, 3$ ,  $j = 0, 1, 2, 3$ 。网格中有四个内点, 均为正则内点。由五点菱形格式, 得方程组:

$$\begin{cases} \frac{1}{h^2}(u_{1,2} + u_{2,1} + u_{1,0} + u_{0,1} - 4u_{1,1}) = 0 \\ \frac{1}{h^2}(u_{2,2} + u_{3,1} + u_{2,0} + u_{1,1} - 4u_{1,2}) = 0 \\ \frac{1}{h^2}(u_{1,3} + u_{2,2} + u_{1,1} + u_{0,2} - 4u_{2,1}) = 0 \\ \frac{1}{h^2}(u_{2,3} + u_{3,2} + u_{2,1} + u_{1,2} - u_{2,2}) = 0 \end{cases} \quad (16)$$

代入边界条件  $u_{1,0}, u_{2,0}, u_{0,1}, u_{0,2}, u_{1,3}, u_{2,3}, u_{3,1}, u_{3,2}$  的值, (16) 式可以化成

$$\begin{bmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 1 & 0 & -4 & 1 \\ 0 & 1 & 1 & -4 \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{1,2} \\ u_{2,1} \\ u_{2,2} \end{bmatrix} = - \begin{bmatrix} u_{1,0} + u_{0,1} \\ u_{3,1} + u_{2,0} \\ u_{1,3} + u_{0,2} \\ u_{2,3} + u_{3,2} \end{bmatrix} \quad (17)$$

解非齐次线性方程组求得

$$u_{1,1} = 0.6348, \quad u_{1,2} = 1.06, \quad u_{2,1} = 0.7985, \quad u_{2,2} = 1.1698$$

计算的 Matlab 程序如下:

```

clc,clear
f1=@(x) 2*log(1+x);f2=@(x) log((1+x).^2+1);
f3=@(y) log(1+y.^2); f4=@(y) log(4+y.^2);
u=zeros(4);m=4;n=4;h=1/3;
u(1,1:m)=feval(f3,0:h:(m-1)*h)';
u(n,1:m)=feval(f4,0:h:(m-1)*h)';
u(1:n,1)=feval(f1,0:h:(n-1)*h);
u(1:n,m)=feval(f2,0:h:(n-1)*h);
b=-[u(2,1)+u(1,2);u(4,2)+u(3,1);u(2,4)+u(1,3);u(3,4)+u(4,3)];
a=[-4 1 1 0;1 -4 0 1;1 0 -4 1;0 1 1 -4];

```

$x=a \setminus b$

实际上，可以使用同步迭代法、异步迭代法、逐次超松弛迭代法等方法求方程组（16）的解。

同步迭代法的迭代格式为

$$u_{k,j}^{(i+1)} = \frac{1}{4} [u_{k-1,j}^{(i)} + u_{k+1,j}^{(i)} + u_{k,j-1}^{(i)} + u_{k,j+1}^{(i)}]$$

异步迭代法的迭代格式为

$$u_{k,j}^{(i+1)} = \frac{1}{4} [u_{k-1,j}^{(i+1)} + u_{k+1,j}^{(i)} + u_{k,j-1}^{(i+1)} + u_{k,j+1}^{(i)}]$$

由于在异步迭代法中有一半是用了迭代的新值，所以可以预料异步迭代法的收敛速度比同步迭代法的收敛速度要快一倍左右。

下面我们用逐次超松弛迭代法求例1的数值解。程序如下（以下所有的程序放在一个文件中）：

```
function sol=main
f1=@(x) 2*log(1+x);f2=@(x) log((1+x).^2+1);
f3=@(y) log(1+y.^2); f4=@(y) log(4+y.^2);
a=1;b=1;h=1/3;tol=0.00001;max1=1000;
sol=dirich(f1,f2,f3,f4,a,b,h,tol,max1)

%*****
function U=dirich(f1,f2,f3,f4,a,b,h,tol,max1)

% Input - f1,f2,f3,f4 are boundary functions input as strings
%       - a and b right endpoints of [0,a] and [0,b]
%       - h step size
%       - tol is the tolerance
% Output - U solution matrix;

% If f1,f2,f3 and f4 are M-file functions call U = dirich(@f1,@f2,@f3,@f4,a,b,h,tol,max1).
% if f1,f2,f3 and f4 are anonymous functions call U = dirich(f1,f2,f3,f4,a,b,h,tol,max1).

% Initialize parameters and U
n=fix(a/h)+1;
m=fix(b/h)+1;
ave=(a*(feval(f1,0)+feval(f2,0)) ...
     +b*(feval(f3,0)+feval(f4,0)))/(2*a+2*b);
U=ave*ones(n,m);

% Boundary conditions
U(1,1:m)=feval(f3,0:h:(m-1)*h);
U(n,1:m)=feval(f4,0:h:(m-1)*h);
U(1:n,1)=feval(f1,0:h:(n-1)*h);
U(1:n,m)=feval(f2,0:h:(n-1)*h);

%逐次超松弛迭代法（SOR）参数
w=4/(2+sqrt(4-(cos(pi/(n-1))+cos(pi/(m-1)))^2));

% Refine approximations and sweep operator throughout the grid
```

```

err=1;
cnt=0;
while((err>tol)&(cnt<=max1))
    err=0;
    for j=2:m-1
        for i=2:n-1
            relx=w*(U(i,j+1)+U(i,j-1)+U(i+1,j)+U(i-1,j)-4*U(i,j))/4;
            U(i,j)=U(i,j)+relx;
            err=abs(relx);
        end
    end
cnt=cnt+1;
end

```

```

U=flipud(U');
%*****

```

上述程序中的函数 `dirich(f1,f2,f3,f4,a,b,h,tol,max1)` 可以求解如下类型的问题:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, (x, y) \in \Omega = \{(x, y) : 0 \leq x \leq a, 0 \leq y \leq b\}$$

而且满足条件

$$u(x, 0) = f_1(x), \quad u(x, b) = f_2(x), \quad 0 \leq x \leq a$$

$$u(0, y) = f_3(y), \quad u(a, y) = f_4(y), \quad 0 \leq y \leq b$$

上述程序中步长  $h$  和  $a, b$  之间的关系为, 存在整数  $n$  和  $m$ , 使得  $a = nh$ ,  $b = mh$ 。

当  $h = \tau$  时, 利用点  $(k, j), (k \pm 1, j \pm 1)$  构造的差分格式

$$\frac{1}{2h^2} (u_{k+1,j+1} + u_{k+1,j-1} + u_{k-1,j+1} + u_{k-1,j-1} - 4u_{k,j}) = f_{k,j} \quad (18)$$

称为五点矩形格式, 简记为

$$\frac{1}{2h^2} \square u_{k,j} = f_{k,j} \quad (19)$$

其中  $\square u_{k,j} = u_{k+1,j+1} + u_{k+1,j-1} + u_{k-1,j+1} + u_{k-1,j-1} - 4u_{k,j}$ 。

## 2.2 抛物型方程的差分解法

以一维热传导方程 (5)

$$\frac{\partial u}{\partial t} - a \frac{\partial^2 u}{\partial t^2} = 0 \quad (a > 0)$$

为基本模型讨论适用于抛物型方程定解问题的几种差分格式。

首先对  $xt$  平面进行网格剖分。分别取  $h, \tau$  为  $x$  方向与  $t$  方向的步长, 用两族平行直线  $x = x_k = kh (k = 0, \pm 1, \pm 2, \dots)$ ,  $t = t_j = j\tau (j = 0, 1, 2, \dots)$ , 将  $xt$  平面剖分成矩形网格, 节点为  $(x_k, t_j) (k = 0, \pm 1, \pm 2, \dots, j = 0, 1, 2, \dots)$ 。为简便起见, 记  $(k, j) = (x_k, y_j)$ ,  $u(k, j) = u(x_k, y_j)$ ,  $\varphi_k = \varphi(x_k)$ ,  $g_{1j} = g_1(t_j)$ ,  $g_{2j} = g_2(t_j)$ ,  $\lambda_{1j} = \lambda_1(t_j)$ ,  $\lambda_{2j} = \lambda_2(t_j)$ 。

### 2.2.1 微分方程的差分近似

在网格内点  $(k, j)$  处, 对  $\frac{\partial u}{\partial t}$  分别采用向前、向后及中心差商公式, 对  $\frac{\partial^2 u}{\partial x^2}$  采用二阶中心差商公式, 一维热传导方程 (5) 可分别表示为

$$\frac{u(k, j+1) - u(k, j)}{\tau} - a \frac{u(k+1, j) - 2u(k, j) + u(k-1, j)}{h^2} = O(\tau + h^2)$$

$$\frac{u(k, j) - u(k, j-1)}{\tau} - a \frac{u(k+1, j) - 2u(k, j) + u(k-1, j)}{h^2} = O(\tau + h^2)$$

$$\frac{u(k, j+1) - u(k, j-1)}{2\tau} - a \frac{u(k+1, j) - 2u(k, j) + u(k-1, j)}{h^2} = O(\tau + h^2)$$

由此得到一维热传导方程的不同的差分近似

$$\frac{u_{k,j+1} - u_{k,j}}{\tau} - a \frac{u_{k+1,j} - 2u_{k,j} + u_{k-1,j}}{h^2} = 0 \quad (20)$$

$$\frac{u_{k,j} - u_{k,j-1}}{\tau} - a \frac{u_{k+1,j} - 2u_{k,j} + u_{k-1,j}}{h^2} = 0 \quad (21)$$

$$\frac{u_{k,j+1} - u_{k,j-1}}{2\tau} - a \frac{u_{k+1,j} - 2u_{k,j} + u_{k-1,j}}{h^2} = 0 \quad (22)$$

### 2.2.2 初、边值条件的处理

为用差分方程求解定解问题 (6), (7) 等, 还需对定解条件进行离散化。

对初始条件及第一类边界条件, 可直接得到

$$u_{k,0} = u(x_k, 0) = \varphi_k \quad (k = 0, \pm 1, \dots \text{或 } k = 0, 1, \dots, n) \quad (23)$$

$$u_{0,j} = u(0, t_j) = g_{1j} \quad (j = 0, 1, \dots, m-1) \quad (24)$$

$$u_{n,j} = u(l, t_j) = g_{2j}$$

其中  $n = \frac{l}{h}, m = \frac{T}{\tau}$ 。

对第二、三类边界条件则需用差商近似。下面介绍两种较简单的处理方法。

(i) 在左边界  $(x = 0)$  处用向前差商近似偏导数  $\frac{\partial u}{\partial x}$ , 在右边界  $(x = l)$  处用向后差

