二维椭圆型方程的差分格式

肖名财　刘礼海

一、 引言

各种物理性质的许多稳定过程都归结为椭圆形偏微分方程，诸如定常热传导问题和扩散问题、导体中电流分布问题、静电学和静磁学问题、弹性理论和渗流理论问题等等。

椭圆型方程边值问题的精确解只在一些特殊情况下求得。有些问题即使求得他的解析解，但计算往往也很复杂，因此必须善于近似地求解这些问题。

较具代表性得椭圆型方程是二维 Poisson 方程

$$-(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2})=f(x,y),\quad (x,y)\in\Omega \tag{1}$$

和 Laplace 方程

$$-(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2})=0,\quad (x,y)\in\Omega \tag{2}$$

其中 Ω 为 R^2 中的一个有界区域。

定解条件通常有三类 (Dirichlet 边界条件、Neumann 边界条件、第三类边值条件)，我们主要考虑 Dirichlet 边值问题

$$\begin{cases} -\Delta u=f(x,y),\quad (x,y)\in\Omega & \text{(3a)} \\ u=\varphi(x,y),\quad (x,y)\in\tau & \text{(3b)} \end{cases}$$

其中 $\Delta u=\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}$, 为简单起见，考虑 Ω 为矩形区域

$$\Omega=\{(x,y)|a<x<,c<y<d\} \tag{4}$$

二、 数值格式

将区间 [a,b] 作 m 等分，记 $h_1=\frac{b-a}{m},x_i=a+ih_1,0\le i\le m$

将区间 [c,d] 作 n 等分，记 $h_2=\frac{d-c}{n},y_j=c+jh_2,0\le j\le n$

分别称 h_1 为 x 方向的步长，h_2 为 y 方向的步长，用两簇平行线

$$x=x_i,0\le i\le m \tag{5}$$

$$y=y_j,0\le j\le n \tag{6}$$

将区域 Ω 剖分为 mn 个小矩形，称两簇直线的交点 (x_i,y_j) 为网格节点。

我们称 $i=0,i=m,j=0,j=n$ 处的节点为边界节点，记为 γ, 其余的节点为内节点，记为 ω。

2.1 五点差分格式

在节点 (x_i, y_j) 处考虑边值问题 (3a),(3b) 有

$$\begin{cases} -[\frac{\partial^2 u}{\partial x^2}(x_i, y_j) + \frac{\partial^2 u}{\partial y^2}(x_i, y_j)] = f(x_i, y_j) \quad (i, j) \in \omega & \text{(7a)} \\ u(x_i, y_j) = \varphi(x_i, y_j) \quad (i, j) \in \gamma & \text{(7b)} \end{cases}$$

用二阶中心差商代替二阶导数

$$-\Delta h u_{i,j} = -[\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_1^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_2^2}] = f_{i,j} \tag{8}$$

其中 $u_{i,j}$ 为节点 (i,j) 上的网函数。导出截断误差，由 Taylor 展开

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_1^2} = \frac{\partial u(x_i, y_j)}{\partial x^2} + \frac{h_1^2}{12} \frac{\partial^4 u(x_i, y_j)}{\partial x^4} + \frac{h_1^4}{360} \frac{\partial^6 u(x_i, y_j)}{\partial x^6} + O(h_1^6) \tag{9}$$

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_1^2} = \frac{\partial u(x_i, y_j)}{\partial y^2} + \frac{h_2^2}{12} \frac{\partial^4 u(x_i, y_j)}{\partial y^4} + \frac{h_2^4}{360} \frac{\partial^6 u(x_i, y_j)}{\partial y^6} + O(h_2^6) \tag{10}$$

差分算子 $-\Delta h$ 的截断误差

$$R_{i,j}(u) = \Delta u(x_i, y_j) - \Delta h u(x_i, y_j) = -\frac{1}{12}[h_1^2 \frac{\partial^4 u}{\partial x^4} + h_2^2 \frac{\partial^4 u}{\partial y^4}] + O(h^4) = O(h^2) \tag{11}$$

去掉截断误差，然后加上边值条件，得到如下差分格式

$$\begin{cases} -[\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_1^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_2^2}] = f_{i,j} \quad (i, j) \in \omega & \text{(12a)} \\ u_{i,j} = \varphi(x_i, y_j) \quad (i, j) \in \gamma & \text{(12b)} \end{cases}$$

截断误差为 $O(h^2)$.

称 (12a)-(12b) 为五点差分格式.

