



中南林业科技大学  
Central South University of Forestry & Technology

# 《微分方程数值解》 课程论文

专业年级： 2017 信息与计算科学专业

指导教师： 郑璇

开课时间： 2020 年上学期（春）

评阅教师： 郑璇

复核教师：

开课学院： 理学院

提交时间： 2020 年 5 月

中南林业科技大学理学院信计教研室 制

# 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

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

和 Laplace 方程

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

其中  $\Omega$  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

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

$$u = \varphi(x, y), \quad (x, y) \in \tau \quad (3b)$$

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  $\Omega$  为矩形区域

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

## 二、数值格式

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

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

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

$$x = x_i, 0 \leq i \leq m \quad (5)$$

$$y = y_j, 0 \leq j \leq n \quad (6)$$

将区域  $\Omega$  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  $\gamma$ , 其余的节点为内节点, 记为  $\omega$ 。

## 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

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

$$(7b)$$

用二阶中心差商代替二阶导数

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

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

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

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

差分算子  $-\Delta h$  的截断误差

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

去掉截断误差, 然后加上边值条件, 得到如下差分格式

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

$$(12b)$$

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

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

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

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

我们将 (12a) 整理得

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

写成矩阵形式

$$Au = f \quad (16)$$

其中

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

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

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

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

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

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

含由  $h^2$  的部分经转化变为

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

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

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

使用二阶中心差商代替二阶导数，所以

$$\begin{cases} a = \frac{u(x_{i+1}, y_{j+1}) - 2u(x_i, y_{j+1}) + u(x_{i-1}, y_{j+1})}{h_1^2} \end{cases} \quad (20a)$$

$$\begin{cases} b = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h_1^2} \end{cases} \quad (20b)$$

$$\begin{cases} c = \frac{u(x_{i+1}, y_{j-1}) - 2u(x_i, y_{j-1}) + u(x_{i-1}, y_{j-1})}{h_1^2} \end{cases} \quad (20c)$$

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

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

略去误差项，加上边值条件，得 Poisson 方程九点差分格式

$$\begin{cases} -\left[ \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h_1^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_2^2} \right] \frac{h_1^2 + h_2^2}{h_1^2 h_2^2} = f_{i,j} + G \end{cases} \quad (22a)$$

$$\begin{cases} G = \frac{1}{12} [h_1^2 f''_{xx}(x_i, y_j) + h_2^2 f''_{yy}(x_i, y_j)] \end{cases} \quad (22b)$$

$$\begin{cases} 1 \leq i \leq m-1, 1 \leq j \leq n-1 \end{cases} \quad (22c)$$

$$\begin{cases} u(x_i, y_j) = \varphi_{i,j} \quad i, j \in \gamma \end{cases} \quad (22d)$$

截断误差为  $O(h^4)$ 。

将九点差分格式进行整理

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

其中， $k_1 = \frac{h_1^2 + h_2^2}{12h_1^2 h_2^2}$ ， $k_2 = \frac{1}{h_1^2}$ ， $k_3 = \frac{1}{h_2^2}$

将 (23) 写成矩阵形式

$$Au = f \quad (24)$$

其中

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

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

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

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

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

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

该问题的精确解为  $e^x \sin(\pi y)$ .

定义误差为

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

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

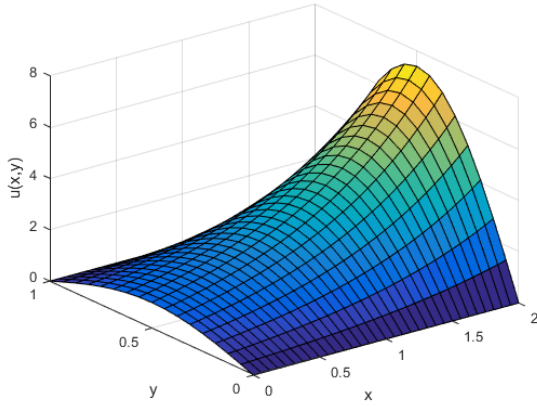


图 1  $h_1 = \frac{1}{16}$   $h_2 = \frac{1}{16}$  时的精确解

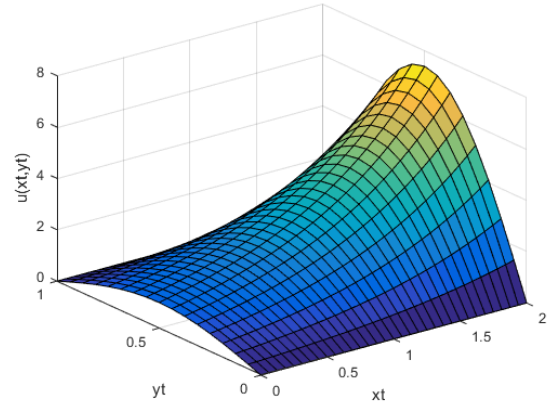


图 2  $h_1 = \frac{1}{16}$   $h_2 = \frac{1}{16}$  时的数值解

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  $h_1 = \frac{1}{16}$   $h_2 = \frac{1}{16}$  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。