商近似偏导数  $\frac{\partial u}{\partial x}$ , 即

$$\left. \frac{\partial u}{\partial x} \right|_{(0,j)} = \frac{u(1, j) - u(0, j)}{h} + O(h)$$

$$\left. \frac{\partial u}{\partial x} \right|_{(n,j)} = \frac{u(n, j) - u(n-1, j)}{h} + O(h) \quad (j = 0, 1, \dots, m)$$

即得边界条件 (8) 的差分近似为

$$\begin{cases} \frac{u_{1,j} - u_{0,j}}{h} - \lambda_{1j} u_{0,j} = g_{1j} \\ \frac{u_{n,j} - u_{n-1,j}}{h} + \lambda_{2j} u_{n,j} = g_{2j} \end{cases} \quad (j = 0, 1, \dots, m) \quad (25)$$

(ii) 用中心差商近似  $\frac{\partial u}{\partial x}$ , 即

$$\begin{aligned} \left. \frac{\partial u}{\partial x} \right|_{(0,j)} &= \frac{u(1,j) - u(-1,j)}{2h} + O(h^2) \\ \left. \frac{\partial u}{\partial x} \right|_{(n,j)} &= \frac{u(n+1,j) - u(n-1,j)}{2h} + O(h^2) \end{aligned} \quad (j = 0, 1, \dots, m)$$

则得边界条件的差分近似为

$$\begin{cases} \frac{u_{1,j} - u_{-1,j}}{2h} - \lambda_{1j} u_{0,j} = g_{1j} \\ \frac{u_{n+1,j} - u_{n-1,j}}{2h} + \lambda_{2j} u_{n,j} = g_{2j} \end{cases} \quad (j = 0, 1, \dots, m) \quad (26)$$

这样处理边界条件, 误差的阶数提高了, 但式 (26) 中出现定解区域外的节点  $(-1, j)$  和  $(n+1, j)$ , 这就需要将解拓展到定解区域外。可以通过用内节点上的  $u$  值插值求出  $u_{-1,j}$  和  $u_{n+1,j}$ , 也可以假定热传导方程 (5) 在边界上也成立, 将差分方程扩展到边界节点上, 由此消去  $u_{-1,j}$  和  $u_{n+1,j}$ 。

### 2.2.3 几种常用的差分格式

下面我们以热传导方程的初边值问题 (7) 为例给出几种常用的差分格式。

#### (i) 古典显式格式

为便于计算, 令  $r = \frac{a\tau}{h^2}$ , 式 (20) 改写成以下形式

$$u_{k,j+1} = ru_{k+1,j} + (1-2r)u_{k,j} + ru_{k-1,j}$$

将式 (20) 与 (23), (24) 结合, 我们得到求解问题 (7) 的一种差分格式

$$\begin{cases} u_{k,j+1} = ru_{k+1,j} + (1-2r)u_{k,j} + ru_{k-1,j} & (k = 1, 2, \dots, n-1, j = 0, 1, \dots, m-1) \\ u_{k,0} = \varphi_k & (k = 1, 2, \dots, n) \\ u_{0,j} = g_{1j}, \quad u_{n,j} = g_{2j} & (j = 1, 2, \dots, m) \end{cases} \quad (27)$$

由于第 0 层 ( $j=0$ ) 上节点处的  $u$  值已知 ( $u_{k,0} = \varphi_k$ ), 由式 (25) 即可算出  $u$  在第一层 ( $j=1$ ) 上节点处的近似值  $u_{k,1}$ 。重复使用式 (25), 可以逐层计算出各层节点的近似值, 因此此差分格式称为古典显示格式。又因式中只出现相邻两个时间层的节点, 故此式是二层显式格式。

#### (ii) 古典隐式格式

将 (21) 整理并与式 (23), (24) 联立, 得差分格式如下

$$\begin{cases} u_{k,j+1} = u_{k,j} + r(u_{k+1,j+1} - 2u_{k,j+1} + u_{k-1,j+1}) & (k=1,2,\dots,n-1, j=0,1,\dots,m-1) \\ u_{k,0} = \varphi_k & (k=0,1,\dots,n) \\ u_{0,j} = g_{1j}, \quad u_{n,j} = g_{2j} & (j=0,1,\dots,m) \end{cases} \quad (28)$$

其中  $r = \frac{a\tau}{h^2}$ 。虽然第 0 层上的  $u$  值仍为已知，但不能由式 (28) 直接计算以上各层节点上的值  $u_{k,j}$ ，必须通过解下列线性方程组

$$\begin{bmatrix} 1+2r & -r & & & \\ -r & 1+2r & -r & & \\ & \ddots & \ddots & \ddots & \\ & & -r & 1+2r & -r \\ & & & -r & 1+2r \end{bmatrix} \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ u_{n-2,j+1} \\ u_{n-1,j+1} \end{bmatrix} = \begin{bmatrix} u_{1,j} + rg_{1j+1} \\ u_{2,j} \\ \vdots \\ u_{n-2,j} \\ u_{n-1,j} + rg_{2j+1} \end{bmatrix}$$

才能由  $u_{k,j}$  计算  $u_{k,j+1}$ ，故此差分方程称为古典隐式格式。此方程组是三对角方程组，且系数矩阵严格对角占优，故解存在唯一。

(iii) 杜福特—弗兰克尔 (DoFort—Frankel) 格式

DoFort—Frankel 格式是三层显式格式，它是由式

$$\frac{u_{k,j+1} - u_{k,j-1}}{2\tau} - a \frac{u_{k+1,j} - u_{k,j+1} - u_{k,j-1} + u_{k-1,j}}{h^2} = 0$$

与初始条件及第一类边界条件结合得到的。具体形式如下：

$$\begin{cases} u_{k,j+1} = \frac{2r}{1+2r}(u_{k+1,j} + u_{k-1,j}) + \frac{1-2r}{1+2r}u_{k,j-1}, & (k=1,2,\dots,n-1, j=1,2,\dots,m-1) \\ u_{k,0} = \varphi_k & (k=0,1,\dots,n) \\ u_{0,j} = g_{1j}, \quad u_{n,j} = g_{2j} & (j=0,1,\dots,m) \end{cases} \quad (29)$$

用这种格式求解时，除了第 0 层上的值  $u_{k,0}$  由初值条件得到，必须先用二层格式求出第 1 层上的值  $u_{k,1}$ ，然后再按格式 (29) 逐层计算  $u_{k,j} (j=2,3,\dots,m)$ 。

例 2 用古典显示格式求初边值问题。

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0 & 0 < t < 3, \quad 0 < x < 3 \\ u(x,0) = x^2 & 0 \leq x \leq 3 \\ u(0,t) = 0, \quad u(3,t) = 9 & 0 \leq t \leq 3 \end{cases}$$

的数值解，取  $h=1, \tau=0.5$ 。

解：这里： $a=1, r = \frac{a\tau}{h^2} = 0.5, \varphi(x) = x^2, g_1(t) = 0, g_2(t) = 9$ 。

由格式

$$\begin{cases} u_{k,j+1} = ru_{k+1,j} + (1-2r)u_{k,j} + ru_{k-1,j}, & (k=1,2,\dots,n-1; j=0,1,\dots,m-1) \\ u_{k,0} = \varphi_k, & (k=0,1,\dots,n) \\ u_{0,j} = g_{1j}, u_{n,j} = g_{2j}, & (j=0,1,\dots,m) \end{cases}$$

可得到

$$\begin{cases} u_{k,j+1} = 0.5(u_{k+1,j} + u_{k-1,j}) & (k=1,2, j=0,1,\dots,5) \\ u_{k,0} = x_k^2 = k^2 & (k=0,1,2,3) \\ u_{0,j} = 0, u_{3,j} = 9 & (j=0,1,\dots,6) \end{cases}$$

将初值  $u_{k,0}$  代入上式, 即可算出:

$$u_{1,1} = 0.5 \times (u_{2,0} + u_{0,0}) = 0.5 \times (4 + 0) = 2$$

$$u_{2,1} = 0.5 \times (u_{3,0} + u_{1,0}) = 0.5 \times (9 + 1) = 5$$

将边界条件  $u_{0,1} = 0, u_{3,1} = 9$  及上述结果代入又可求得

$$u_{1,2} = 2.5, u_{2,2} = 5.5$$

如此逐层计算, 得全部节点上的数值解为

$$u_{1,3} = 2.75, u_{2,3} = 5.75, \dots$$

#### 2.2.4 求解抛物形方程的 Matlab 程序

(1) 前向差分法求解热传导方程的 MATLAB 程序

设  $u(x,0) = f(x)$ , 其中,  $0 \leq x \leq a$ , 而且,  $u(0,t) = c_1, u(a,t) = c_2$ , 其中  $0 \leq t \leq b$ ,

求解区间  $R = \{(x,t) : 0 \leq x \leq a, 0 \leq t \leq b\}$  内  $u_t(x,t) = c^2 u_{xx}(x,t)$  的近似解。

```
%*****
```

```
function U=forwdif(f,c1,c2,a,b,c,n,m);
```

```
%Input - f=u(x,0) as a string 'f'
```

```
% - c1=u(0,t) and c2=u(a,t)
```

```
% - a and b right endpoints of [0,a] and [0,b]
```

```
% - c the constant in the heat equation
```

```
% - n and m number of grid points over [0,a] and [0,b]
```

```
%Output - U solution matrix;
```

```
%Initialize parameters and U
```

```
h=a/(n-1);k=b/(m-1);
```

```
r=c^2*k/h^2;s=1-2*r;
```

```
U=zeros(n,m);
```

```
%Boundary conditions
```

```
U(1,1:m)=c1;U(n,1:m)=c2;
```

```
%Generate first row
```

```
U(2:n-1,1)=feval(f,h:h:(n-2)*h)';
```

```
%Generate remaining rows of U
```

```
for j=2:m
```

```

    for i=2:n-1
        U(i,j)=s*U(i,j-1)+r*(U(i-1,j-1)+U(i+1,j-1));
    end
end

```

```
U=U';
```

```
%*****
```

利用上述函数，我们可以编写如下程序计算例 2 的数值解。

```
c=1;a=3;b=3;c1=0;c2=9;n=4;m=7;
```

```
f=@(x) x.^2;
```

```
sol=forwdif(f,c1,c2,a,b,c,n,m)
```

## (2) Crank-Nicholson 求解热传导方程的 MATLAB 程序

设  $u(x,0) = f(x)$ ，其中  $0 \leq x \leq a$ ，而且  $u(0,t) = c_1, u(a,t) = c_2$ ，其中  $0 \leq t \leq b$ ，