注:

(1) 若 $h_1 = h_2 = h_3$, 则 (12a) 可简化为

$$u_{i,j} - \frac{1}{4}(u_{i-1,j} + u_{i+1,j} + ui, j-1 + u_{i,j+1}) = \frac{h^2}{4} f_{i,j} \tag{13}$$

(2) 若 $f \equiv 0$(Laplace 方程), 则 (12a) 可简化为

$$u_{i,j} = \frac{1}{4}(u_{i-1,j} + u_{i+1,j} + ui, j-1 + u_{i,j+1}) \tag{14}$$

我们将 (12a) 整理得

$$-\frac{1}{h_2^2} u_{i,j-1} - \frac{1}{h_1^2} u_{i-1,j} + 2(\frac{1}{h_1^2} + \frac{1}{h_2^2}) u_{i,j} - \frac{1}{h_1^2} u_{i+1,j} - \frac{1}{h_2^2} u_{i,j+1} = f_{i,j} \tag{15}$$

写成矩阵形式

$$Au = f \tag{16}$$

其中

$$A = \begin{bmatrix} D & C & & \\ C & D & C & \\ & \ddots & \ddots & \ddots \\ & & C & D \end{bmatrix}$$

$$D = \begin{bmatrix} 2(\frac{1}{h_1^2}+\frac{1}{h_2^2}) & -\frac{1}{h_2^2} & & \\ -\frac{1}{h_2^2} & 2(\frac{1}{h_1^2}+\frac{1}{h_2^2}) & -\frac{1}{h_2^2} & \\ & \ddots & \ddots & \ddots \\ & & -\frac{1}{h_2^2} & 2(\frac{1}{h_1^2}+\frac{1}{h_2^2}) \end{bmatrix}$$

$$C = \begin{bmatrix} -\frac{1}{h_1^2} & & & \\ & -\frac{1}{h_1^2} & & \\ & & \ddots & \\ & & & -\frac{1}{h_1^2} \end{bmatrix}$$

$$u = \begin{bmatrix} u_{1,1}, u_{1,2}, \cdots, u_{1,n-1} \\ u_{2,1}, u_{2,2}, \cdots, u_{2,n-1} \\ \vdots \\ u_{m-2,1}, u_{m-2,2}, \cdots, u_{m-2,n-1} \\ u_{m-1,1}, u_{m-1,2}, \cdots, u_{m-1,n-1} \end{bmatrix}$$

解方程 (16), 可以得到数值解。

2.2 九点差分格式

同样类似五点差分格式，我们用二阶中心差商代替二阶导数，由 Taylor 展开得到 (9)、(10) 两式。

九点格式是为了得到更高精度的截断误差，因此我们将 (11) 式中的 h^2 项由 u 转化为已知函数 f 得

$$\Delta u(x_i, y_j) = \Delta u(x_i, y_j) + \frac{1}{12}(h_1^2 \frac{\partial^4 u(x_i, y_j)}{\partial x^4} + h_2^2 \frac{\partial^4 u(x_i, y_j)}{\partial y^4}) + O(h^4) \tag{17}$$

含由 h^2 的部分经转化变为

$$\frac{1}{12}(h_1^2 \frac{\partial^2}{\partial x^2} + h_2^2 \frac{\partial^2}{\partial y^2})(\frac{\partial^2 u(x_i, y_j)}{\partial x^2} + \frac{\partial^2 u(x_i, y_j)}{\partial y^2}) - \frac{h_1^2 + h_2^2}{12} \frac{\partial^4 u(x_i, y_j)}{\partial x^2 \partial y^2} + O(h^4) \tag{18}$$

为了方便计算

我们将 $u''_{xx}(x_i,y_{j+1})$、$u''_{xx}(x_i,y_j)$、$u''_{xx}(x_i,y_{j-1})$ 分别记为 a、b、c
而

$$\frac{\partial^4 u(x_i,y_j)}{\partial x^2 \partial y^2} = \frac{a-2b+c}{h_2^2} + O(h_2^2) \tag{19}$$

使用二阶中心差商代替二阶导数，所以

$$\begin{cases} a = \dfrac{u(x_{i+1},y_{j+1}) - 2u(x_i,y_{j+1}) + u(x_{i-1},y_{j+1})}{h_1^2} & (20a) \\ b = \dfrac{u(x_{i+1},y_j) - 2u(x_i,y_j) + u(x_{i-1},y_j)}{h_1^2} & (20b) \\ c = \dfrac{u(x_{i+1},y_{j-1}) - 2u(x_i,y_{j-1}) + u(x_{i-1},y_{j-1})}{h_1^2} & (20c) \end{cases}$$

使用 (20a)、(20b)、(20c) 替换 (19) 中的 a、b、c, 然后带入到 (18) 得

$$\begin{aligned} -f(x_i,y_j) + O(h^4) =& [\frac{u(x_{i_1},y_j) - 2u(x_i,y_j) + u(x_{i+1},y_j)}{h_1^2} + \frac{u(x_i,y_{j-1}) - 2u(x_i,y_j) + u(x_i,y_{j+1})}{h_2^2}] \\ &+ \frac{1}{12}[4u(x_i,y_j) - 2(u(x_i,y_{j+1}) + u(x_i,y_{j-1}) + u(x_{i+1,y_j}) + u(x_i,y_{j+1})) + \\ & u(x_{i+1},y_{j-1}) + u(x_{i+1},y_{j-1}) + u(x_{i+1},y_{j+1}) + u(x_{i-1},y_{j+1})]\frac{h_1^2+h_2^2}{h_1^2h_2^2} + \\ & \frac{1}{12}[h_1^2\frac{\partial^2 f(x_i,y_j)}{\partial x^2} + h_2^2\frac{\partial^2 f(x_i,y_j)}{\partial y^2}] \end{aligned} \tag{21}$$