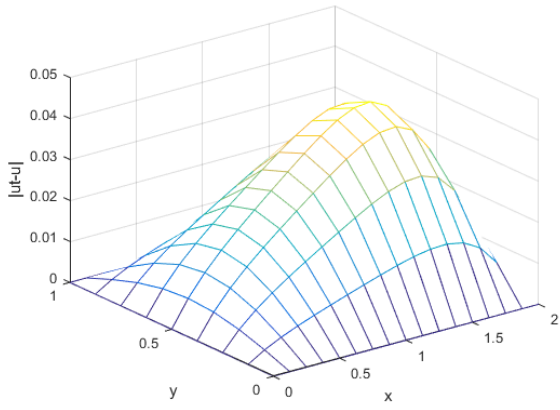


图 3  $h_1 = \frac{1}{8}$   $h_2 = \frac{1}{8}$  时的误差图

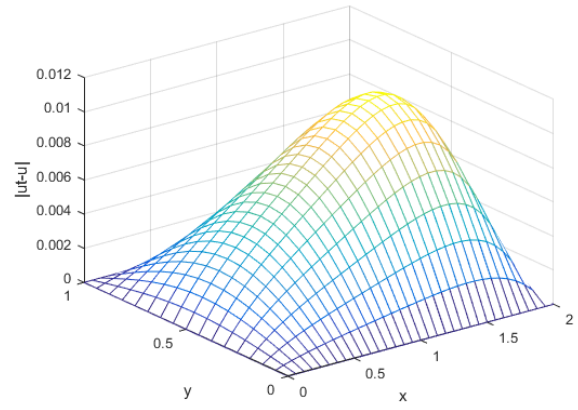


图 4  $h_1 = \frac{1}{16}$   $h_2 = \frac{1}{16}$  时的误差图

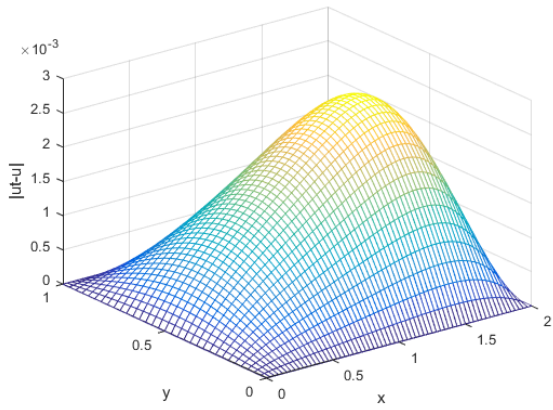


图 5  $h_1 = \frac{1}{32}$   $h_2 = \frac{1}{32}$  时的误差图

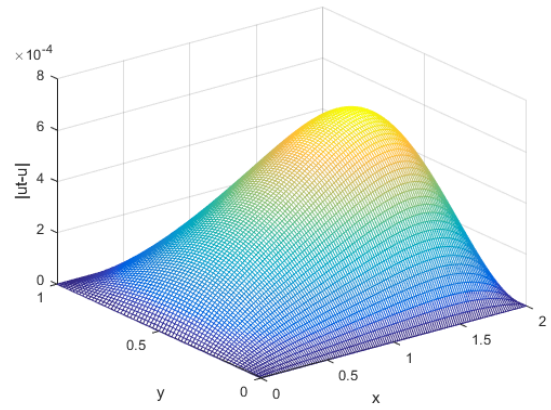


图 6  $h_1 = \frac{1}{64}$   $h_2 = \frac{1}{64}$  时的误差图

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               |



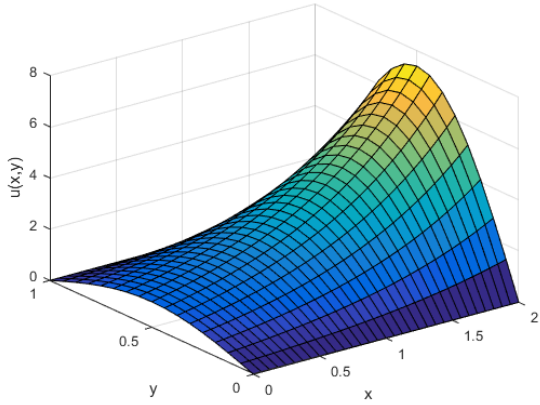


图 7  $h_1 = \frac{1}{16}$   $h_2 = \frac{1}{16}$  时的精确解

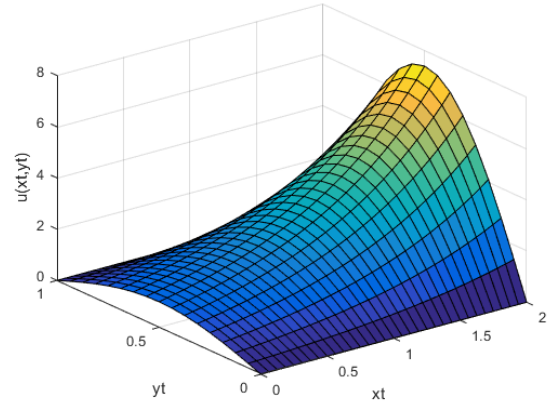


图 8  $h_1 = \frac{1}{16}$   $h_2 = \frac{1}{16}$  时的数值解

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               |

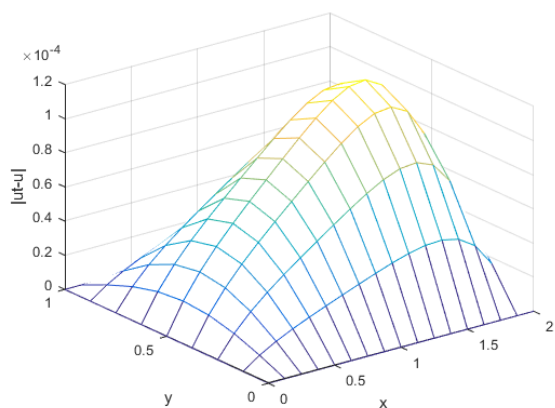


图 9  $h_1 = \frac{1}{8}$   $h_2 = \frac{1}{8}$  时的误差图

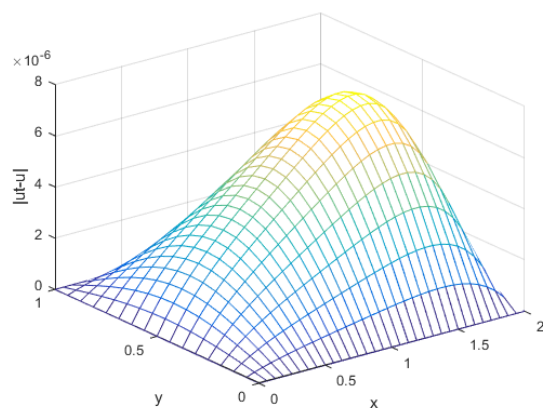


图 10  $h_1 = \frac{1}{16}$   $h_2 = \frac{1}{16}$  时的误差图

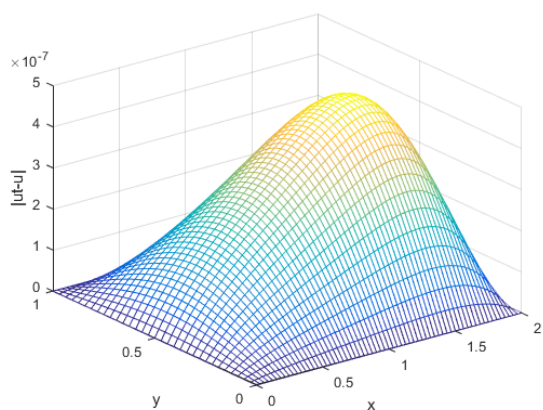


图 11  $h_1 = \frac{1}{32}$   $h_2 = \frac{1}{32}$  时的误差图

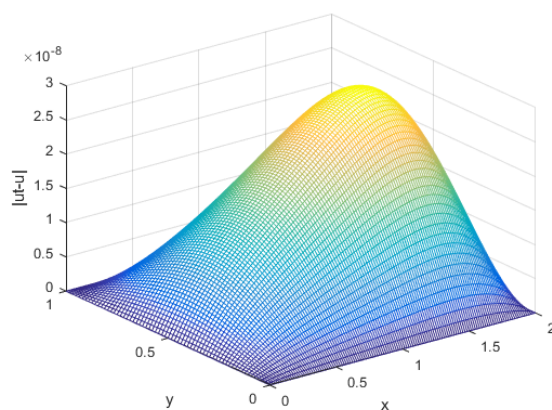


图 12  $h_1 = \frac{1}{64}$   $h_2 = \frac{1}{64}$  时的误差图

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

```

# 一维抛物型方程的差分格式

肖名财 刘礼海

## 一、引言

在研究热传导过程、气体膨胀过程和电磁场的传播等问题时，常常会遇到抛物型偏微分方程。这类方程的自变量中有一个是实际中的时间变量，用  $t$  表示。本文研究一维非齐次热传导方程的 Dirichlet 初边值问题：

$$\begin{cases} \frac{\partial u}{\partial t} - a \frac{\partial^2 u}{\partial x^2} = f(x, t), 0 \leq x \leq 1, 0 \leq t \leq T & (1a) \\ u(x, 0) = \varphi(x), 0 \leq x \leq 1 & (1b) \\ u(0, t) = \alpha(t), u(1, t) = \beta(t), 0 \leq t \leq 1 & (1c) \end{cases}$$

其中  $a$  为正常数， $f(x, t), \varphi(t), \alpha(t), \beta(t)$  为已知函数， $\varphi(0) = \alpha(0), \varphi(1) = \beta(0)$  称 (1b) 为初值条件，(1c) 为边值条件。

## 二、数值格式

首先，将区间  $[0, 1]$  作  $m$  等分，将区间  $[0, T]$  作  $n$  等分，记  $h = \frac{1}{m}, \tau = \frac{T}{n}, x_i = ih, 0 \leq i \leq m, t_k = k\tau, 0 \leq k \leq n$

分别称  $h$  和  $\tau$  为空间步长和时间步长。

称在  $t=0, x=0, x=1$  处的节点为内节点，其他节点为边界节点，称在直线  $t = t_k$  上的所有节点  $(x_i, t_k) | 0 \leq i \leq m$  为第  $k$  层节点。

### 2.1 向后 Euler 格式

在节点  $(x_i, t_k)$  处考虑微分方程 (1a)，有

$$\frac{\partial u}{\partial t}(x_i, t_k) - a \frac{\partial^2 u}{\partial x^2}(x_i, t_k) = f(x_i, t_k), 1 \leq i \leq m-1, 0 \leq k \leq n-1 \quad (2)$$

用二阶中心差商代替二阶导数

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_k) = \frac{1}{h^2} [u(x_{i-1}, t_k) - 2u(x_i, t_k) + u(x_{i+1}, t_k)] + O(h^2) \quad (3)$$

用向后差商代替一阶导数

$$\frac{\partial u}{\partial t}(x_i, t_k) = \frac{u(x_i, t_k) - u(x_i, t_{k-1})}{\tau} + O(\tau) \quad (4)$$

将 (3) 和 (4) 带入 (2) 式, 得

$$\frac{u(x_i, t_k) - u(x_i, t_{k-1})}{\tau} - a \frac{u(x_{i-1}, t_k) - 2u(x_i, t_k) + u(x_{i+1}, t_k))}{h^2} = f(x_i, t_k) + O(h^2 + \tau) \quad (5)$$

截断误差为  $O(h^2 + \tau)$ .

在 (5) 中略去截断误差, 得差分格式

$$\begin{cases} \frac{u(x_i, t_k) - u(x_i, t_{k-1})}{\tau} - a \frac{u(x_{i-1}, t_k) - 2u(x_i, t_k) + u(x_{i+1}, t_k))}{h^2} = f(x_i, t_k) & (6a) \end{cases}$$

$$\begin{cases} u_i^0 = \varphi(x_i), 0 \leq i \leq m & (6b) \end{cases}$$

$$\begin{cases} u_0^k = \alpha(t_k), u_m^k = \beta(t_k), 1 \leq k \leq n & (6c) \end{cases}$$

称 (6a)-(6c) 为向后 Euler 格式.

记  $\gamma = \frac{a\tau}{h^2}$ , 称  $\gamma$  为步长比, (6a) 整理得