求解区间  $R = \{(x,t) : 0 \leq x \leq a, 0 \leq t \leq b\}$  内  $u_t(x,t) = c^2 u_{xx}(x,t)$  的近似解。

```
%*****
```

```
function U=crnich(f,c1,c2,a,b,c,n,m)
```

```
%Input - f=u(x,0)
```

```
% - c1=u(0,t) and c2=u(a,t)
```

```
% - a and b right endpoints of [0,a] and [0,b]
```

```
% - c the constant in the heat equation
```

```
% - n and m number of grid points over [0,a] and [0,b]
```

```
%Output - U solution matrix;
```

```
%If f is an M-file function call U=crnich(@f,c1,c2,a,b,c,n,m).
```

```
%If f is an anonymous function call U=crnich(f,c1,c2,a,b,c,n,m).
```

```
%Initialize parameters and U
```

```
h=a/(n-1);k=b/(m-1);
```

```
r=c^2*k/h^2;s1=2+2/r;s2=2/r-2;
```

```
U=zeros(n,m);
```

```
%Boundary conditions
```

```
U(1,1:m)=c1;U(n,1:m)=c2;
```

```
%Generate first row
```

```
U(2:n-1,1)=f(h:h:(n-2)*h)';
```

```
%Form the diagonal and off-diagonal elements of A and
```

```
%the constant vector B and solve tridiagonal system AX=B
```

```
Vd(1,1:n)=s1*ones(1,n);Vd(1)=1;Vd(n)=1;
```

```
Va=-ones(1,n-1);Va(n-1)=0;
```

```
Vc=-ones(1,n-1);Vc(1)=0;
```

```
Vb(1)=c1;Vb(n)=c2;
```

```
for j=2:m
```

```
    for i=2:n-1
```

```
        Vb(i)=U(i-1,j-1)+U(i+1,j-1)+s2*U(i,j-1);
```

```
    end
```

```
    X=trisys(Va,Vd,Vc,Vb);
```

```

    U(1:n, j)=X';
end
U=U';

function X=trisy(A,D,C,B)
%Input   - A is the sub diagonal of the coefficient matrix
%         - D is the main diagonal of the coefficient matrix
%         - C is the super diagonal of the coefficient matrix
%         - B is the constant vector of the linear system
%Output  - X is the solution vector

N=length(B);
for k=2:N
    mult=A(k-1)/D(k-1);
    D(k)=D(k)-mult*C(k-1);
    B(k)=B(k)-mult*B(k-1);
end

X(N)=B(N)/D(N);
for k=N-1:-1:1
    X(k)=(B(k)-C(k)*X(k+1))/D(k);
end
%*****

```

### 2.3 双曲型方程的差分解法

#### 2.3.1 一阶双曲型方程的差分格式

考虑一阶双曲型方程的初值问题：

$$\begin{cases} \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 & t > 0, -\infty < x < +\infty \\ u(x, 0) = \varphi(x) & -\infty < x < +\infty \end{cases} \quad (30)$$

将  $x-t$  平面剖分成矩形网格，取  $x$  方向步长为  $h$ ， $t$  方向步长为  $\tau$ ，网格线为  $x = x_k = kh (k = 0, \pm 1, \pm 2, \dots)$ ， $t = t_j = j\tau (j = 0, 1, 2, \dots)$

为简便，记

$$(k, j) = (x_k, t_j), \quad u(k, j) = u(x_k, t_j), \quad \varphi_k = \varphi(x_k)$$

以不同的差商近似偏导数，可以得到方程的不同的差分近似

$$\frac{u_{k,j+1} - u_{k,j}}{\tau} + a \frac{u_{k+1,j} - u_{k,j}}{h} = 0 \quad (31)$$

$$\frac{u_{k,j+1} - u_{k,j}}{\tau} + a \frac{u_{k,j} - u_{k-1,j}}{h} = 0 \quad (32)$$

$$\frac{u_{k,j+1} - u_{k,j}}{\tau} + a \frac{u_{k+1,j} - u_{k-1,j}}{2h} = 0 \quad (33)$$

截断误差分别为  $O(\tau + h)$ ， $O(\tau + h)$  与  $O(\tau + h^2)$ 。结合离散化的初始条件，可以得到几种简单的差分格式

$$\begin{cases} u_{k,j+1} = u_{k,j} - ar(u_{k+1,j} - u_{k,j}) \\ u_{k,0} = \varphi_k \end{cases} \quad (k = 0, \pm 1, \pm 2, \dots, j = 0, 1, 2, \dots) \quad (34)$$

$$\begin{cases} u_{k,j+1} = u_{k,j} - ar(u_{k,j} - u_{k-1,j}) \\ u_{k,0} = \varphi_k \end{cases} \quad (k = 0, \pm 1, \pm 2, \dots, j = 0, 1, 2, \dots) \quad (35)$$

$$\begin{cases} u_{k,j+1} = u_{k,j} - \frac{ar}{2}(u_{k+1,j} - u_{k-1,j}) \\ u_{k,0} = \varphi_k \end{cases} \quad (k = 0, \pm 1, \pm 2, \dots, j = 0, 1, 2, \dots) \quad (36)$$

其中:  $r = \frac{\tau}{h}$ 。如果已知第  $j$  层节点上的值  $u_{k,j}$ , 按上面三种格式就可求出第  $j+1$  层上的值  $u_{k,j+1}$ 。因此, 这三种格式都是显式格式。

如果对  $\frac{\partial u}{\partial t}$  采用向后差商,  $\frac{\partial u}{\partial x}$  采用向前差商, 则方程可化成

$$\frac{u(k, j) - u(k, j-1)}{\tau} + a \frac{u(k+1, j) - u(k, j)}{h} + O(\tau + h) = 0 \quad (37)$$

相应的差分格式为:

$$\begin{cases} u_{k,j+1} = u_{k,j} - ar(u_{k+1,j+1} - u_{k,j+1}) \\ u_{k,0} = \varphi_k \end{cases} \quad (k = 0, \pm 1, \pm 2, \dots, j = 0, 1, 2, \dots) \quad (38)$$

此差分格式是一种隐式格式, 必须通过解方程组才能由第  $j$  层节点上的值  $u_{k,j}$ , 求出第  $j+1$  层节点上的值  $u_{k,j+1}$ 。

例 3 对初值问题: 
$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 & t > 0, -\infty < x < +\infty \\ u(x, 0) = \varphi(x) & -\infty < x < +\infty \end{cases}$$

其中  $\varphi(x) = \begin{cases} 1 & x < 0 \\ \frac{1}{2} & x = 0 \\ 0 & x > 0 \end{cases}$ , 用差分格式求其数值解  $u_{k,j}$  ( $j = 1, 2, 3, 4$ ), 取  $r = \frac{\tau}{h} = \frac{1}{2}$ 。

解: 记  $x_k = kh$  ( $k = 0, \pm 1, \pm 2, \dots$ ), 由初始条件:

$$\varphi_k = \varphi(x_k) = \begin{cases} 1 & k = -1, -2, \dots \\ \frac{1}{2} & k = 0 \\ 0 & k = 1, 2, \dots \end{cases}$$

按差分格式:

$$\begin{cases} u_{k,j+1} = u_{k,j} - ar(u_{k+1,j} - u_{k,j}) \\ u_{k,0} = \varphi_k \end{cases} \quad (k = 0, \pm 1, \pm 2, \dots, j = 0, 1, 2, \dots)$$

计算公式为： $u_{k,j+1} = \frac{3}{2}u_{k,j} - \frac{1}{2}u_{k+1,j}$ 。计算结果略。

如果用差分格式：

$$\begin{cases} u_{k,j+1} = u_{k,j} - ar(u_{k,j} - u_{k-1,j}) \\ u_{k,0} = \varphi_k \end{cases} \quad (k = 0, \pm 1, \pm 2, \dots, j = 0, 1, 2, \dots)$$

求解，计算公式为：

$$u_{k,j+1} = \frac{1}{2}(u_{k,j} + u_{k-1,j})$$

计算结果略。与准确解：

$$u(x,t) = \begin{cases} 1 & x < t \\ \frac{1}{2} & x = t \\ 0 & x > 0 \end{cases}$$

比较知，按前一个差分格式所求得的数值解不收敛到初值问题的解，而后一个差分格式的解收敛到原问题的解。

### 2.3.2 波动方程的差分格式

对二阶波动方程 (10)

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

如果令  $v_1 = \frac{\partial u}{\partial t}, v_2 = \frac{\partial u}{\partial x}$ ，则方程 (10) 可化成一阶线性双曲型方程组

$$\begin{cases} \frac{\partial v_1}{\partial t} = a^2 \frac{\partial v_2}{\partial x} \\ \frac{\partial v_2}{\partial t} = \frac{\partial v_1}{\partial x} \end{cases} \quad (39)$$

记  $v = (v_1, v_2)^T$ ，则方程组 (39) 可表成矩阵形式

$$\frac{\partial v}{\partial t} = \begin{bmatrix} 0 & a^2 \\ 1 & 0 \end{bmatrix} \frac{\partial v}{\partial x} = A \frac{\partial v}{\partial x} \quad (40)$$

矩阵  $A$  有两个不同的特征值  $\lambda = \pm a$ ，故存在非奇异矩阵  $P$ ，使得

$$PAP^{-1} = \begin{bmatrix} a & 0 \\ 0 & -a \end{bmatrix} = \Lambda$$

作变换  $w = Pv = (w_1, w_2)^T$ ，方程组 (31) 可化成

$$\frac{\partial w}{\partial t} = \Lambda \frac{\partial w}{\partial x} \quad (41)$$

方程组 (41) 由两个独立的一阶双曲型方程联立而成。因此我们可以把二阶波动方程化成一阶方程组进行求解。

下面给出如下波动方程和边界条件的差分格式

$$u_{tt}(x,t) = c^2 u_{xx}(x,t) \quad 0 < x < a, \quad 0 < t < b \quad (42)$$

$$\begin{cases} u(0,t) = 0, u(a,t) = 0 & 0 \leq t \leq b \\ u(x,0) = f(x) & 0 \leq x \leq a \\ u_t(x,0) = g(x) & 0 < x < a \end{cases} \quad (43)$$

将矩形  $R = \{(x,t) : 0 \leq x \leq a, 0 \leq t \leq b\}$  划分成  $(n-1) \times (m-1)$  个小矩形，长宽分别为： $\Delta x = h$ ， $\Delta t = k$ ，形成一个网格。