略去误差项，加上边值条件，得 Poisson 方程九点差分格式

$$\begin{cases} -[\dfrac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h_1^2} + \dfrac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_2^2}]\dfrac{h_1^2+h_2^2}{h_1^2h_2^2} = f_{i,j} + G & (22a) \\ G = \dfrac{1}{12}[h_1^2 f''_{xx}(x_i,y_j) + h_2^2 f''_{yy}(x_i,y_j)] & (22b) \\ 1 \le i \le m-1, 1 \le j \le n-1 & (22c) \\ u(x_i,y_j) = \varphi_{i,j} \quad i,j \in \gamma & (22d) \end{cases}$$

截断误差为 $O(h^4)$.
将九点差分格式进行整理

$$\begin{aligned} f_{i,j} + G =& -k_1u_{i-1,j-1} + (2k_1 - k2)u_{i-1,j} - k_1u_{i-1,j+1} + (2k_1 - k_3)u_{i,j-1} + \\ & (2k_2 + 2k_3 - 4k_1)u_{i,j} + (2k_1 - k_3)u_{i,j+1} - k_1u_{i+1,j-1} + \\ & (2k_1 - k_2)u_{i+1,j} - k_1u_{i+1,j+1} \end{aligned} \tag{23}$$

其中，$k_1 = \dfrac{h_1^2+h_2^2}{12h_1^2h_2^2}$ $\quad ,k_2 = \dfrac{1}{h_1^2}$ $\quad ,k_3 = \dfrac{1}{h_2^2}$
将 (23) 写成矩阵形式

$$Au = f \tag{24}$$

其中

$$A = \begin{bmatrix} D & C & & \\ C & D & C & \\ & \ddots & \ddots & \ddots \\ & & C & D \end{bmatrix}$$

$$D = \begin{bmatrix} 2k_2+2k_3-4k_1 & 2k_1-k_3 & & & \\ 2k_1-k_3 & 2k_2+2k_3-4k_1 & -k_1 & & \\ & \ddots & \ddots & \ddots & \\ & & & 2k_1-k_3 & 2k_2+2k_3-4k_1 \end{bmatrix}$$

$$C = \begin{bmatrix} 2k_1-k_2 & -k_1 & & \\ -k_1 & 2k_1-k_2 & -k_1 & \\ & \ddots & \ddots & \ddots \\ & & -k_1 & 2k_1-k_2 \end{bmatrix}$$

$$u = \begin{bmatrix} u_{1,1}, u_{1,2}, \cdots, u_{1,n-1} \\ u_{2,1}, u_{2,2}, \cdots, u_{2,n-1} \\ \vdots \\ u_{m-2,1}, u_{m-2,2}, \cdots, u_{m-2,n-1} \\ u_{m-1,1}, u_{m-1,2}, \cdots, u_{m-1,n-1} \end{bmatrix}$$

f 要根据边界条件具体确定

解 (24), 即可得数值解。

三、 数值例子

数值例子为

$$\begin{cases} -(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}) = (\pi^2-1)e^x sin(\pi y) & , \quad 0<x<2, 0<y<1 \\ u(0,y) = sin(\pi y) \quad u(2,y) = e^2 sin(\pi y) & , \quad 0 \le y \le 1 \\ u(x,0) = 0, u(x,1) = 0, & , \quad 0 \le x \le 2 \end{cases} \tag{25}$$

该问题的精确解为 $e^x sin(\pi y)$.

定义误差为

$$E_\infty(h_1,h_2) = \max_{\substack{1\le i\le N_1-1 \\ 1\le j\le N_2-1}} |u(x_i,y_j) - u_{i,j}|$$

现在分别使用两种数值格式对该例子进行求解，解题程序运行于 Matlab 2018a.