$$-\gamma u_{i-1}^k + (1 + 2\gamma)u_i^k - \gamma u_{i+1}^k = u_i^{k-1} + \tau f(x_i, t_k), 1 \leq i \leq m-1, 0 \leq k \leq n-1 \quad (7)$$

写成矩阵形式

$$Au = f \quad (8)$$

其中

$$A = \begin{bmatrix} 1+2\gamma & -\gamma & & & \\ -\gamma & 1+2\gamma & -\gamma & & \\ & & \ddots & \ddots & \ddots \\ & & & -\gamma & 1+2\gamma \end{bmatrix}$$

$$u = \begin{bmatrix} u_1^k \\ u_2^k \\ \vdots \\ u_{m-2}^k \\ u_{m-1}^k \end{bmatrix}$$

$$f = \begin{bmatrix} u_1^{k-1} + \gamma u_0^k \\ u_2^{k-1} \\ \vdots \\ u_{m-2}^{k-1} \\ u_{m-1}^{k-1} + \gamma u_m^k \end{bmatrix} + \begin{bmatrix} \tau f(x_1, t_k) \\ \tau f(x_2, t_k) \\ \vdots \\ \tau f(x_{m-2}, t_k) \\ \tau f(x_{m-1}, t_k) \end{bmatrix}$$

解方程 (8), 可以得到数值解。

## 2.2 Crank-Nicolson 格式

引入半整数点记  $t_{k+\frac{1}{2}} = \frac{1}{2}(t_k + t_{k+1})$

在点  $(x_i, t_{k+\frac{1}{2}})$  处考虑微分方程 (1a), 有

$$\frac{\partial u}{\partial t}(x_i, t_{k+\frac{1}{2}}) - a \frac{\partial^2 u}{\partial x^2}(x_i, t_k) = f(x_i, t_{k+\frac{1}{2}}), 1 \leq i \leq m-1, 0 \leq k \leq n-1 \quad (9)$$

由 Taylor 展开得

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_k) = \frac{\partial^2 u}{\partial x^2}(x_i, t_{k+\frac{1}{2}}) - \frac{\tau}{2} \frac{\partial^3 u}{\partial x^2 \partial t} + \frac{1}{2} \left(\frac{\tau}{2}\right)^2 \frac{\partial^4 u}{\partial x^2 \partial t^2} + O(\tau^3) \quad (10)$$

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_{k+1}) = \frac{\partial^2 u}{\partial x^2}(x_i, t_{k+\frac{1}{2}}) + \frac{\tau}{2} \frac{\partial^3 u}{\partial x^2 \partial t} + \frac{1}{2} \left(\frac{\tau}{2}\right)^2 \frac{\partial^4 u}{\partial x^2 \partial t^2} + O(\tau^3) \quad (11)$$

将 (10) 和 (11) 相加得

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_{k+\frac{1}{2}}) = \frac{1}{2} \left[ \frac{\partial^2 u}{\partial x^2}(x_i, t_{k+1}) + \frac{\partial^2 u}{\partial x^2}(x_i, t_k) \right] + O(\tau^2) \quad (12)$$

将 (12) 代入 (9) 得

$$\frac{\partial u}{\partial t}(x_i, t_{k+\frac{1}{2}}) - \frac{1}{2} a \left[ \frac{\partial^2 u}{\partial x^2}(x_i, t_{k+1}) + \frac{\partial^2 u}{\partial x^2}(x_i, t_k) \right] = f(x_i, t_{k+\frac{1}{2}}) + O(\tau^2) \quad (13)$$

用一阶中心差商近似一阶导数

$$\frac{\partial u}{\partial t}(x_i, t_{k+\frac{1}{2}}) = \frac{u(x_i, t_{k+1}) - u(x_i, t_k)}{\tau} + O(\tau^2) \quad (14)$$

用二阶中心差商近似二阶导数

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_k) = \frac{u(x_{i+1}, t_k) - 2u(x_i, t_k) + u(x_{i-1}, t_k))}{h^2} + O(h^2) \quad (15)$$

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_{k+1}) = \frac{u(x_{i+1}, t_{k+1}) - 2u(x_i, t_{k+1}) + u(x_{i-1}, t_{k+1}))}{h^2} + O(h^2) \quad (16)$$

将 (14)、(15) 和 (16) 代入 (13), 得

$$\begin{aligned} & \frac{u(x_i, t_{k+1}) - u(x_i, t_k)}{\tau} - \frac{a}{2} \left[ \frac{u(x_{i+1}, t_k) - 2u(x_i, t_k) + u(x_{i-1}, t_k))}{h^2} + \right. \\ & \left. \frac{u(x_{i+1}, t_{k+1}) - 2u(x_i, t_{k+1}) + u(x_{i-1}, t_{k+1}))}{h^2} \right] = f(x_i, t_{k+\frac{1}{2}}) + O(h^2 + \tau^2) \end{aligned} \quad (17)$$

略去截断误差, 从而可得差分方程

$$\begin{cases} \frac{u_i^{k+1} - u_i^k}{\tau} = \frac{a}{2} \left[ \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{h^2} + \frac{u_{i+1}^{k+1} - 2u_i^{k+1} + u_{i-1}^{k+1}}{h^2} \right] + f(x_i, t_{k+\frac{1}{2}}) \end{cases} \quad (18a)$$