把方程 (42) 离散化成差分方程

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} = c^2 \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \quad (44)$$

为方便起见，可将  $r = ck/h$  代入上式，可得：

$$u_{i,j+1} - 2u_{i,j} + u_{i,j-1} = r^2(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) \quad (45)$$

设行  $j$  和  $j-1$  的近似值已知，可用上式求网格的行  $j+1$

$$u_{i,j+1} = (2 - 2r^2)u_{i,j} + r^2(u_{i+1,j} + u_{i-1,j}) - u_{i,j-1} \quad (46)$$

用上式时，必须注意，如果计算的某个阶段带来的误差最终会越来越小，则方法是稳定的。为了保证上式的稳定性，必然使  $r = ck/h \leq 1$ 。还存在其他一些差分方程方法，称为隐格式法，它们更难实现，但对  $r$  无限制。

### 2.3.3 差分方法求解波动方程的 MATLAB 程序

求解区间  $R = \{(x,t) : 0 \leq x \leq a, 0 \leq t \leq b\}$ ，以 (43) 为初边值条件的波动方程的差分方法程序。

```

%*****
function U = finedif(f,g,a,b,c,n,m)
%Input - f=u(x,0) as a string 'f'
%       - g=ut(x,0) as a string 'g'
%       - a and b right endpoints of [0,a] and [0,b]
%       - c the constant in the wave equation
%       - n and m number of grid points over [0,a] and [0,b]
%Output - U solution matrix;

% If f and g are M-file functions call U = finedif(@f,@g,a,b,c,n,m).
% if f and g are anonymous functions call U = finedif(f,g,a,b,c,n,m).

%Initialize parameters and U
h = a/(n-1);k = b/(m-1);r = c*k/h;
r2=r^2;r22=r^2/2;
s1 = 1 - r^2;s2 = 2 - 2*r^2;
U = zeros(n,m);

%Comput first and second rows
for i=2:n-1
    U(i,1)=feval(f,h*(i-1));
    U(i,2)=s1*feval(f,h*(i-1))+k*feval(g,h*(i-1)) ...
        +r22*(feval(f,h*i)+feval(f,h*(i-2)));
end

```

```

%Compute remaining rows of U
for j=3:m,
    for i=2:(n-1),
        U(i,j) = s2*U(i,j-1)+r2*(U(i-1,j-1)+U(i+1,j-1))-U(i,j-2);
    end
end
U=U';
%*****

```

### § 3 一维状态空间的偏微分方程的 MATLAB 解法

Matlab 有专门的指令 `pdepe` 解一维的抛物型或椭圆型方程组的初边值问题, 使用指令 `pdepe` 要写三个辅助函数文件, 即微分方程函数文件, 初始条件函数文件, 边界条件函数文件。

#### 3.1 pdepe 的用法

Matlab 提供了一个指令 `pdepe`, 用以解以下的一维偏微分方程式

$$c(x,t,u, \frac{\partial u}{\partial x}) \frac{\partial u}{\partial t} = x^{-m} \frac{\partial}{\partial x} (x^m f(x,t,u, \frac{\partial u}{\partial x})) + s(x,t,u, \frac{\partial u}{\partial x}) \quad (47)$$

其中  $u$  可以为向量, 时间介于  $t_0 \leq t \leq t_f$  之间, 而位置  $x$  则介于有限区域  $[a,b]$  之间。  $m$  值表示问题的对称性, 其可为 0, 1 或 2, 分别表示平板(slab), 圆柱(cylindrical)或球体(spherical)的情形。式中  $f(x,t,u, \partial u/\partial x)$  为通量项 (flux), 而  $s(x,t,u, \partial u/\partial x)$  为源项 (source)。  $c(x,t,u, \partial u/\partial x)$  为偏微分方程的系数对角矩阵。若某一对角线元素为 0, 则表示该偏微分方程为椭圆型偏微分方程, 若为正值(不为 0), 则为抛物型偏微分方程。请注意  $c$  的对角线元素一定不全为 0。偏微分方程初始值可表示为

$$u(x,t_0) = v_0(x) \quad (48)$$

而边界条件为

$$p_a(x,t,u) + q_a(x,t) f(x,t,u, \partial u/\partial x) = 0, \quad x = a, \quad t_0 \leq t \leq t_f \quad (49)$$

$$p_b(x,t,u) + q_b(x,t) f(x,t,u, \partial u/\partial x) = 0, \quad x = b, \quad t_0 \leq t \leq t_f \quad (50)$$

解上述初边值条件的偏微分方程的 MATLAB 命令 `pdepe` 的用法如下:

$$sol = pdepe(m, pdefun, icfun, bcfun, xmesh, tspan, options)$$

其中:

$m$  为问题之对称参数;

$xmesh$  为空间变量  $x$  的网格点(mesh)位置向量, 即  $xmesh = [x_0, x_1, \dots, x_N]$ , 其中  $x_0 = a$  (起点),  $x_N = b$  (终点)。

$tspan$  为时间变量  $t$  的向量, 即  $tspan = [t_0, t_1, \dots, t_M]$ , 其中  $t_0$  为起始时间,  $t_M = t_f$  为终点时间。

$pdefun$  为使用者提供的 `pde` 函数文件。其函数格式如下:

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

亦即, 使用者仅需提供偏微分方程中的系数向量。  $c$ ,  $f$  和  $s$  均为列(column)向量, 而向量  $c$  即为矩阵  $c$  的对角线元素。

$icfun$  提供解  $u$  的起始值, 其格式为  $u = icfun(x)$ , 值得注意的是  $u$  为列向量。

*bcfun* 为用户提供的边界条件函数，格式如下：

$$[pa, qa, pb, qb] = bcfun(xa, ua, xb, ub, t)$$

其中 *ua* 和 *ub* 分别表示左边界 ( $xa = a$ ) 和右边界 ( $xb = b$ ) *u* 的近似解。输出变量中，*pa* 和 *qa* 分别表示左边界  $p_a$  和  $q_b$  的列向量，而 *pb* 和 *qb* 则为右边界  $p_b$  和  $q_b$  的列向量。

*sol* 为输出的解，*sol* 为三维矩阵， $sol(j, k, i)$  是 *u* 的第 *i* 个分量在  $t = tspan(j)$ ， $x = xmesh(k)$  的值。

*options* 为求解器的相关解法参数。详细说明参见 *odeset* 的使用方法。

注：

1. MATLAB PDE 求解器 *pdepe* 的算法，主要是将原来的椭圆型和抛物型偏微分方程转化为一组常微分方程。此转换的过程是基于使用者所指定的 *mesh* 点，以二阶空间离散化 (spatial discretization) 技术为之 (Keel and Berzins, 1990)，然后以 *ode15s* 的指令求解。采用 *ode15s* 的 *ode* 解法，主要是因为离散化的过程中，椭圆型偏微分方程被转化为一组代数方程，而抛物型的偏微分方程则被转化为一组联立的微分方程。因而，原偏微分方程被离散化后，变成一组同时伴有微分方程与代数方程的微分代数方程组，故以 *ode15s* 便可顺利求解。

2. *x* 的取点 (*mesh*) 位置对解的精确度影响很大，若 *pdepe* 求解器给出 “...has difficulty finding consistent initial condition” 的讯息时，使用者可进一步将 *mesh* 点取密一点，即增加 *mesh* 点数。另外，若状态 *u* 在某些特定点上有较快速的变动时，亦需将此处的点取密集些，以增加精确度。值得注意的是 *pdepe* 并不会自动做 *xmesh* 的自动取点，使用者必须观察解的特性，自行作取点的操作。一般而言，所取的点数至少需大于 3 以上。

3. *tspan* 的选取主要是基于使用者对那些特定时间的状态有兴趣而选定。而间距 (*step size*) 的控制由程序自动完成。

4. 若要获得特定位置及时间下的解，可配合以 *pdeval* 命令。使用格式如下：

$$[uout, duoutdx] = pdeval(m, xmesh, ui, xout)$$

其中

*m* 代表问题的对称性。*m* = 0 表示平板；*m* = 1 表示圆柱体；*m* = 2 表示球体。其意义同 *pdepe* 中的自变量 *m*。

*xmesh* 为使用者在 *pdepe* 中所指定的输出点位置向量。 $xmesh = [x_0, x_1, \dots, x_N]$ 。

*ui* 即  $sol(j, :, i)$ 。也就是说其为 *pdepe* 输出中第 *i* 个输出 *ui* 在各点位置 *xmesh* 处，时间固定为  $t_j = tspan(j)$  下的解。

*xout* 为所欲内插输出点位置向量。此为使用者重新指定的位置向量。

*uout* 为基于所指定位置 *xout*，固定时间  $t_f$  下的相对应输出。

*duoutdx* 为相对应的  $du/dx$  输出值。

以下将以数个例子，详细说明 *pdepe* 的用法。

### 3.2 求解一维偏微分方程

#### 例 4 试解偏微分方程

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

其中  $0 \leq x \leq 1$ ，且满足以下初边值条件

(i)初值条件:  $u(x,0) = \sin(\pi x)$

(ii)边界条件

$$\text{BC1: } u(0,t) = 0; \quad \text{BC2: } \pi e^{-t} + \frac{\partial u(1,t)}{\partial x} = 0。$$

注: 本问题的解析解为  $u(x,t) = e^{-t} \sin(\pi x)$

解 下面将叙述求解的步骤与过程。当完成以下各步骤后, 可进一步将其汇总为主程序 `ex20_1.m`, 然后求解。

步骤 1 将欲求解的偏微分方程改写成如下的标准形式。

$$\pi^2 \frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial x} \right) + 0$$

此即

$$c(x,t,u,\partial u/\partial x) = \pi^2$$

$$f(x,t,u,\partial u/\partial x) = \frac{\partial u}{\partial x}$$

$$s(x,t,u,\partial u/\partial x) = 0$$

和  $m = 0$ 。

步骤 2 编写偏微分方程的系数向量函数。

```
function [c,f,s]=ex20_1pdefun(x,t,u,dudx)
```

```
c=pi^2;f=dudx;s=0;
```

步骤 3 编写起始值条件。

```
function u0=ex20_1ic(x)
```

```
u0=sin(pi*x);
```

步骤 4 编写边界条件。在编写之前, 先将边界条件改写成标准形式, 如式 (49),

(50), 找出相对应的函数  $p_a(\cdot), q_a(\cdot)$  和  $p_b(\cdot), q_b(\cdot)$ , 然后写出 MATLAB 的边界条件函数, 例如, 原边界条件可写成

$$\text{BC1: } u(0,t) + 0 \cdot \frac{\partial}{\partial x}(0,t) = 0, \quad x = 0$$

$$\text{BC2: } \pi e^{-t} + 1 \cdot \frac{\partial u}{\partial x}(1,t) = 0, \quad x = 1$$