3.1 五点差分格式

当 $h_1 = \dfrac{1}{16} \quad h_2 = \dfrac{1}{16}$ 时的数值解和精确解见图1和2, 从图像上看很接近。

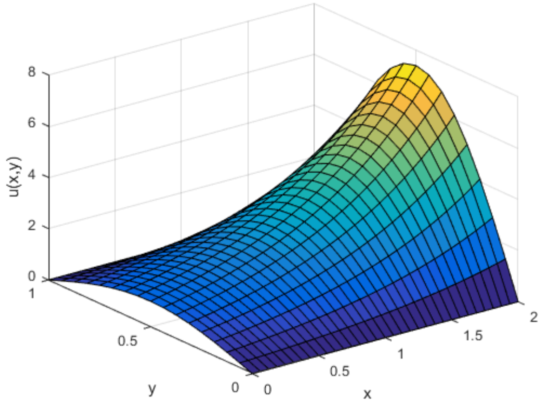

图 **1** $\quad h_1 = \dfrac{1}{16} \quad h_2 = \dfrac{1}{16}$ 时的精确解

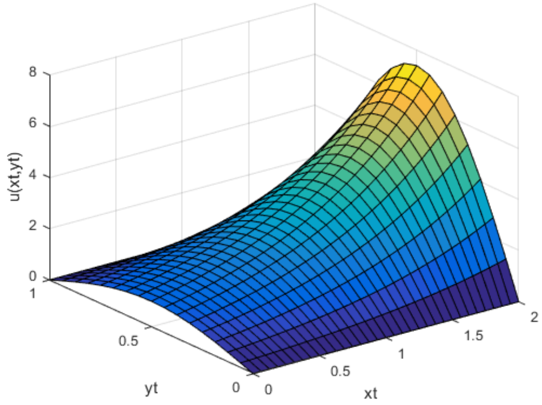

图 **2** $\quad h_1 = \dfrac{1}{16} \quad h_2 = \dfrac{1}{16}$ 时的数值解

表 1 是不同步长下，部分节点的精确解和数值解的具体数值，我们看到随着步长的减小，数值解越接近精确解。

表 **1** 不同步长下部分节点的精确解与数值解

(h_1,h_2)	x=1/8,y=1/2	x=3/8,y=1/2	x=5/8,y=1/2	x=7/8,y=1/2	x=9/8,y=1/2
1/8,1/8	1.1396038	1.4710228	1.8919087	2.4296181	3.1175586
1/16,1/16	1.1347613	1.4589884	1.8741409	2.4065360	3.0895320
1/32,1/32	1.1335502	1.4559860	1.8697125	2.4007821	3.0825377
1/64,1/64	1.1332460	1.4552319	1.8686005	2.3993379	3.0807831
精确解	1.1331480	1.4549919	1.8682465	2.3988759	3.0802171

取不同 h_1 和 h_2 时的误差图3, 图4, 图5, 图6, 步长越小，误差也越小。取不同步长时，最大误差和误差阶见表2，误差阶为 2 阶，符合 $O(h^2)$ 的截断误差