$$\begin{cases} u_i^0 = \varphi(x_i), 0 \leq i \leq m \end{cases} \quad (18b)$$

$$\begin{cases} u_0^k = \alpha(t_k), u_m^k = \beta(t_k), 1 \leq k \leq n \end{cases} \quad (18c)$$

截断误差为  $O(\tau^2 + h^2)$ .

将 (18a) 写成矩阵形式

$$A_1 u^{k+1} = A_2 u^k + f \quad (19)$$

其中

$$A_1 = \begin{bmatrix} 1+\gamma & -\frac{\gamma}{2} & & & \\ -\frac{\gamma}{2} & 1+\gamma & -\frac{\gamma}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{\gamma}{2} & 1+\gamma & \\ & & & & \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1-\gamma & \frac{\gamma}{2} & & & \\ \frac{\gamma}{2} & 1-\gamma & \frac{\gamma}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{\gamma}{2} & 1-\gamma & \\ & & & & \end{bmatrix}$$

$$u^k = \begin{bmatrix} u_1^k \\ u_2^k \\ \vdots \\ u_{m-2}^k \\ u_{m-1}^k \end{bmatrix}$$

$$f = \begin{bmatrix} \frac{\gamma}{2}(u_0^k + u_0^{k+1}) + \tau f(x_1, t_{k+\frac{1}{2}}) \\ \tau f(x_2, t_{k+\frac{1}{2}}) \\ \vdots \\ \tau f(x_{m-2}, t_{k+\frac{1}{2}}) \\ \frac{\gamma}{2}(u_m^k + u_m^{k+1}) + \tau f(x_{m-1}, t_{k+\frac{1}{2}}) \end{bmatrix}$$

解 (19), 即可得数值解。

### 2.3 二阶 BDF 差分格式

当  $k=1$  时, 直接对 (1a) 运用向后 Euler 差分格式, 有

$$\frac{u_i^1 - u_i^0}{\tau} = a \frac{u_{i-1}^1 - 2u_i^1 + u_{i+1}^1}{h^2} + f_i^1, 1 \leq i \leq m-1 \quad (20)$$

截断误差  $R = O(h^2 + \tau)$

当  $k \geq 2$  时, 在点  $(x_i, t_k)$  处考虑方程 (1a), 有

$$\frac{\partial u}{\partial t}(x_i, t_k) = a \frac{\partial^2 u}{\partial t^2}(x_i, t_k) + f(x_i, t_k) \quad (21)$$

将  $u(x_i, t_{k-1}), u(x_i, t_{k-2})$  在点  $(x_i, t_k)$  处 Taylor 展开, 有

$$u(x_i, t_{k-1}) = u(x_i, t_k) - \tau \frac{\partial u}{\partial t}(x_i, t_k) + \frac{1}{2!} \tau^2 \frac{\partial^2 u}{\partial t^2}(x_i, t_k) - \frac{1}{3!} \tau^3 \frac{\partial^3 u}{\partial t^3}(x_i, t_k) + O(\tau^3) \quad (22)$$

$$u(x_i, t_{k-2}) = u(x_i, t_k) - 2\tau \frac{\partial u}{\partial t}(x_i, t_k) + \frac{1}{2!} (2\tau)^2 \frac{\partial^2 u}{\partial t^2}(x_i, t_k) - \frac{1}{3!} (2\tau)^3 \frac{\partial^3 u}{\partial t^3}(x_i, t_k) + O(\tau^3) \quad (23)$$

(23) - 4 × (22) 得

$$\frac{u(x_i, t_{k-2}) - 4u(x_i, t_{k-1}) + 3u(x_i, t_k)}{2\tau} = \frac{\partial u}{\partial t}(x_i, t_k) + O(\tau^2) \quad (24)$$

用二阶中心差商代替二阶导数

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_k) = \frac{u(x_{i+1}, t_k) - 2u(x_i, t_k) + u(x_{i-1}, t_k))}{h^2} + O(h^2) \quad (25)$$

将 (24) 和 (25) 代入方程 (21), 有

$$\begin{aligned} \frac{u(x_i, t_{k-2}) - 4u(x_i, t_{k-1}) + 3u(x_i, t_k)}{2\tau} &= a \frac{u(x_{i+1}, t_k) - 2u(x_i, t_k) + u(x_{i-1}, t_k))}{h^2} \\ &+ f(x_i, t_k) + O(\tau^2 + h^2) \end{aligned} \quad (26)$$

所以差分格式为

$$\begin{cases} \frac{3u_i^k - 4u_i^{k-1} + u_i^{k-2}}{2\tau} = a \frac{u_{i-1}^k - 2u_i^k + u_{i+1}^k}{h^2} + f_i^k, k \geq 2, 1 \leq i \leq m-1 \end{cases} \quad (27a)$$

$$\begin{cases} \frac{u_i^1 - u_i^0}{\tau} = a \frac{u_{i-1}^1 - 2u_i^1 + u_{i+1}^1}{h^2} + f_i^1, k = 1, 1 \leq i \leq m-1 \end{cases} \quad (27b)$$

$$\begin{cases} u_i^0 = \varphi(x_i), 1 \leq i \leq m, u_0^k = \alpha(t_k), u_m^k = \beta(t_k), 1 \leq k \leq n \end{cases} \quad (27c)$$

当  $k \geq 2$  时, 整理为

$$\gamma u_{i-1}^k - (2\gamma + \frac{3}{2})u_i^k + \gamma u_{i+1}^k = -2u_i^{k-1} + \frac{1}{2}u_i^{k-2} - \tau f_i^k \quad (28)$$

写成矩阵形式