即

$$p_a = u(0,t), \quad q_a = 0$$

和

$$p_b = \pi e^{-t}, \quad q_b = 1$$

因而, 边界条件函数可编写成

```
function [pa,qa,pb,qb]=ex20_1bc(xa,ua,xb,ub,t)
```

```
pa=ua;qa=0;
```

```
pb=pi*exp(-t);qb=1;
```

步骤 5 取点。例如

```
x=linspace(0,1,20); %x 取 20 点
```

```
t=linspace(0,2,5); %时间取 5 点输出
```

步骤 6 利用 `pdepe` 求解。

```
m=0; %依步骤 1 之结果
```

```

sol=pdepe(m,@ex20_1pdefun,@ex20_1ic,@ex20_1bc,x,t);
%这里 sol 实际上是二维矩阵
步骤 7 显示结果。
surf(x,t,sol)
title('pde 数值解'),xlabel('位置'),ylabel('时间'),zlabel('u')
若要显示特定点上的解,可进一步指定 x 或 t 的位置,以便绘图。例如,欲了解时
间为 2(终点)时,各位置下的解,可输入以下指令(利用 pdeval 指令):
figure(2); %绘成图 2
M=length(t); %取终点时间的下标
xout=linspace(0,1,100); %输出点位置
[uout,dudx]=pdeval(m,x,sol(M,:),xout);
plot(xout,uout); %绘图
title('时间为 2 时,各位置下的解'),xlabel('x'),ylabel('u')

```

综合以上各步骤,可写成一个程序求解例 4。其程序(全部程序放在一个文件中)如下:

```

function sol=ex20_1
%*****
%求解一维热传导偏微分方程的一个综合函数程序
%*****
m=0;
x=linspace(0,1,20); %xmesh
t=linspace(0,2,20); %tspan
%*****
%以 pdepe 求解
%*****
sol=pdepe(m,@ex20_1pdefun,@ex20_1ic,@ex20_1bc,x,t);
%这里 sol 实际上是二维矩阵
%*****
%绘图输出
%*****
figure(1),surf(x,t,sol)
title('pde 数值解'),xlabel('位置 x'),ylabel('时间 t'),zlabel('数值解 u')
%*****
%与解析解做比较
%*****
figure(2)
surf(x,t,exp(-t)*sin(pi*x));
title('解析解'),xlabel('位置 x'),ylabel('时间 t'),zlabel('数值解 u')
%*****
%t=tf=2 时各位置之解
%*****
figure(3)
M=length(t); %取终点时间的下标
xout=linspace(0,1,100); %输出点位置
[uout,dudx]=pdeval(m,x,sol(M,:),xout);
plot(xout,uout); %绘图
title('时间为 2 时,各位置下的解'),xlabel('x'),ylabel('u')
%*****

```

```

%偏微分方程函数
%*****
function [c,f,s]=ex20_1pdefun(x,t,u,dudx)
c=pi^2;f=dudx;s=0;
%*****
%初始条件函数
%*****
function u0=ex20_1ic(x)
u0=sin(pi*x);
%*****
%边界条件函数
%*****
function [pa,qa,pb,qb]=ex20_1bc(xa,ua,xb,ub,t)
pa=ua;qa=0;
pb=pi*exp(-t);qb=1;

```

例 5 试解以下联立的偏微分方程系统

$$\begin{aligned}\frac{\partial u_1}{\partial t} &= 0.024 \frac{\partial^2 u_1}{\partial x^2} - F(u_1 - u_2) \\ \frac{\partial u_2}{\partial t} &= 0.170 \frac{\partial^2 u_2}{\partial x^2} + F(u_1 - u_2)\end{aligned}$$

其中  $F(u_1 - u_2) = \exp(5.73(u_1 - u_2)) - \exp(-11.46(u_1 - u_2))$ ，且  $0 \leq x \leq 1$  和  $t \geq 0$ 。此联立偏微分方程系统满足以下初边值条件。

(i) 初值条件

$$u_1(x,0) = 1, \quad u_2(x,0) = 0$$

(ii) 边值条件

$$\begin{aligned}\frac{\partial u_1}{\partial x}(0,t) &= 0, \quad u_2(0,t) = 0 \\ u_1(1,t) &= 1, \quad \frac{\partial u_2}{\partial x}(1,t) = 0\end{aligned}$$

解 步骤 1: 改写偏微分方程为标准形式

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot \frac{\partial}{\partial t} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \frac{\partial}{\partial x} \begin{bmatrix} 0.024 \frac{\partial u_1}{\partial x} \\ 0.170 \frac{\partial u_2}{\partial x} \end{bmatrix} + \begin{bmatrix} -F(u_1 - u_2) \\ F(u_1 - u_2) \end{bmatrix}$$

这里 “ $\cdot$ ” 运算表示两个向量的对应元素相乘。因此

$$c = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad f = \begin{bmatrix} 0.024 \frac{\partial u_1}{\partial x} \\ 0.170 \frac{\partial u_2}{\partial x} \end{bmatrix}, \quad s = \begin{bmatrix} -F(u_1 - u_2) \\ F(u_1 - u_2) \end{bmatrix}$$

和  $m = 0$ 。另外，左边界条件( $x = 0$ 处)。写成

$$\begin{bmatrix} 0 \\ u_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0.024 \frac{\partial u_1}{\partial x} \\ 0.170 \frac{\partial u_2}{\partial x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

即

$$p_a = \begin{bmatrix} 0 \\ u_2 \end{bmatrix}, \quad q_a = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

同理, 右边界条件( $x=1$ 处)为

$$\begin{bmatrix} u_1 - 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 0.024 \frac{\partial u_1}{\partial x} \\ 0.170 \frac{\partial u_2}{\partial x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

即

$$p_b = \begin{bmatrix} u_1 - 1 \\ 0 \end{bmatrix}, \quad q_b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

步骤 2: 编写偏微分方程的系数向量函数。

```
function [c,f,s]=ex20_2pdefun(x,t,u,dudx)
c=[1 1]';f=[0.024 0.170]'.*dudx;
y=u(1)-u(2);F=exp(5.73*y)-exp(-11.47*y);
s=[-F F]';
```

步骤 3: 编写初始条件函数

```
function u0=ex20_2ic(x)
u0=[1 0]';
```

步骤 4: 编写边界条件函数

```
function [pa,qa,pb,qb]=ex20_2bc(xa,ua,xb,ub,t)
pa=[0 ua(2)]';qa=[1 0]';
pb=[ub(1)-1 0]';qb=[0 1]';
```

步骤 5: 取点。

由于此问题的端点均受边界条件的限制, 且时间  $t$  很小时状态的变动很大(由多次求解后的经验得知), 故在两端点处的点可稍微密集些。同时对于  $t$  小处亦可取密一些。例如,

```
x=[0 0.005 0.01 0.05 0.1 0.2 0.5 0.7 0.9 0.95 0.99 0.995 1];
t=[0 0.005 0.01 0.05 0.1 0.5 1 1.5 2];
```

以上几个主要步骤编写完成后, 事实上就可直接完成主程序来求解。此问题的求解程序如下:

```
function sol=ex20_2
%*****
%求解一维偏微分方程组的一个综合函数程序
%*****
m=0;
x=[0 0.005 0.01 0.05 0.1 0.2 0.5 0.7 0.9 0.95 0.99 0.995 1];
t=[0 0.005 0.01 0.05 0.1 0.5 1 1.5 2];
%*****
%利用 pdepe 求解
%*****
```

```

sol=pdepe(m,@ex20_2pdefun,@ex20_2ic,@ex20_2bc,x,t);
u1=sol(:,1); % 第一个状态之数值解输出
u2=sol(:,2); % 第二个状态之数值解输出
% *****
% 绘图输出
% *****
figure(1), surf(x,t,u1)
title('u1 之数值解'), xlabel('x'), ylabel('t')
%
figure(2), surf(x,t,u2)
title('u2 之数值解'), xlabel('x'), ylabel('t')
% *****
% pdefun 函数
% *****
function [c,f,s]=ex20_2pdefun(x,t,u,dudx)
c=[1 1]'; f=[0.024 0.170]'*dudx;
y=u(1)-u(2); F=exp(5.73*y)-exp(-11.47*y); s=[-F F]';
% *****
% 初始条件函数
% *****
function u0=ex20_2ic(x)
u0=[1 0]';
% *****
% 边界条件函数
% *****
function [pa,qa,pb,qb]=ex20_2bc(xa,ua,xb,ub,t)
pa=[0 ua(2)]'; qa=[1 0]';
pb=[ub(1)-1 0]'; qb=[0 1]';

```

### 3.3 应用实例

例 6 求下列偏微分方程组的数值解

$$-\frac{\partial T}{\partial L} + 0.59 \left( \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} \right) + \frac{9.611}{T} = 0$$

$$-\frac{\partial f}{\partial L} + 0.94 \left( \frac{\partial^2 f}{\partial r^2} + \frac{1}{r} \frac{\partial f}{\partial r} \right) + \frac{0.047}{T} = 0$$

其中  $T$  为温度( $^{\circ}\text{C}$ ),  $f$  为反应率,  $L$  为轴向距离,  $r$  为径向距离。此系统的边界条件为

$$r = 0: T = 0, f = 0, \frac{\partial T}{\partial r} = \frac{\partial f}{\partial r} = 0$$

$$r = 2: -0.65 \frac{\partial T}{\partial r} = 112(T - 273), \frac{\partial f}{\partial r} = 0$$

当  $L = 0$  时,  $T(r,0) = 125, f(r,0) = 0$ 。

将原方程改写成如式 (47) 的标准式

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial T}{\partial L} \\ \frac{\partial f}{\partial L} \end{bmatrix} = r^{-1} \frac{\partial}{\partial r} \begin{bmatrix} 0.59r \frac{\partial T}{\partial L} \\ 0.94r \frac{\partial f}{\partial L} \end{bmatrix} + \frac{1}{T} \begin{bmatrix} 9.611 \\ 0.047 \end{bmatrix}$$