3.2 九点差分格式

当 $h_1 = \dfrac{1}{16} \quad h_2 = \dfrac{1}{16}$ 时的数值解和精确解见图7和8, 从图像上看很接近。

表3是不同步长下，部分节点的精确解和数值解的具体数值，我们看到随着步长的减小，数值解越接近精确解。

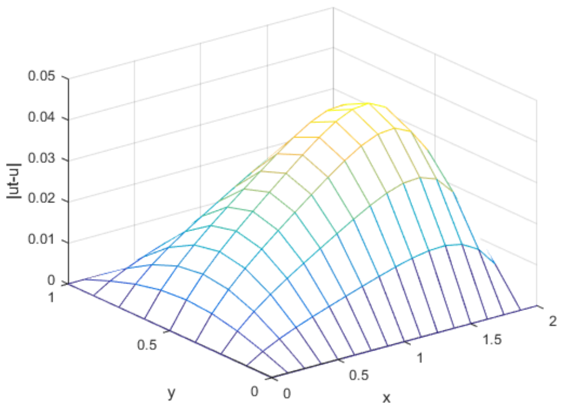

图 **3** $h_1=\frac{1}{8}$ $h_2=\frac{1}{8}$ 时的误差图

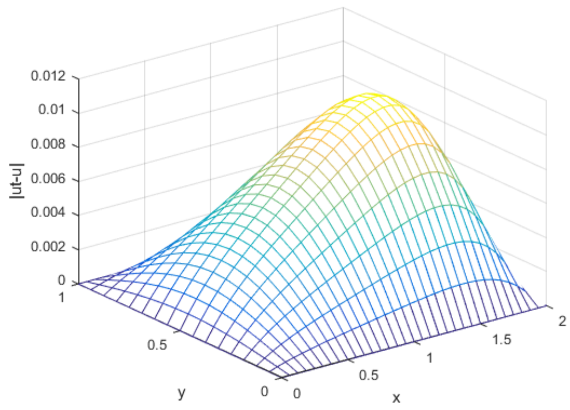

图 **4** $h_1=\frac{1}{16}$ $h_2=\frac{1}{16}$ 时的误差图

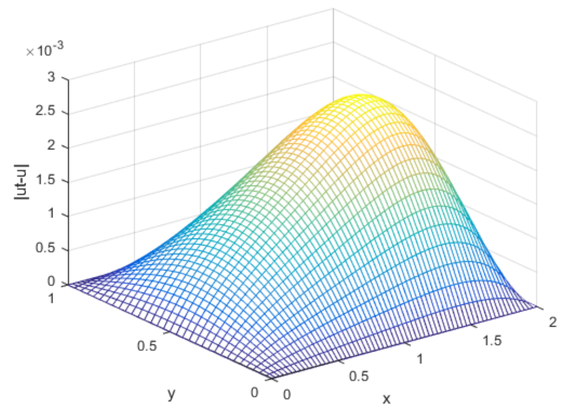

图 **5** $h_1=\frac{1}{32}$ $h_2=\frac{1}{32}$ 时的误差图

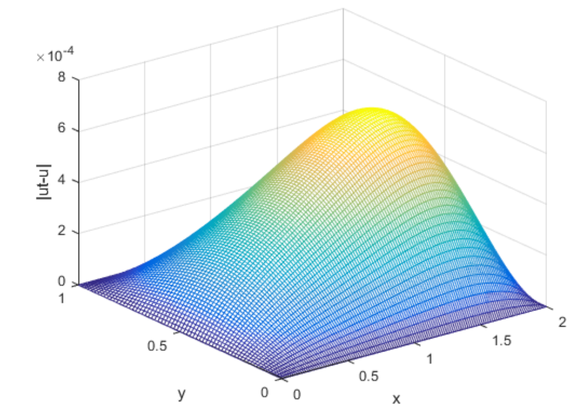

图 **6** $h_1=\frac{1}{64}$ $h_2=\frac{1}{64}$ 时的误差图

表 **2** 取不同步长是的最大误差和误差阶

(h_1,h_2)	E(h_1,h_2)	log(E$(2h_1,2h_2)$)
1/8,1/8	4.2377E-02	*
1/16,1/16	1.0605E-02	1.99854
1/32,1/32	2.6505E-03	2.00042
1/64,1/64	6.5230E-04	2.02264

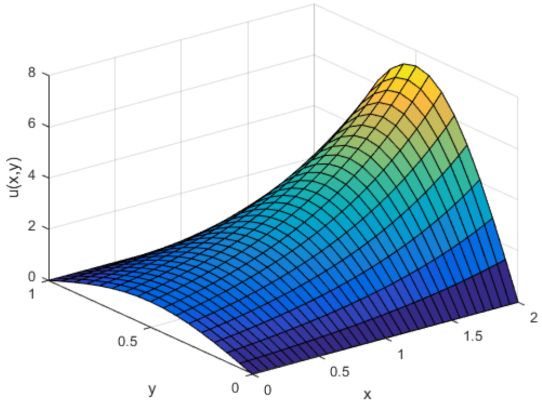

图 **7** $h_1 = \dfrac{1}{16}$ $h_2 = \dfrac{1}{16}$ 时的精确解

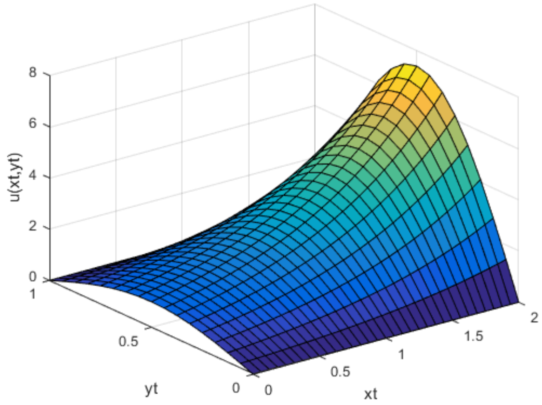

图 **8** $h_1 = \dfrac{1}{16}$ $h_2 = \dfrac{1}{16}$ 时的数值解

表 **3** 不同步长下部分节点的数值解与精确解 **(**$h_1 = h_2$**)**

(h_1,h_2)	x=1/8,y=1/2	x=3/8,y=1/2	x=5/8,y=1/2	x=7/8,y=1/2	x=9/8,y=1/2
1/8,1/8	1.1331304	1.4549467	1.8681801	2.3987897	3.0801127
1/16,1/16	1.1331473	1.4549886	1.8682419	2.3988700	3.0802104
1/32,1/32	1.1331484	1.4549912	1.8682457	2.3988750	3.0802164
1/64,1/64	1.1331484	1.4549914	1.8682459	2.3988753	3.0802168
精确解	1.1331485	1.4549914	1.8682460	2.3988753	3.0802168

取不同 h_1 和 h_2 时的误差图9, 图10, 图11, 图12, 步长越小，误差也越小。因此通过增加点数 (减小步长)，可以增加值的精确性。

$h_1 = h_2$ 时的最大误差和误差阶见表4，误差阶为 4 阶，符合 $O(h^4)$ 的截断误差

表 **4** 九点差分下的最大误差及误差阶 **(**$h_1 = h_2$**)**

(h_1, h_2)	E(h_1, h_2)	log(E$(2h_1, 2h_2)$)
1/8,1/8	1.18580E-04	*
1/16,1/16	7.35587E-06	4.01082
1/32,1/32	4.59576E-07	4.00052
1/64,1/64	2.87191E-08	4.00022