$$Au^k = -2u^{k-1} + \frac{1}{2}u^{k-2} + f \quad (29)$$

其中

$$A = \begin{bmatrix} -(2\gamma + \frac{3}{2}) & \gamma & & & \\ \gamma & -(2\gamma + \frac{3}{2}) & \gamma & & \\ & & \ddots & \ddots & \ddots \\ & & & -(2\gamma + \frac{3}{2}) & \gamma \end{bmatrix}$$



$$u^k = \begin{bmatrix} u_1^k \\ u_2^k \\ \vdots \\ u_{m-2}^k \\ u_{m-1}^k \end{bmatrix}$$

$$f = \begin{bmatrix} -\gamma u_0^k - \tau f(x_1, t_k) \\ -\tau f(x_1, t_k) \\ \vdots \\ -\tau f(x_{m-2}, t_k) \\ -\gamma u_m^k - \tau f(x_{m-1}, t_k) \end{bmatrix}$$

解方程 (29), 即可得数值解。

### 三、数值例子

数值例子为

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} & , \quad 0 \leq x \leq 1, 0 \leq t \leq 1 \\ u(x, 0) = e^x & , \quad 0 \leq x \leq 1 \\ u(0, t) = e^t, u(1, t) = e^{1+t}, & , \quad 0 \leq t \leq 1 \end{cases} \quad (30)$$

该问题的精确解为  $u(x, t) = e^{x+t}$ .

定义误差为

$$E_\infty(h, \tau) = \max_{\substack{1 \leq i \leq m-1 \\ 1 \leq k \leq n}} |u(x_i, t_k) - u_i^k|$$

现在分别使用三种数值格式对该例子进行求解, 解题程序运行于 Matlab 2018a.

#### 3.1 向后 Euler 格式

当  $\tau = \frac{1}{100}, h = \frac{1}{10}$  时,  $t=1$  处的数值解和精确解见图1, 从图像上看很接近。

当  $\tau = \frac{1}{100}, h = \frac{1}{10}$  时, 取  $x=0.5$ , 不同  $t$  处的值见表1, 随着  $t$  的增大, 误差不断累积, 越来越大, 到  $t=1$  处误差变得最大。

取不同  $\tau$  和  $h$  时,  $t=1$  处的误差见图2, 步长越小, 误差也越小。

取不同步长时, 误差和误差比见表2,  $\tau$  变为原来的 4 倍,  $h$  变为原来的 2 倍, 误差会变为原来的 4 倍, 符合  $O(\tau + h^2)$  的截断误差

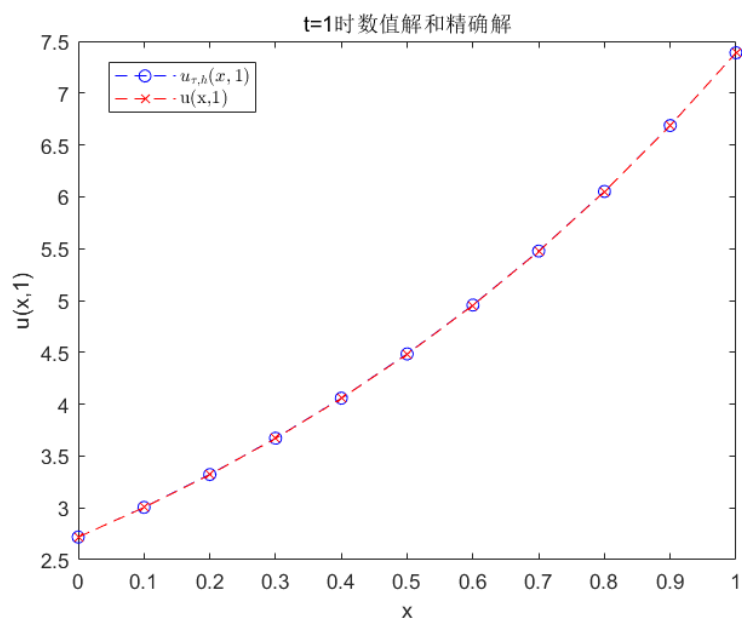


图 1  $t=1$  处的数值解和精确解 (向后 Euler 格式)

表 1  $x=0.5$  时, 不同  $t$  处的数值解、精确解和误差 (向后 Euler 格式)

| k   | t   | 数值解      | 精确解      | 误差         |
|-----|-----|----------|----------|------------|
| 10  | 0.1 | 1.822891 | 1.822119 | 7.7174E-04 |
| 20  | 0.2 | 2.014927 | 2.013753 | 1.1743E-03 |
| 30  | 0.3 | 2.226965 | 2.225541 | 1.4242E-03 |
| 40  | 0.4 | 2.461227 | 2.459603 | 1.6237E-03 |
| 50  | 0.5 | 2.720096 | 2.718282 | 1.8140E-03 |
| 60  | 0.6 | 3.006178 | 3.004166 | 2.0124E-03 |
| 70  | 0.7 | 3.322344 | 3.320117 | 2.2271E-03 |
| 80  | 0.8 | 3.671759 | 3.669297 | 2.4625E-03 |
| 90  | 0.9 | 4.057922 | 4.055200 | 2.7219E-03 |
| 100 | 1   | 4.484697 | 4.481689 | 3.0084E-03 |

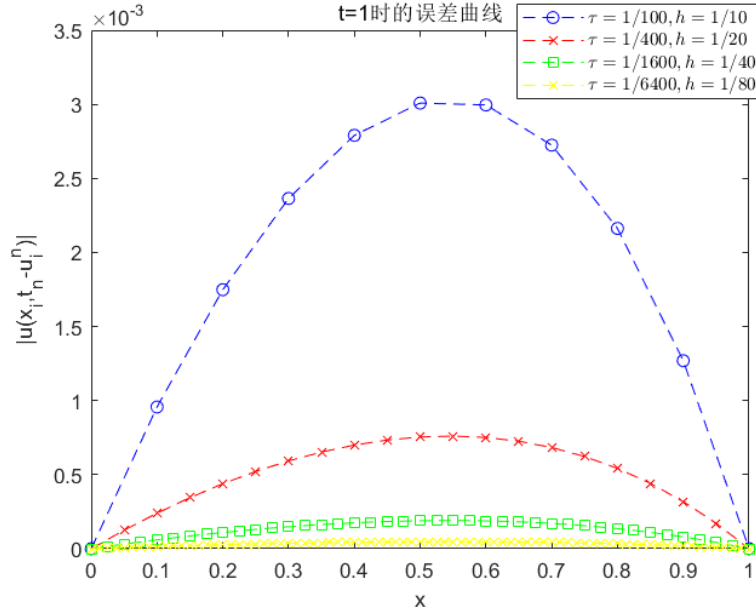


图 2 不同步长下的误差 (向后 Euler 格式)

表 2 取不同步长时的误差和误差比 (向后 Euler 格式)

| $h, \tau$    | $E_{\infty}(h, \tau)$ | $E_{\infty}(2h, 4\tau)/E_{\infty}(h, \tau)$ |
|--------------|-----------------------|---|
| 1/100, 1/10  | 3.0084E-03            | *   |
| 1/400, 1/20  | 7.6035E-04            | 3.956615                                    |
| 1/1600, 1/40 | 1.9023E-04            | 3.997046                                    |
| 1/6400, 1/80 | 4.7566E-05            | 3.999261                                    |

### 3.2 Crank-Nicolson 格式

当  $\tau = \frac{1}{10}, h = \frac{1}{10}$  时,  $t=1$  处的数值解和精确解见图3, 从图像上看很接近。

当  $\tau = \frac{1}{10}, h = \frac{1}{10}$  时, 取  $x=0.5$ , 不同  $t$  处的值见表3, 当层数越深时, 误差越大, 这是因为, 每一次由  $k$  层求解  $k+1$  层时都有误差, 随着  $t$  的增大, 误差不断累积, 越来越大, 到  $t=1$  处误差变得最大。

取不同  $\tau$  和  $h$  时,  $t=1$  处的误差见图4, 步长越小, 误差也越小。

使用 Crank-Nicolson 格式求数值解, 取不同步长时最大误差和最大误差的比值见表4,  $\tau$  变为原来的 2 倍,  $h$  变为原来的 2 倍, 误差会变为原来的 4 倍, 符合  $O(\tau^2 + h^2)$  的截断误差。

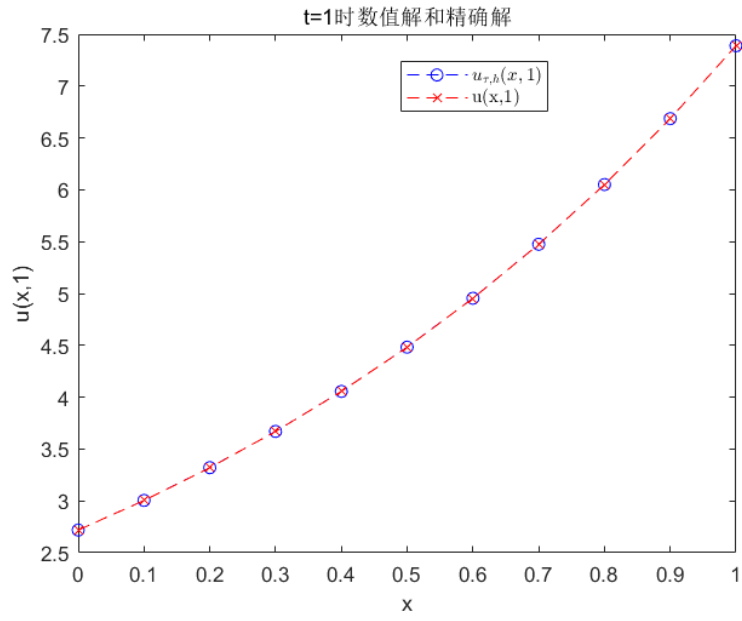


图 3  $t=1$  处的数值解和精确解 (Crank-Nicolson 格式)

表 3  $x=0.5$  时, 不同  $t$  处的数值解、精确解和误差 (Crank-Nicolson 格式)