因此

$$\tilde{c} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \tilde{f} = \begin{bmatrix} 0.59 \frac{\partial T}{\partial L} \\ 0.94 \frac{\partial f}{\partial L} \end{bmatrix}, \quad \tilde{s} = \frac{1}{T} \begin{bmatrix} 9.611 \\ 0.047 \end{bmatrix}$$

和  $m = +1$  (圆柱)。另外, 左边界条件 ( $r = 0$  处) 写成

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot \tilde{f} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

即

$$p_a = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad q_a = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

同理右边界条件 ( $r = 2$ ) 可写成

$$\begin{bmatrix} 112(T-273) \\ 0 \end{bmatrix} + \begin{bmatrix} 0.65/0.59 \\ 1 \end{bmatrix} \cdot \tilde{f} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

即

$$p_b = \begin{bmatrix} 112(T-273) \\ 0 \end{bmatrix}, \quad q_b = \begin{bmatrix} 0.65/0.59 \\ 1 \end{bmatrix}$$

根据以上的分析, 可编写 MATLAB 程序求解此 PDE 问题, 其求解程序如下:

```
function sol=main20_6
m=1;
%*****
% 取点
%*****
r=linspace(0,2,20);
L=linspace(0,5,40);
%*****
% 利用 pdepe 求解
%*****
sol=pdepe(m,@ex20_6pdefun,@ex20_6ic,@ex20_6bc,r,L);
T=sol(:,:,1); %温度
f=sol(:,:,2); %反应率
%*****
% 绘图输出
%*****
figure(1),surf(L,r,T')
title('temp'),xlabel('L'),ylabel('r'),zlabel('temp (0C)')
%
figure(2),surf(L,r,f')
title('reaction rate'),xlabel('L'),ylabel('r'),zlabel('reaction rate')
```

```

%*****
%偏微分方程函数
%*****
function [c,f,s]=ex20_6pdefun(r,L,u,DuDr)
T=u(1);f=u(2);
c=[1 1]';
f=[0.59 0.94]'.*DuDr;
s=[9.611;0.047]/T;
%*****
%初始条件函数
%*****
function u0=ex20_6ic(x)
u0=[125 0]';
%*****
% 边界条件函数
%*****
function [pa,qa,pb,qb]=ex20_6bc(ra,ua,rb,ub,L)
pa=[0 0]';qa=[1 1]';
pb=[112*(ub(1)-273) 0]';
qb=[0.65/0.59 1]';

```

### 例 7 扩散系统之浓度分布

管中储放静止液体 B，高度为  $L=10$  cm，放置于充满 A 气体的环境中。假设与 B 液体接触面之浓度为  $C_{A0} = 0.01 \text{ mol/m}^3$ ，且此浓度不随时间改变而改变，即在操作时间内 ( $h = 10$  天) 维持定值。气体 A 在液体 B 中之扩散系数为  $D_{AB} = 2 \times 10^{-9} \text{ m}^2/\text{s}$ 。试决定以下两种情况下，气体 A 溶于液体 B 中的流通量(flux)。

(a) A 与 B 不发生反应；

(b) A 与 B 发生反应

解 (a) 因气体 A 与液体 B 不发生反应，故其扩散现象的质量平衡方程如下：

$$\frac{\partial C_A}{\partial t} = D_{AB} \frac{\partial^2 C_A}{\partial z^2}$$

依题意，其初始及边界条件为

$$\text{I.C. } C_A(z,0) = 0, \quad z > 0$$

$$\text{B.C. } C_A(0,t) = C_{A0}, \quad t \geq 0; \quad \left. \frac{\partial C_A}{\partial z} \right|_{z=L} = 0, \quad t \geq 0$$

在获得浓度分布后，即可用 Fick 定律

$$N_{Az}(t) = -D_{AB} \left. \frac{\partial C_A}{\partial z} \right|_{z=0}$$

计算流通量。

与标准式 (47) 比较，可得  $C = 1$ ， $f = D_{AB} \partial C_A / \partial z$ ， $s = 0$ ，和  $m = 0$ 。另外，经与式 (49) 和 (50) 比较后得知，左边界及右边界条件的系数分别为

$$\text{左边界}(z = 0): \quad p_a = C_A(0,t) - C_{A0}, \quad q_a = 0。$$

$$\text{右边界}(z = L): \quad p_b = 0, \quad q_b = \frac{1}{D_{AB}}。$$

(b)在气体 A 与液体 B 会发生一次反应的情况下，其质量平衡方程需改写为

$$\frac{\partial C_A}{\partial t} = D_{AB} \frac{\partial^2 C_A}{\partial z^2} + kC_A$$

其中  $k = 2 \times 10^{-7}$ 。

而起始及边界条件同上。

与标准式 (47) 比较，可得  $m = 0$ ， $C = 1$ ， $f = D_{AB} \partial C_A / \partial z$ ，和  $s = kC_A$ 。而边界条件的系数同 (a)。

利用以上的处理结果，可编写 MATLAB 程序如下：

```
function ex20_7
%*****
% 扩散系统之浓度分布
%*****
clear,clc
global DAB k CA0
%*****
% 给定数据
%*****
CA0=0.01;L=0.1;DAB=2e-9;
k=2e-7;h=10*24*3600;
%*****
% 取点
%*****
t=linspace(0,h,100);z=linspace(0,L,10);
%*****
% case (a)
%*****
m=0;
sol=pdepe(m,@ex20_7pdefuna,@ex20_7ic,@ex20_7bc,z,t);
CA=sol;
for i=1:length(t)
    [CA_i,dCAdz_i]=pdeval(m,z,CA(i,:),0);
    NAz(i)=-dCAdz_i*DAB;
end
figure(1),subplot(211)
surf(z,t/(24*3600),CA),title('case (a)')
xlabel('length (m)'),ylabel('time (day)'),zlabel('conc. (mol/m^3)')
subplot(212)
plot(t/(24*3600),NAz'*24*3600)
xlabel('time (day)'),ylabel('flux (mol/m^2.day)')
%*****
% case (b)
%*****
m=0;
solb=pdepe(m,@ex20_7pdefunb,@ex20_7ic,@ex20_7bc,z,t);
CA=sol;
for i=1:length(t)
    [CA_i,dCAdz_i]=pdeval(m,z,CA(i,:),0);
    NAz(i)=-dCAdz_i*DAB;
end
%
figure(2)
```

```

subplot(211)
surf(z,t/(24*3600),CA),title('case (b)')
xlabel('length (m)'),ylabel('time (day)'),zlabel('conc. (mol/m^3)')
subplot(212)
plot(t/(24*3600),NAz'*24*3600)
xlabel('time (day)'),ylabel('flux (mol/m^2.day)')
%*****
%偏微分方程函数
%*****
% case (a)
%*****
function [c,f,s]=ex20_7pdefuna(z,t,CA,dCA dz)
global DAB k CA0
c=1;f=DAB*dCA dz;s=0;
%*****
% case (b)
%*****
function [c,f,s]=ex20_7pdefunb(z,t,CA,dCA dz)
global DAB k CA0
c=1;f=DAB*dCA dz;s=k*CA;
%*****
% 初始条件函数
%*****
function CA_i=ex20_7ic(z)
CA_i=0;
%*****
% 边界条件函数
%*****
function [pa,qa,pb,qb]=ex20_7bc(za,CAa,zb,CAb,t)
global DAB k CA0
pa=CAa-CA0;qa=0;
pb=0;qb=1/DAB;

```

#### §4 二维状态空间的偏微分方程的 MATLAB 解法

MATLAB 中的偏微分方程 (PDE) 工具箱是用有限元法寻求典型偏微分方程的数值近似解,该工具箱求解偏微分方程具体步骤与用有限元方法求解偏微分方程的过程是一致的,包括几个步骤,即几何描述、边界条件描述、偏微分方程类型选择、有限元划分计算网格、初始化条件输入,最后给出偏微分方程的数值解(包括画图)。

下面我们讨论的方程是定义在平面上的有界区域  $\Omega$  上,区域的边界记作  $\partial\Omega$ 。

##### 4.1 方程类型

MATLAB 工具箱可以解决下列类型的偏微分方程:

(i) 椭圆型偏微分方程

$$-\nabla \cdot (c\nabla u) + au = f \quad \text{in } \Omega$$

其中  $c, a, f$  和未知的  $u$  可以是  $\Omega$  上的复值函数。

(ii) 抛物型偏微分方程

$$d \frac{\partial u}{\partial t} - \nabla \cdot (c\nabla u) + au = f \quad \text{in } \Omega$$

其中  $c, a, f, d$  可以依赖于时间  $t$ 。

(iii) 双曲型偏微分方程

$$d \frac{\partial^2 u}{\partial t^2} - \nabla \cdot (c \nabla u) + au = f \quad \text{in } \Omega$$

(iv) 特征值问题

$$-\nabla \cdot (c \nabla u) + au = \lambda du \quad \text{in } \Omega$$

其中  $\lambda$  是未知的特征值,  $d$  是  $\Omega$  上的复值函数。

(v) 非线性椭圆偏微分方程

$$-\nabla \cdot (c(u) \nabla u) + a(u)u = f(u) \quad \text{in } \Omega$$

其中  $c, a, f$  可以是  $u$  的函数。

(vi) 方程组

$$\begin{cases} -\nabla \cdot (c_{11} \nabla u_1) - \nabla \cdot (c_{12} \nabla u_2) + a_{11}u_1 + a_{12}u_2 = f_1 \\ -\nabla \cdot (c_{21} \nabla u_1) - \nabla \cdot (c_{22} \nabla u_2) + a_{21}u_1 + a_{22}u_2 = f_2 \end{cases}$$

#### 4.2 边界条件

边界条件有如下三种:

(i) Dirichlet 条件:  $hu = r$  on  $\partial\Omega$ 。

(ii) Neumann 条件:  $\vec{n} \cdot (c \nabla u) + qu = g$  on  $\partial\Omega$ 。

这里  $\vec{n}$  为区域的单位外法线,  $h, r, q, g$  是定义在  $\partial\Omega$  上的复值函数。

对于二维方程组情形, Dirichlet 边界条件为

$$\begin{aligned} h_{11}u_1 + h_{12}u_2 &= r_1, \\ h_{21}u_1 + h_{22}u_2 &= r_2; \end{aligned}$$

Neumann 边界条件为:

$$\begin{aligned} \vec{n} \cdot (c_{11} \nabla u_1) + \vec{n} \cdot (c_{12} \nabla u_2) + q_{11}u_1 + q_{12}u_2 &= g_1 \\ \vec{n} \cdot (c_{21} \nabla u_1) + \vec{n} \cdot (c_{22} \nabla u_2) + q_{21}u_1 + q_{22}u_2 &= g_2 \end{aligned}$$