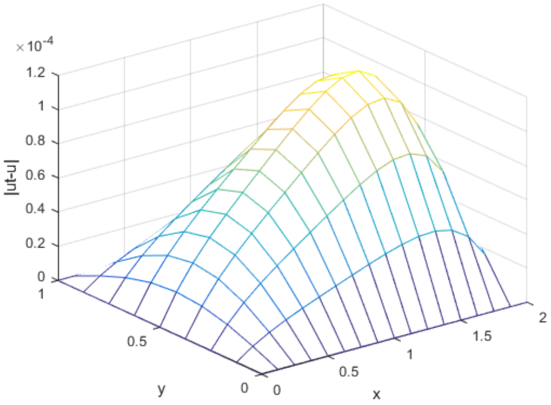

图 9　$h_1 = \dfrac{1}{8}$　$h_2 = \dfrac{1}{8}$ 时的误差图

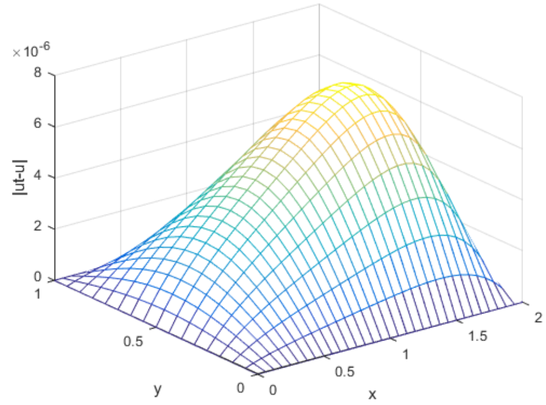

图 10　$h_1 = \dfrac{1}{16}$　$h_2 = \dfrac{1}{16}$ 时的误差图

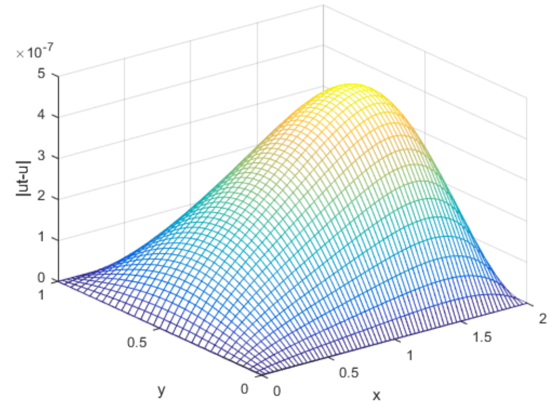

图 11　$h_1 = \dfrac{1}{32}$　$h_2 = \dfrac{1}{32}$ 时的误差图

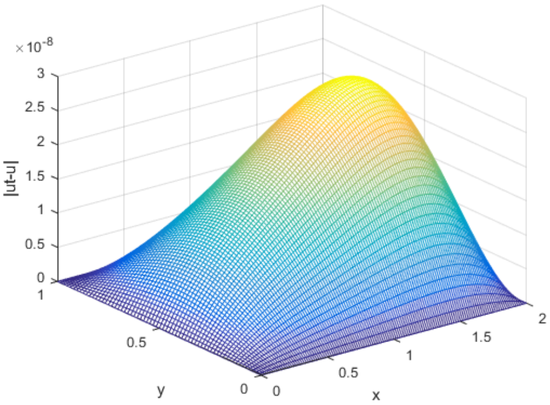

图 12　$h_1 = \dfrac{1}{64}$　$h_2 = \dfrac{1}{64}$ 时的误差图

同样我们还做了 $h_1>h_2$ 和 $h_1<h_2$ 时的最大误差和误差阶见表5和表6。

表 **5**　九点差分下的最大误差及误差阶 **(**$h_1>h_2$**)**

(h_1, h_2)	E(h_1, h_2)	log(E$(2h_1, 2h_2)$)
1/4,1/8	4.97132E-05	*
1/8,1/16	3.10557E-06	4.00070
1/16,1/32	1.94120E-07	3.99984
1/32,1/64	1.21572E-08	3.99706

通过表4，表5，表6可知, 误差阶接近 4 阶。并且随着步长的减小越精确。

表 **6**　九点差分下的最大误差及误差阶 **(h_1<h_2)**

(h_1, h_2)	E(h_1, h_2)	log(E$(2h_1, 2h_2)$)
1/8,1/4	2.49261E-03	*
1/16,1/8	1.50889E-04	4.04610
1/32,1/16	9.36919E-06	4.00942
1/64,1/32	5.84407E-07	4.00288

四、 总结

本文建立了两种二维椭圆型方程差分格式：五点差分格式和九点差分格式。用这两种格式，编写 matlab 程序，对具体的算例进行数值求解。五点差分格式的截断误差为 $O(h^2)$, 九点差分格式的截断误差为 $O(h^4)$，精度为：九点差分格式 > 五点差分格式。

参考文献

[1] 李荣华. 偏微分方程数值解法 [M]. 高等教育出版社, 2005.

[2] 孙志忠, 计算数学. 偏微分方程数值解法 [M]. 科学出版社, 2005.

附录 A　程序流程图

程序的流程见图13

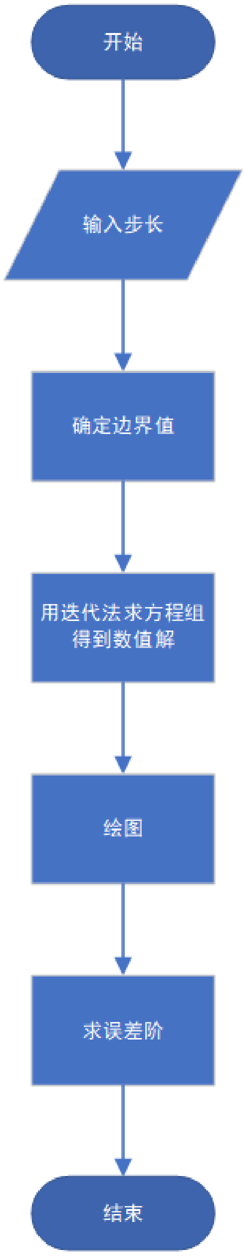

图 **13**　程序流程图

附录 B　五点差分格式代码

fu.m 求精确解

```
function fh=fu(x,y)
%精确解函数
```

```
fh=exp(x).*sin(pi.*y);
end
```