| k  | t   | 数值解      | 精确解      | 误差         |
|----|-----|----------|----------|------------|
| 1  | 0.1 | 1.822349 | 1.822119 | 2.3053E-04 |
| 2  | 0.2 | 2.014105 | 2.013753 | 3.5224E-04 |
| 3  | 0.3 | 2.225953 | 2.225541 | 4.1241E-04 |
| 4  | 0.4 | 2.460072 | 2.459603 | 4.6922E-04 |
| 5  | 0.5 | 2.718802 | 2.718282 | 5.2042E-04 |
| 6  | 0.6 | 3.004743 | 3.004166 | 5.7700E-04 |
| 7  | 0.7 | 3.320755 | 3.320117 | 6.3794E-04 |
| 8  | 0.8 | 3.670002 | 3.669297 | 7.0507E-04 |
| 9  | 0.9 | 4.055979 | 4.055200 | 7.7949E-04 |
| 10 | 1   | 4.482550 | 4.481689 | 8.6123E-04 |

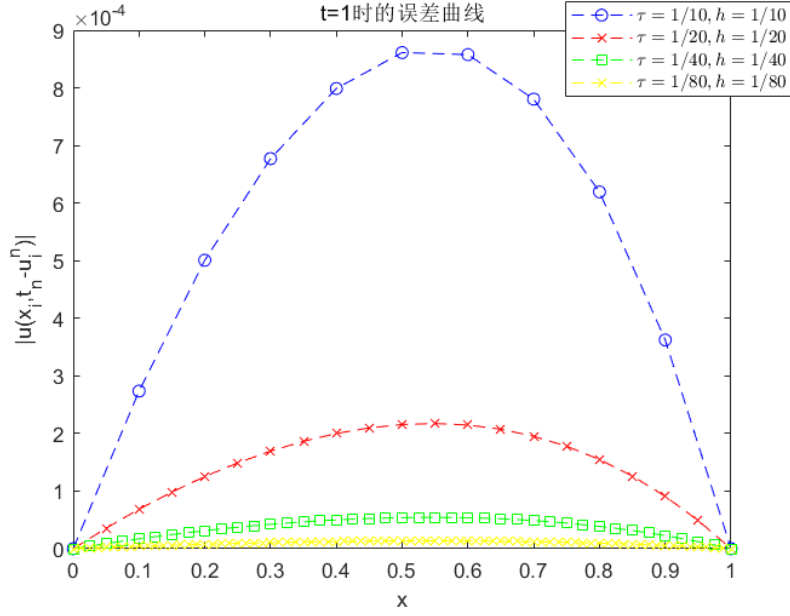


图 4 不同步长下的误差 (Crank-Nicolson 格式)

表 4 不同步长的最大误差和最大误差的比 (Crank-Nicolson 格式)

| $h, \tau$  | $E_{\infty}(h, \tau)$ | $E_{\infty}(2h, 2\tau)/E_{\infty}(h, \tau)$ |
|------------|-----------------------|---|
| 1/10, 1/10 | 8.61E-04              | *   |
| 1/20, 1/20 | 2.17E-04              | 3.962274                                    |
| 1/40, 1/40 | 5.44E-05              | 3.998647                                    |
| 1/80, 1/80 | 1.36E-05              | 3.999657                                    |

### 3.3 二阶 BDF 差分格式

当  $\tau = \frac{1}{10}, h = \frac{1}{10}$  时,  $t=1$  处的数值解和精确解见图5, 从图像上看很接近。

当  $\tau = \frac{1}{10}, h = \frac{1}{10}$  时, 取  $x=0.5$ , 不同  $t$  处的值见表5。

取不同  $\tau$  和  $h$  时,  $t=1$  处的误差见图6, 步长越小, 误差也越小。

取不同步长时, 误差和误差比见表6,  $\tau$  变为原来的 2 倍,  $h$  变为原来的 2 倍, 误差变为原来的 2 到 3 倍, 这是因为, 虽然  $k \geq 2$  时, 截断误差为  $O(\tau^2 + h^2)$ , 但是  $k = 1$  时, 截断误差为  $O(\tau + h^2)$ , 所以达不到 4 倍。

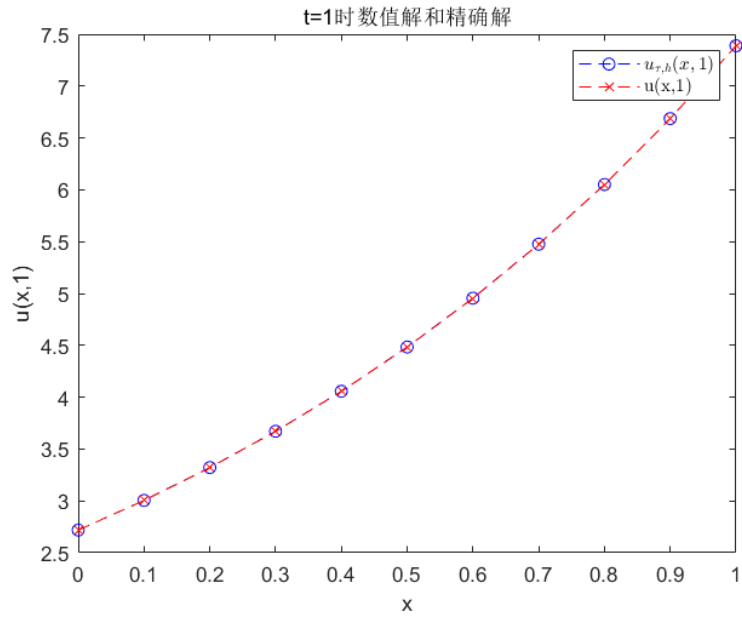


图 5  $t=1$  处的数值解和精确解 (二阶 **BDF** 差分格式)

表 5  $x=0.5$  时, 不同  $t$  处的数值解、精确解和误差 (二阶 **BDF** 差分格式)

| k  | t   | 数值解      | 精确解      | 误差       |
|----|-----|----------|----------|----------|
| 1  | 0.1 | 1.827620 | 1.822119 | 5.50E-03 |
| 2  | 0.2 | 2.018779 | 2.013753 | 5.03E-03 |
| 3  | 0.3 | 2.228923 | 2.225541 | 3.38E-03 |
| 4  | 0.4 | 2.461791 | 2.459603 | 2.19E-03 |
| 5  | 0.5 | 2.719898 | 2.718282 | 1.62E-03 |
| 6  | 0.6 | 3.005622 | 3.004166 | 1.46E-03 |
| 7  | 0.7 | 3.321620 | 3.320117 | 1.50E-03 |
| 8  | 0.8 | 3.670940 | 3.669297 | 1.64E-03 |
| 9  | 0.9 | 4.057022 | 4.055200 | 1.82E-03 |
| 10 | 1   | 4.483712 | 4.481689 | 2.02E-03 |

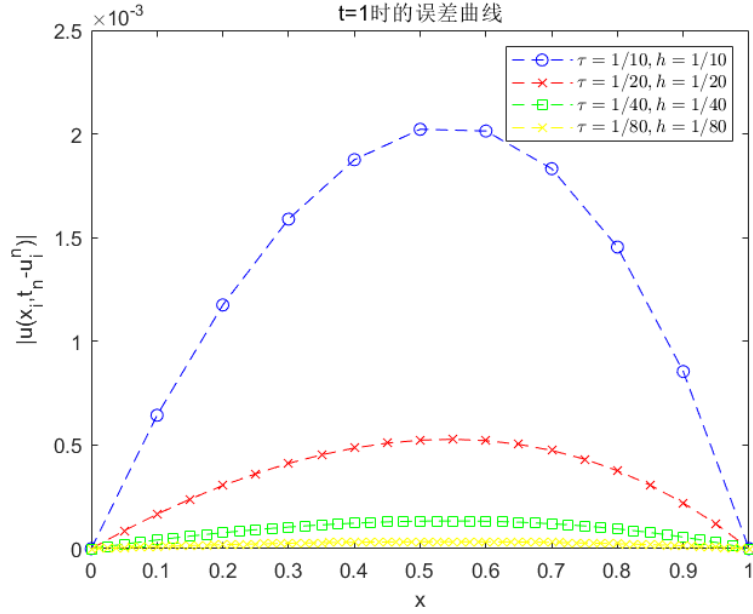


图 6 不同步长下的误差 (二阶 **BDF** 差分格式)

表 6 取不同步长时的误差和误差比 (二阶 **BDF** 差分格式)

| $h, \tau$  | $E_{\infty}(h, \tau)$ | $E_{\infty}(2h, 4\tau)/E_{\infty}(h, \tau)$ |
|------------|-----------------------|---|
| 1/10, 1/10 | 5.61E-03              | *   |
| 1/20, 1/20 | 1.90E-03              | 2.9544                                      |
| 1/40, 1/40 | 6.18E-04              | 3.0719                                      |
| 1/80, 1/80 | 1.81E-04              | 3.4202                                      |

## 四、总结

本文建立了三种一维抛物型方程差分格式：向后 Euler 格式、Crank-Nicolson 格式和二阶 BDF 格式。并运用这三种格式，通过编写 matlab 程序，对具体的算例进行了数值求解。向后 Euler 格式的截断误差为  $O(\tau + h^2)$ ，Crank-Nicolson 格式的截断误差为  $O(\tau^2 + h^2)$ ，而二阶 BDF 格式的截断误差则分为两段：当  $k = 1$  时，截断误差为  $O(\tau + h^2)$ ，当  $k \geq 2$  时，截断误差为  $O(\tau^2 + h^2)$ ，精度为：Crank-Nicolson 格式 > 二阶 BDF 格式 > 向后 Euler 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7



图 7 程序流程图

## 附录 B 向后 Euler 格式代码

fa.m 求精确解