(iii) 对于偏微分方程组, 混合边界条件为

$$\begin{aligned} h_{11}u_1 + h_{12}u_2 &= r_1 \\ \vec{n} \cdot (c_{11} \nabla u_1) + \vec{n} \cdot (c_{12} \nabla u_2) + q_{11}u_1 + q_{12}u_2 &= g_1 + \mu h_{11} \\ \vec{n} \cdot (c_{21} \nabla u_1) + \vec{n} \cdot (c_{22} \nabla u_2) + q_{21}u_1 + q_{22}u_2 &= g_2 + \mu h_{12} \end{aligned}$$

这里  $\mu$  的计算是使得满足 Dirichlet 边界条件。

#### 4.3 求解偏微分方程

例 8 求解泊松方程

$$-\nabla^2 u = 1,$$

求解区域为单位圆盘, 边界条件为在圆盘边界上  $u = 0$ 。

解 它的精确解为

$$u(x, y) = \frac{1 - x^2 - y^2}{4}$$

下面求它的数值解, 编写程序如下:

% (1) 问题定义

g='circleg';

% 单位圆

b='circlebl';

% 边界上为零条件

c=1;a=0;f=1;

% (2) 产生初始的三角形网格

```

[p,e,t]=initmesh(g);
% (3) 迭代直至得到误差允许范围内的合格解
error=[]; err=1;
while err > 0.01,
[p,e,t]=refinemesh(g,p,e,t);
u=asempde(b,p,e,t,c,a,f); %求得数值解
exact=(1-p(1,:).^2-p(2,:).^2)/4;
err=norm(u-exact',inf);
error=[error err];
end
%结果显示
subplot(2,2,1),pdemesh(p,e,t);
subplot(2,2,2),pdesurf(p,t,u)
subplot(2,2,3),pdesurf(p,t,u-exact')

```

例9 考虑最小表面问题

$$-\nabla \cdot \left( \frac{1}{\sqrt{1+|\nabla u|^2}} \nabla u \right) = 0 \quad \text{在} \quad \Omega = \{(x,y) | x^2 + y^2 \leq 1\},$$

在圆盘边界上  $u = x^2$ 。

解 这是椭圆型方程，其中  $c = \frac{1}{\sqrt{1+|\nabla u|^2}}$ ,  $a = 0, f = 0$ ，编写程序如下：

```

g='circleg';
b='circleb2';
c='1./sqrt(1+ux.^2+uy.^2)';
rtol=1e-3;
[p,e,t]=initmesh(g);
[p,e,t]=refinemesh(g,p,e,t);
u=pdnonlin(b,p,e,t,c,0,0,'Tol',rtol);
pdesurf(p,t,u)

```

例10 求解正方形区域  $\{(x,y) | -1 \leq x, y \leq 1\}$  上的热传导方程

$$\frac{\partial u}{\partial t} = \Delta u$$

$$\text{初始条件为 } u(0) = \begin{cases} 1, & x^2 + y^2 < 0.4^2 \\ 0, & \text{其它} \end{cases}$$

边界条件为Dirichlet条件  $u = 0$ 。

解 这里是抛物型方程，其中  $c = 1, a = 0, f = 0, d = 1$ 。编写程序如下：

```

%(1)问题定义
g='squareg'; %定义正方形区域
b='squareb1'; %边界上为零条件
c=1;a=0;f=0;d=1;
%(2)产生初始的三角形网格
[p,e,t]=initmesh(g);
%(3)定义初始条件
u0=zeros(size(p,2),1);
ix=find(sqrt(p(1,:).^2+p(2,:).^2)<0.4);

```

```

u0(ix)=1
%(4)在时间段为0到0.1的20个点上求解
nframe=20;
tlist=linspace(0,0.1,nframe);
u1=parabolic(u0,tlist,b,p,e,t,c,a,f,d);
%(5)动画图示结果
for j=1:nframe
    pdesurf(p,t,u1(:,j));
    mv(j)=getframe;
end
movie(mv,10)

```

例11 求解正方形区域  $\{(x, y) | -1 \leq x, y \leq 1\}$  上的波方程

$$\frac{\partial^2 u}{\partial t^2} = \Delta u$$

初始条件为  $u(0) = \arctan(\cos(\pi x))$ ,  $\frac{du(0)}{dt} = 3 \sin(\pi x) \exp(\cos(\pi y))$ , 边界条件为在

$x = \pm 1$  上满足Dirichlet条件  $u = 0$ , 在  $y = \pm 1$  上满足Neumann条件  $\frac{\partial u}{\partial n} = 0$ 。

解 这里是双曲型方程, 其中  $c = 1, a = 0, f = 0, d = 1$ 。编写程序如下:

```

%(1)问题定义
g='squareg';          %定义正方形区域
b='squareb3';        %定义边界
c=1;a=0;f=0;d=1;
%(2)产生初始的三角形网格
[p,e,t]=initmesh(g);
%(3)定义初始条件
x=p(1,:);y=p(2,:);
u0=atan(cos(pi*x));
ut0=3*sin(pi*x).*exp(cos(pi*y));
%(4)在时间段为0到5的31个点上求解
n=31;
tlist=linspace(0,5,n);
uu=hyperbolic(u0,ut0,tlist,b,p,e,t,c,a,f,d);
%(5)动画图示结果
for j=1:n
    pdesurf(p,t,uu(:,j));
    mv(j)=getframe;
end
movie(mv,10)

```

例12 求解泊松方程

$$-\nabla^2 u = \delta(0,0),$$

求解区域为单位圆盘, 边界条件为在圆盘边界上  $u = 0$ 。

解 它的精确解为

$$u(x, y) = -\frac{1}{2\pi} \ln \sqrt{x^2 + y^2}。$$

下面求它的数值解，编写程序如下：

```
g='circleg';
b='circlebl';
c=1;a=0;f='circlef';
[p,e,t]=initmesh(g);
[p,e,t]=refinemesh(g,p,e,t);
u=asempde(b,p,e,t,c,a,f);
exact=-1/(2*pi)*log(sqrt(p(1,:).^2+p(2,:).^2));
subplot(2,2,1),pdemesh(p,e,t);
subplot(2,2,2),pdesurf(p,t,u)
subplot(2,2,3),pdesurf(p,t,u-exact')
```

#### 4.4 偏微分方程的 pdetool 解法

##### 4.4.1 图形界面解法简介

对于一般的区域，任意边界条件的偏微分方程，我们可以利用 MATLAB 中 pdetool 提供的偏微分方程用户图形界面解法。

图形界面解法步骤大致上为：

- (1) 定义 PDE 问题，包括二维空间范围，边界条件以及 PDE 系数等。
- (2) 产生离散化点，并将原 PDE 方程离散化。
- (3) 利用有限元素法(finite element method; FEM)求解并显示答案。

在说明此解法工具之前，先介绍此 PDE 图形界面的菜单下方的功能图标(icon)按钮。



透过这些按钮，使用者可轻松地完成偏微分方程的求解。现将这些按钮的主要功能叙述如下：

前五个按钮为 PDE 系统的边界范围绘制功能，由左至右的用法为：

：以对角绘制矩形或正方形。按住鼠标左键可绘制矩形，而正方形需以按住右键的方式绘制。

：从中心点至某一角边的方式绘制矩形或正方形。同样地，鼠标左键绘矩形，右键绘正方形。

：由周围界线的方式绘制椭圆或圆形区域。鼠标左键用以绘制椭圆，而右键用来绘制圆形图形。

：以中心点向外的方式绘制椭圆或圆。同样地，鼠标左键及右键，分别用以绘制椭圆及圆形的区域。

：用以绘制多边形等不规则区域，欲关闭此功能需按鼠标右键。

在这些绘制按钮之后的按钮功能依序如下：

：用以给定边界条件。在此功能选定后，使用者可在任一图形边界上按住鼠标左键双击，然后在对话框中输入边界条件。

：用以指定 PDE 问题及相关参数。

：产生图形区域内离散化的网点。

：用以进一步将离散化的网点再取密一点(refine mesh)。

：在指定 PDE 系统，边界条件及区域后，按此按钮即开始解题。

：用以指定显示结果绘制方式。



放大缩小功能，便于图形绘制及显示。

#### 4.4.2 图形界面解法的使用步骤

要利用 `pdetool` 接口求解之前，需先定义 PDE 问题，其包含三大部份：

- (1) 利用绘图(draw)模式，定义欲解问题的空间范围(domain)  $\Omega$ 。
- (2) 利用 boundary 模式，指定边界条件。
- (3) 利用 PDE 模式，指定 PDE 系数，即输入  $c$ ， $a$ ， $f$  和  $d$  等 PDE 模式中的系数。

在定义 PDE 问题之后，可依以下两个步骤求解

- (1) 在 mesh 模式下，产生 mesh 点，以便将原问题离散化。
- (2) 在 solve 模式下，求解。
- (3) 最后，在 Plot 模式下，显示答案。

下面以 *Poisson's* 方程式  $-\Delta u = 10$  的求解为例，详细说明 `pdetool` 的用法。此问题的几何图形及相关边界条件，将于求解过程中加以说明。

步骤 1：在命令窗口中键入 `pdetool` 以进入 GUI(graphical user interface)界面。选取 Options 中之 Grid 功能，以显示网格线。

步骤 2：利用 Draw 功能，画出问题之几何图形。请注意：使用者可利用内定对象“多边形”，“矩形”，“正方形”，“圆形”，及“椭圆型”，予以组合，例如

(i)先选取“矩形/正方形”对象 ，移动鼠标至所欲输入左上角点，如坐标(-1,0)点，按住鼠标左键，往右下角拉至坐标为(1,-0.4)处，即形成代号为 R1 的矩形。其余图形 C1, R1 和 C2 可选取适当对象并类似地画出，以形成如图 1 的图形区域。以代数公式而言，其为  $R1+C1+R2+C2$ 。