f.m 右端项

```
function ft= f(x,y)
%右端函数
ft=(pi^2-1)*exp(x)*sin(pi*y);
end
```

ft.m 用五点差分格式求数值解

```
%五点差分格式,高斯赛德尔求解
function uf = ft(M,N)
%步长
h1=2/M;
h2=1/N;
%内节点
x=h1:h1:2-h1;
y=h2:h2:1-h2;
uf=ones(M+1,N+1);
u0=zeros(M+1,N+1);
u1=zeros(M+1,N+1);
%y边值
y0=0:h2:1;
u0(1,:)=sin(pi*y0);
u0(M+1,:)=exp(2)*sin(pi*y0);
%x边值
u0(:,1)=0;
u0(:,N+1)=0;
u1=u0;
while(1)
for m=1:M-1
for n=1:N-1
u1(m+1,n+1)=(f(x(m),y(n))+u1(m+1,n)/h2^2+u1(m,n+1)/h1^2+...
u0(m+2,n+1)/h1^2+u0(m+1,n+2)/h2^2)/(2*(1/h1^2+1/h2^2));
end
end
if norm(u1-u0,'inf')<1e-6
uf=u1';break;
else
u0=u1;
end
end
```

wudianchafen.m 绘制误差图，精确解和数值解对比图以及误差和误差阶

```
clc;clear;
%m,n为等分大小
m=[16,32,64,128];
n=[8,16,32,64];
%步长1/8、1/16、1/32、1/64
x1=0:2/m(1):2;x2=0:2/m(2):2;x3=0:2/m(3):2;x4=0:2/m(4):2;
y1=0:1/n(1):1;y2=0:1/n(2):1;y3=0:1/n(3):1;y4=0:1/n(4):1;
%精确解图像
figure(1)
%使用散点绘图，m=32,n=16的精确解图像
[X,Y]=meshgrid(x2,y2);
u=fu(X,Y);
surf(X,Y,u);
xlabel('x');ylabel('y');zlabel('u(x,y)');
%m=32,n=16的数值解图像
[Xt,Yt]=meshgrid(x2,y2);
ut=ft(m(2),n(2));
figure(2);
surf(Xt,Yt,ut);
xlabel('xt');ylabel('yt');zlabel('u(xt,yt)');
%误差以及误差阶
[X1,Y1]=meshgrid(x1,y1);
[X2,Y2]=meshgrid(x2,y2);
[X3,Y3]=meshgrid(x3,y3);
[X4,Y4]=meshgrid(x4,y4);
pt1=abs(fu(X1,Y1)-ft(m(1),n(1)));%误差
pt2=abs(fu(X2,Y2)-ft(m(2),n(2)));
pt3=abs(fu(X3,Y3)-ft(m(3),n(3)));
pt4=abs(fu(X4,Y4)-ft(m(4),n(4)));
%误差曲面图
figure(3)
mesh(X1,Y1,pt1);
xlabel('x');ylabel('y');zlabel('|ut-u|');
figure(4)
mesh(X2,Y2,pt2);
xlabel('x');ylabel('y');zlabel('|ut-u|');
figure(5)
mesh(X3,Y3,pt3);
xlabel('x');ylabel('y');zlabel('|ut-u|');
figure(6)
mesh(X4,Y4,pt4);
xlabel('x');ylabel('y');zlabel('|ut-u|');
%最大误差
maxpt1=max(pt1(:));
maxpt2=max(pt2(:));
maxpt3=max(pt3(:));
maxpt4=max(pt4(:));
```

```
%误差比例
rate1=log2(maxpt1/maxpt2);
rate2=log2(maxpt2/maxpt3);
rate3=log2(maxpt3/maxpt4);
%x取1/8、3/8、5/8、7/8、9/8和y取1/2处的精确解和数值解
x0=[1/8,3/8,5/8,7/8,9/8];
y0=1/2;
%精确解
f0=null(1);
for i=1:5
f0(i)=fu(x0(i),y0);
end
%数值解
ut1=ft(m(1),n(1))';
ut2=ft(m(2),n(2))';
ut3=ft(m(3),n(3))';
ut4=ft(m(4),n(4))';
utf=ones(4,5);%记录部分节点的值
for j=1:5
utf(1,j)=ut1(m(1)/8*j,n(1)/2+1);
utf(2,j)=ut2(m(2)/8*j-1,n(2)/2+1);
utf(3,j)=ut3(m(3)/8*j-3,n(3)/2+1);
utf(4,j)=ut4(m(4)/8*j-7,n(4)/2+1);
end
```