```
function r =fa(t,x)
% 求精确解
```

```

% @t 时间向量
% @x 空间向量
[t,x]=meshgrid(t,x);
t=t';
x=x';
r=exp(x+t);
end

```

## fsolve.m 用向后 Euler 格式求数值解

```

function [t,x,u] = fsolve(tau,h)
% 用向后Euler法解抛物方程
% @t 时间向量
% @x 空间向量
% @tau 时间步长
% @h 空间步长
% @u 数值解
N=1/tau;%t被分割的区间数
M=1/h;%x被分割的区间数
t=0:tau:1;
x=0:h:1;
u=ones(N+1,M+1);
u(1,:)=exp(x);%初值
%边值
u(:,1)=exp(t);
u(:,M+1)=exp(1+t);

r=tau/h^2;%步长系数
a=ones(M-2,1)*(-r);%下对角线
c=ones(M-2,1)*(-r);%上对角线
b=ones(M-1,1)*(1+2*r);%主对角线
A=diag(b,0)+diag(a,-1)+diag(c,1);%系数矩阵
for k=2:N+1
%右端项
f=u(k-1,2:M);
f(1)=f(1)+r*u(k,1);
f(M-1)=f(M-1)+r*u(k,M+1);
u(k,2:M)=(A\f)';%由k-1层求第k层的值
end
end

```

## fig.m 绘制误差图，精确解和数值解对比图

```

clc;clear;
tau=[1/100,1/400,1/1600,1/6400];%时间步长
h=[1/10,1/20,1/40,1/80];%空间步长
K=size(tau,2);

```

```

t_cell=cell(K,1);%时间向量
x_cell=cell(K,1);%空间向量
u_cell=cell(K,1);%数值解
ua_cell=cell(K,1);%精确解
epsilon_cell=cell(K,1);%绝对误差
for n=1:K
    [t_cell{n,1},x_cell{n,1},u_cell{n,1}]=fsolve(tau(n),h(n));%求数值解
    ua_cell{n,1}=fa(t_cell{n,1},x_cell{n,1});%精确解
    epsilon_cell{n,1}=abs(ua_cell{n,1}-u_cell{n,1});%求绝对误差
end
figure(1);
%t=1,tau=1/100,h=1/10时数值解和精确解对比
plot(x_cell{1,1},u_cell{1,1}(1/tau(1)+1,:), 'b--o' ...
,x_cell{1,1},ua_cell{1,1}(1/tau(1)+1,:), 'r--x')
h=legend('$u_{\tau,h}(x,1)$','u(x,1)');
set(h,'Interpreter','latex') %设置legend为latex解释器
title('t=1时数值解和精确解');
xlabel('x');ylabel('u(x,1)');

%绘制t=1时的误差曲线
figure(2)
plot(x_cell{1,1},epsilon_cell{1,1}(1/tau(1)+1,:), 'b--o');
hold on;
plot(x_cell{2,1},epsilon_cell{2,1}(1/tau(2)+1,:), 'r--x');
hold on;
plot(x_cell{3,1},epsilon_cell{3,1}(1/tau(3)+1,:), 'g--s');
hold on;
plot(x_cell{4,1},epsilon_cell{4,1}(1/tau(4)+1,:), 'y--x');
hold on;
h=legend('$\tau=1/100,h=1/10$', '$\tau=1/400,h=1/20$', '$\tau=1/1600,h=1/40$', '$\tau=1/6400,h=1/80$');
set(h,'Interpreter','latex') %设置legend为latex解释器
title('t=1时的误差曲线');
xlabel('x');ylabel('|u(x_i,t_n)-u_i^n|');

```

data.m 求特定点的数值解，误差分析。

```

clc;clear;
tao=[1/100,1/400,1/1600,1/6400];%时间步长
h=[1/10,1/20,1/40,1/80];%空间步长
K=size(tao,2);
t_cell=cell(K,1);%时间向量
x_cell=cell(K,1);%空间向量
u_cell=cell(K,1);%数值解
ua_cell=cell(K,1);%精确解
epsilon_cell=cell(K,1);%绝对误差
max_epsilon=ones(K,1);
rate=ones(3,1);%误差比

```

```

for n=1:K
[t_cell{n,1},x_cell{n,1},u_cell{n,1}]=fsolve(tao(n),h(n));%求数值解
ua_cell{n,1}=fa(t_cell{n,1},x_cell{n,1});%精确解
epsilon_cell{n,1}=abs(ua_cell{n,1}-u_cell{n,1});%求绝对误差
max_epsilon(n,1)=max(epsilon_cell{n,1}(:));%求最大误差
end

%E(t,h)/E(4t,2h)
for n=1:K-1
rate(n,1)=max_epsilon(n,1)/max_epsilon(n+1,1);
end

%当x=0.5时,取t=0.1,0.2,...1时,误差的变化,tao=1/100,h=1/10;
u_1=ones(10,1);%特定点数值解
ua_1=ones(10,1);%特定点精确解
[t,x,u]=fsolve(tao(1),h(1));
ua=fa(t,x);
u_1=u(11:10:101,6);
ua_1=ua(11:10:101,6);
epsilon_1=abs(u_1-ua_1);%误差

```

## 附录 C Crank-Nicolson 格式代码

fa.m 求精确解: 与向后 Euler 格式相同

fsolve.m 用 Crank-Nicolson 格式求数值解