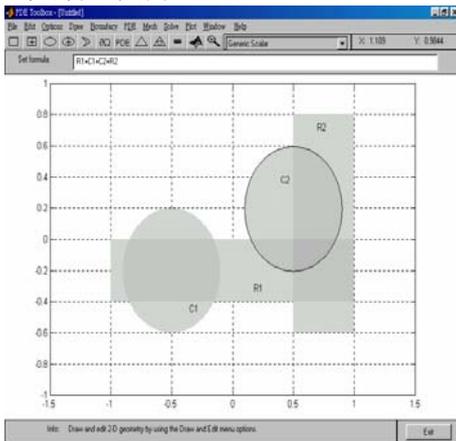


图 1 画求解区域

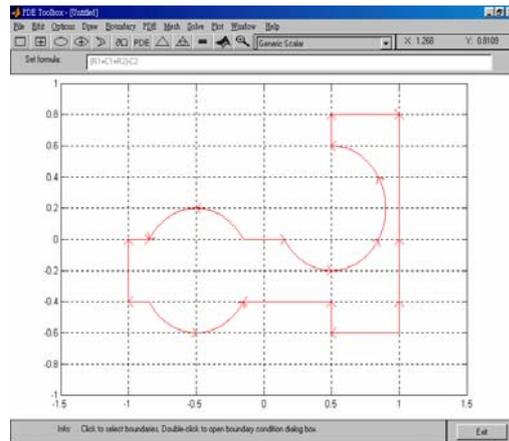


图 2 求解区域图

值得注意的是，圆形区域需以按住鼠标右键的方式来制作(非左键)。同时，如欲进一步修改各图形对象之大小及位置数据，可在该图上双击鼠标左键，然后在对象对话框上输入数据。

(ii)若所欲形成的图形区域，需将  $C2$  去除，则可在公式列中直接输入  $R1+C1+R2-C2$  即可。

步骤 3：选取 PDE 功能项，以输入 PDE 方程的系数及类型。因问题为  $-\Delta u = f$ ，故此为椭圆型的问题，且其标准形式为  $-\nabla \cdot (c \nabla u) + au = f$ ，比较得知， $c=1$ ， $a=0$  和  $f=10$ ，所以对对话框输入的情况如图 3。

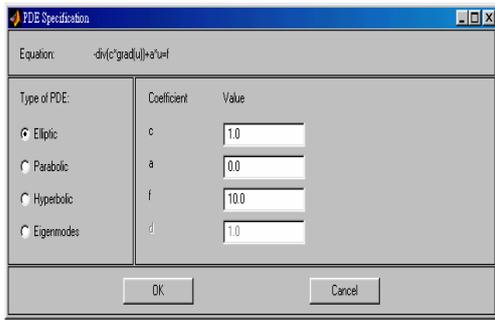


图3 对话框输入

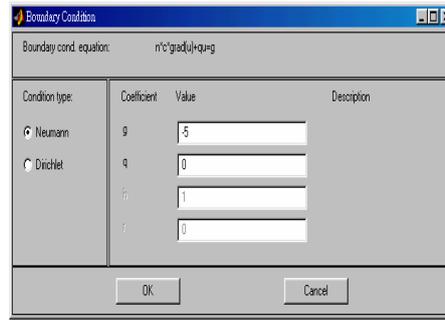


图4 对话框输入

步骤4: 选取 Boundary 功能, 以输入边界条件。假设弧形部分边界条件为 Neumann 形, 且为  $\partial u / \partial n = -5$ , 与标准式比较知,  $g=-5$  且  $q=0$ 。直线部分其边界条件则在 Dirichlet type 使  $h=0$ ,  $r=0$ 。对话框输入情况见图4。

步骤5: 选取 Mesh 功能, 产生网点。使用者亦可进一步利用  将网点取得密一点(refine mesh), 见图5。

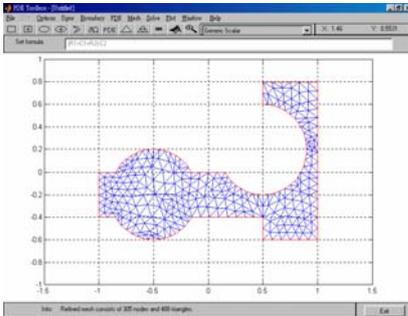


图5 网格剖分求解区域

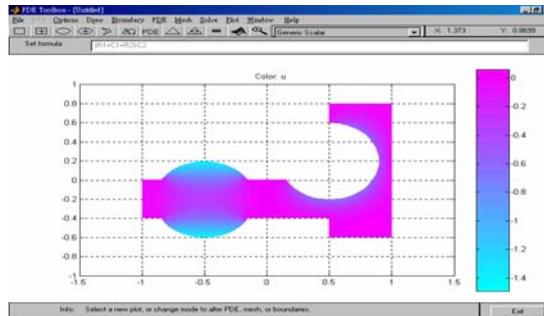


图6 求解结果示意图

步骤6: 选取 solve 功能, 解此 PDE, 见图6。

注意:

1. MATLAB 会以图形的方式展示结果, 使用者亦可点选 plot 下的“parameters”功能, 选择适当的方式显示图形及数据。例如用 3D 方式显示求解结果。参数设置见图7, 显示结果见图8。

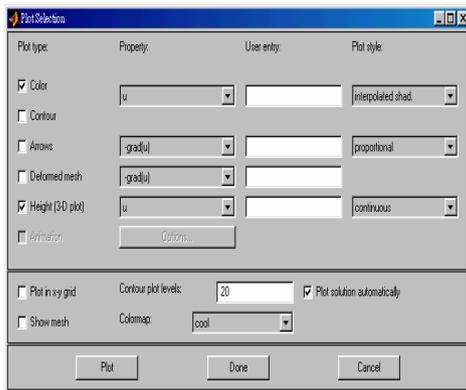


图7 显示参数设置

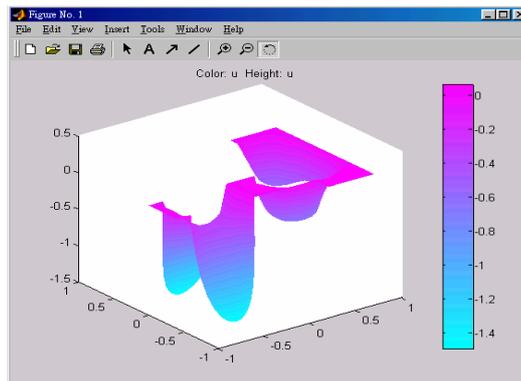


图8 求解结果显示

2. 另外,若使用者欲将结果输出到命令窗口中,以供后续处理,可利用 solve 功能项下的“export solution”指定变量名称来完成。

3. 如果求抛物型或双曲型方程的数值解,还需要通过“solve”菜单下的“parameters...”选项设置初值条件。

4. 在上面定义边界条件和初始条件时,可以使用一些内置变量。

(1) 在边界条件输入框中,可以使用如下变量:

二维坐标  $x$  和  $y$ , 边界线段长度参数  $s$ , 外法向矢量的分量  $n_x$  和  $n_y$  (如果需要边界的切线方向,可以通过  $t_x=-n_y$  和  $t_y=n_x$  表示), 解  $u$ 。

(2) 在初值条件的输入框中,也可以输入用户定义的 MATLAB 可接受变量 ( $p, e, t, x, y$ ) 的函数。

例 13 使用 PDETOOL 重新求例 10 的数值解。

1) 定义 PDE 问题,包括二维空间范围,边界条件以及 PDE 系数等。我们这里就省略了。

2) 区域剖分以后,通过“Mesh”菜单下的“Export Mesh...”选项可以把  $p, e, t$  三个参数分别输出到 Matlab 工作空间。

3) 然后编写函数 fun1(x,y)如下:

```
function f=fun1(x,y);  
f=zeros(length(x),1);  
ix=find(x.^2+y.^2<0.16);  
f(ix)=1;
```

其中的变量  $x, y$  是 MATLAB 可接受的内置变量。

设置“solve”菜单下的“parameters...”选项如下:

时间框中输入: linspace(0,0.1,20);

初值框中输入: fun1。

4) 设置“plot”菜单下的“parameters...”选项如下:选择 Height(3-D plot) 和 Animation 两项。

5) 用鼠标点一下工具栏上的“=”按钮,就可以画出数值解的 3-D 图形。

例 14 两端固定的均匀弦的自由振动的定解问题是

$$\begin{cases} u_{tt} = u_{xx} \\ u(0,t) = u(2,t) = 0 \\ u(x,0) = e^{-(x-1)^2}, \quad u_t(x,0) = 0 \end{cases}$$

也可以使用 pde tool 求解弦振动问题的数值解。

使用 pde tool,可以在用户界面上操作,省去编程的过程,关键是如何利用解二维问题的工具箱来解一维问题。我们是用二维的窄带来表示一维的弦,带的侧面不受力,也就是自由的,对于两端固定的弦而言,带的两端是固定的。

在菜单 Options/Axes Limits 下选择  $x$  轴范围为  $0 \sim 2$ ,  $y$  轴范围为  $-0.2 \sim 0.2$ 。以圆点为顶点画一个长为 2 宽为 0.2 的矩形,矩形的顶点为  $(0,-0.1), (0,0.1), (2,-0.1), (2,0.1)$ 。

按照题意,矩形的左右边界是齐次的 Dirichlet 边界条件,可取  $h=1, r=0$ , 上下边界则取齐次的 Neumann 边界条件,即  $g=0, q=0$ 。

方程的设置是 hyperbolic 型,系数是  $c=1, a=0, f=0, d=1$ 。

为了有足够的精度，将初始化的网格作了两次细分。在解方程的参数设置对话框 Solve Parameters 中，将 Time 设为 0:0.05:3.6， $u(t_0)$  设为  $\exp(-(x-1)^2)$ ，而  $u'(t_0)$  设为 0。作图选项对话框 Plot Parameters 中，在 Plot Type 下，选 Color, Height (3-D Plot), Animation。

最后用鼠标点一下工具栏上的“=”按钮，就可以画出数值解的 3-D 图形。

## 习题二十

1. 求二维拉普拉斯方程

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

在边界条件  $u(x, y)|_{x=0} = u(x, y)|_{x=6} = u(x, y)|_{y=0} = 0$ ， $u(x, y)|_{y=2} = 10$  下的数值解。

2. 求初边值问题

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, & t > 0, 0 < x < 1 \\ u(x, 0) = 1, & 0 \leq x \leq 1 \\ u'_x(0, t) - u(0, t) = 0 \\ u'_x(1, t) + u(1, t) = 0 \end{cases} \quad t \geq 0$$

在  $0 \leq t \leq 3$  范围内的数值解。

3. 求热传导方程初边值问题

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} & 0 < x < 1, 0 < t < 0.5 \\ u(x, 0) = \sin \pi x & 0 \leq x \leq 1 \\ u(0, t) = u(1, t) = 0 & 0 \leq t \leq 0.5 \end{cases}$$

的数值解，并将计算结果与准确解  $u(x, t) = e^{-\pi^2 t} \sin \pi x$  比较。