附录 C　九点差分格式代码

fu.m 求精确解

```
function fh=fu(x,y)
%精确解函数
fh=exp(x).*sin(pi.*y);
end
```

f.m 右端项

```
function ft= f(x,y)
%右端函数
ft=(pi^2-1)*exp(x)*sin(pi*y);
end
```

ft.m 用九点差分格式求数值解

```
%九点差分格式,高斯赛德尔求解
function uf = ft(M,N)
%步长
```

```
h1=2/M;
h2=1/N;
k1=(h1^2+h2^2)/(12*(h1^2)*(h2^2));
k2=1/h1^2;
k3=1/h2^2;
%内节点
x=h1:h1:2-h1;
y=h2:h2:1-h2;
uf=ones(M+1,N+1);
u0=zeros(M+1,N+1);
u1=zeros(M+1,N+1);
%y边值
y0=0:h2:1;
u0(1,:)=sin(pi*y0);
u0(M+1,:)=exp(2)*sin(pi*y0);
%x边值
u0(:,1)=0;
u0(:,N+1)=0;
u1=u0;
while(1)
for m=1:M-1
for n=1:N-1
u1(m+1,n+1)=(f(x(m),y(n))+1/12*(h1^2*f(x(m),y(n))+...
h2^2*(pi^2-pi^4)*exp(x(m))*sin(pi*y(n)))+k1*...
u1(m,n)-(2*k1-k2)*u1(m,n+1)+k1*u1(m,n+2)-...
(2*k1-k3)*u1(m+1,n)-(2*k1-k3)*u0(m+1,n+2)+...
k1*u0(m+2,n)-(2*k1-k2)*u0(m+2,n+1)+...
k1*u0(m+2,n+2))/(2*k2+2*k3-4*k1);
end
end
if norm(u1-u0,'inf')<1e-12
uf=u1';break;
else
u0=u1;
end
end
```

jiudianchafen.m 绘制误差图，精确解和数值解对比图以及误差和误差阶

```
clc;clear;
m=[16,32,64,128];
n=[8,16,32,64];
% m=[8,16,32,64];
%n=[8,16,32,64];
%m=[16,32,64,128];
%n=[4,8,16,32];
%步长1/8、1/16、1/32、1/64
```

```
x1=0:2/m(1):2;x2=0:2/m(2):2;x3=0:2/m(3):2;x4=0:2/m(4):2;
y1=0:1/n(1):1;y2=0:1/n(2):1;y3=0:1/n(3):1;y4=0:1/n(4):1;
%精确解图像
figure(1)
%使用散点绘图，m=32,n=16的精确解图像
[X,Y]=meshgrid(x2,y2);
u=fu(X,Y);
surf(X,Y,u);
xlabel('x');ylabel('y');zlabel('u(x,y)');
%m=32,n=16的数值解图像
[Xt,Yt]=meshgrid(x2,y2);
ut=ft(m(2),n(2));
figure(2);
surf(Xt,Yt,ut);
xlabel('xt');ylabel('yt');zlabel('u(xt,yt)');
%误差以及误差阶
[X1,Y1]=meshgrid(x1,y1);
[X2,Y2]=meshgrid(x2,y2);
[X3,Y3]=meshgrid(x3,y3);
[X4,Y4]=meshgrid(x4,y4);
%的误差
pt1=abs(fu(X1,Y1)-ft(m(1),n(1)));
pt2=abs(fu(X2,Y2)-ft(m(2),n(2)));
pt3=abs(fu(X3,Y3)-ft(m(3),n(3)));
pt4=abs(fu(X4,Y4)-ft(m(4),n(4)));
%误差曲面图
figure(3)
mesh(X1,Y1,pt1);
xlabel('x');ylabel('y');zlabel('|ut-u|');
figure(4)
mesh(X2,Y2,pt2);
xlabel('x');ylabel('y');zlabel('|ut-u|');
figure(5)
mesh(X3,Y3,pt3);
xlabel('x');ylabel('y');zlabel('|ut-u|');
figure(6)
mesh(X4,Y4,pt4);
xlabel('x');ylabel('y');zlabel('|ut-u|');
%h1=h2最大误差
maxpt1=max(pt1(:));
maxpt2=max(pt2(:));
maxpt3=max(pt3(:));
maxpt4=max(pt4(:));
%误差比例
rate1=log2(maxpt1/maxpt2);
rate2=log2(maxpt2/maxpt3);
rate3=log2(maxpt3/maxpt4);
```

```
%x取1/8、3/8、5/8、7/8、9/8和y取1/2处的精确解和数值解
x0=[1/8,3/8,5/8,7/8,9/8];
y0=1/2;
%精确解
f0=null(1);
for i=1:5
f0(i)=fu(x0(i),y0);
end
%数值解
ut1=ft(m(1),n(1))';
ut2=ft(m(2),n(2))';
ut3=ft(m(3),n(3))';
ut4=ft(m(4),n(4))';
utf=ones(4,5);%记录部分节点的值
for j=1:5
utf(1,j)=ut1(m(1)/8*j,n(1)/2+1);
utf(2,j)=ut2(m(2)/8*j-1,n(2)/2+1);
utf(3,j)=ut3(m(3)/8*j-3,n(3)/2+1);
utf(4,j)=ut4(m(4)/8*j-7,n(4)/2+1);
end
```