```

function [t,x,u] = fsolve(tau,h)
% 用Crank-Nicolson格式解求数值解
% @t 时间向量
% @x 空间向量
% @tau 时间步长
% @h 空间步长
% @u 数值解
N=1/tau;%t被分割的区间数
M=1/h;%x被分割的区间数
t=0:tau:1;
x=0:h:1;
u=ones(N+1,M+1);
u(1,:)=exp(x);%初值
%边值
u(:,1)=exp(t);
u(:,M+1)=exp(1+t);

r=tau/h^2;%步长系数
a1=ones(M-2,1)*(-r/2);%下对角线

```

```

b1=ones(M-1,1)*(1+r);%主队角线
c1=ones(M-2,1)*(-r/2);%上对角线
A1=diag(b1,0)+diag(a1,-1)+diag(c1,1);%系数矩阵

a2=-a1;
b2=ones(M-1,1)*(1-r);
c2=-c1;
A2=diag(b2,0)+diag(a2,-1)+diag(c2,1);%系数矩阵
for k=2:N+1
%右端项
f=A2*u(k-1,2:M)';
f(1)=f(1)+r/2*(u(k,1)+u(k-1,1));
f(M-1)=f(M-1)+r/2*(u(k,M+1)+u(k-1,M+1));
u(k,2:M)=(A1\f)';%由k-1层求第k层的值
end
end

```

fig.m 绘制误差图，精确解和数值解对比图

```

clc;clear;
tau=[1/10,1/20,1/40,1/80];%时间步长
h=[1/10,1/20,1/40,1/80];%空间步长
K=size(tau,2);
t_cell=cell(K,1);%时间向量
x_cell=cell(K,1);%空间向量
u_cell=cell(K,1);%数值解
ua_cell=cell(K,1);%精确解
epsilon_cell=cell(K,1);%绝对误差
for n=1:K
[t_cell{n,1},x_cell{n,1},u_cell{n,1}]=fsolve(tau(n),h(n));%求数值解
ua_cell{n,1}=fa(t_cell{n,1},x_cell{n,1});%精确解
epsilon_cell{n,1}=abs(ua_cell{n,1}-u_cell{n,1});%求绝对误差
end
figure(1);
%t=1,tau=1/100,h=1/10时数值解和精确解对比
plot(x_cell{1,1},u_cell{1,1}(1/tau(1)+1,:), 'b--o')...
,x_cell{1,1},ua_cell{1,1}(1/tau(1)+1,:), 'r--x')
h=legend('$u_{\tau,h}(x,1)$','u(x,1)');
set(h,'Interpreter','latex') %设置legend为latex解释器
title('t=1时数值解和精确解');
xlabel('x');ylabel('u(x,1)');

%绘制t=1时的误差曲线
figure(2)
plot(x_cell{1,1},epsilon_cell{1,1}(1/tau(1)+1,:), 'b--o');
hold on;
plot(x_cell{2,1},epsilon_cell{2,1}(1/tau(2)+1,:), 'r--x');

```

```

hold on;
plot(x_cell{3,1},epsilon_cell{3,1}(1/tau(3)+1,:), 'g--s');
hold on;
plot(x_cell{4,1},epsilon_cell{4,1}(1/tau(4)+1,:), 'y--x');
hold on;
h=legend('$\tau=1/10,h=1/10$', '$\tau=1/20,h=1/20$', '$\tau=1/40,h=1/40$', '$\tau=1/80,h=1/80$');
set(h, 'Interpreter', 'latex') %设置legend为latex解释器
title('t=1时的误差曲线');
xlabel('x'); ylabel('|u(x_i), t_n - u_i^{n}|');

```

data.m 求特定点的数值解，误差分析。

```

clc; clear;
tau=[1/10,1/20,1/40,1/80];%时间步长
h=[1/10,1/20,1/40,1/80];%空间步长
K=size(tau,2);
t_cell=cell(K,1);%时间向量
x_cell=cell(K,1);%空间向量
u_cell=cell(K,1);%数值解
ua_cell=cell(K,1);%精确解
epsilon_cell=cell(K,1);%绝对误差
max_epsilon=ones(K,1);
rate=ones(3,1);%误差比
for n=1:K
    [t_cell{n,1},x_cell{n,1},u_cell{n,1}]=fsolve(tau(n),h(n));%求数值解
    ua_cell{n,1}=fa(t_cell{n,1},x_cell{n,1});%精确解
    epsilon_cell{n,1}=abs(ua_cell{n,1}-u_cell{n,1});%求绝对误差
    max_epsilon(n,1)=max(epsilon_cell{n,1}(:));%求最大误差
end

%E(t,h)/E(2t,2h)
for n=1:K-1
    rate(n,1)=max_epsilon(n,1)/max_epsilon(n+1,1);
end

%当x=0.5时，取t=0.1,0.2,...1时，误差的变化(tau=1/10,h=1/10);
u_1=zeros(10,1);%特定点数值解
ua_1=zeros(10,1);%特定点精确解
[t,x,u]=fsolve(tau(1),h(1));
ua=fa(t,x);
u_1=u(2:1:11,1/h(1)/2+1);
ua_1=ua(2:1:11,1/h(1)/2+1);
epsion_1=abs(u_1-ua_1);%误差

```

## 附录 D 二阶 BDF 格式代码

fa.m 求精确解, 与前面相同

fig.m 绘制误差图, 精确解和数值解对比图, 与 Crank-Nicolson 格式相同

data.m 求特定点的数值解, 误差分析, 与 Crank-Nicolson 格式相同

fsolve.m 用 BDF 格式求数值解

```
function [t,x,u] = fsolve(tau,h)
% 用BDF格式解抛物方程
% @t 时间向量
% @x 空间向量
% @tau 时间步长
% @h 空间步长
% @u 数值解
N=1/tau;%t被分割的区间数
M=1/h;%x被分割的区间数
t=0:tau:1;
x=0:h:1;
u=ones(N+1,M+1);
u(1,:)=exp(x);%初值
%边值
u(:,1)=exp(t);
u(:,M+1)=exp(1+t);

r=tau/h^2;%步长系数
a=ones(M-2,1)*(-r);%下对角线
c=ones(M-2,1)*(-r);%上对角线
b=ones(M-1,1)*(1+2*r);%主对角线
A1=diag(b,0)+diag(a,-1)+diag(c,1);%向后Euler法系数矩阵

a=ones(M-2,1)*(r);%下对角线
c=ones(M-2,1)*(r);%上对角线
b=ones(M-1,1)*(-(2*r+3/2));%主对角线
A2=diag(b,0)+diag(a,-1)+diag(c,1);%向后Euler法系数矩阵

%由第0层求第1层
f=u(1,2:M);
f(1)=f(1)+r*u(2,1);
f(M-1)=f(M-1)+r*u(2,M+1);
u(2,2:M)=(A1\f')';%由0层求第1层的值

%用Crank-Nicolson格式求出的第1层替代向后Euler法求出的第一层数值解
% [t2,x2,u2]=fsolve12(tau,h);
% u(2,2:M)=u2(2,2:M);

%第2层到N层
```

```

for k=3:N+1
%右端项
F=-2*u(k-1,2:M)+1/2*u(k-2,2:M);
F(1)=F(1)-r*u(k,1);
F(M-1)=F(M-1)-r*u(k,M+1);
u(k,2:M)=(A2\F')';%由k-1层求第k层的值
end
end

```