有限差分法

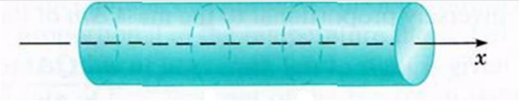
案例 2：均匀直棒热传导问题

舒适 魏华祎 易年余 岳孝强

1. 均匀直棒热传导问题
   - 背景问题与数学建模
   - 有限差分方法
   - 算法设计与实现
   - 数值实验
   - 理论分析

设一根长为 l 的均匀直棒，水平放置，试建立热流穿过直棒的数学模型。

为了叙述的方便，将 x 轴(即横轴) 的正向取为水平方向向右，直棒的左端点为原点，直棒的右端点为l，如下图所示。



下面设温度函数 u 仅与时间变量 t 和空间变量 x 有关，即 $u = u(x, t)$.

下面基于如下两个基本原理来建立数学模型.

1. Fourier 定律：单位时间内通过单位截面的热能 (量) 为

$$k(x)\partial u/\partial x$$

其中 $k(x)$ 称为热传导系数。

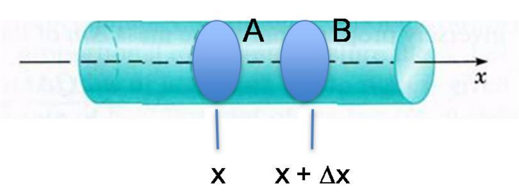
2. 比热容定律：在时间 Δt 内将质量为 m 的物体温度升高 Δu 所吸收的热能为

$$cm\Delta u$$

其中 c 为质量为 m 的物体材料的代表比热容.

采用微元法进行分析，考虑微元

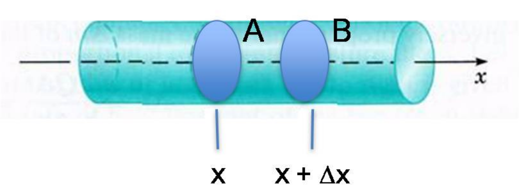
$$[x, x + \Delta x] \times [t, t + \Delta t]$$



不妨设微元中各点处的截面积 $s(x) \equiv 1,\ x \in [x, x + \Delta x]$.

由 Fourier 定律: 在时间 Δt 内，区间 $[x, x+\Delta x]$ 中热能的变化 (增量) 为

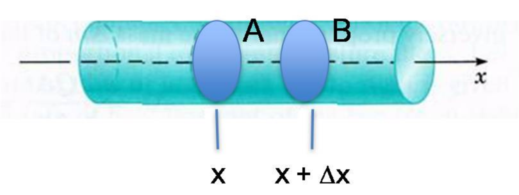
$$\left[k(x+\Delta x)\frac{\partial u}{\partial x}(x+\Delta x, t) - k(x)\frac{\partial u}{\partial x}(x, t)\right]\Delta t \tag{1}$$



另外, 由比热容定律知: 在时间 Δt 内, 区间 $[x, x+\Delta x]$ 中热能变化 (增量)为:

$$c \cdot m \cdot \Delta u = c(x) \cdot (\rho(x)\Delta x) \cdot \Delta u \tag{2}$$

其中 $\rho(x)$ 为直棒的线密度。



利用 (1) 和 (2), 由热能守恒定律, 有

$$c(x)\cdot(\rho(x)\Delta x)\cdot\Delta u=\left[k(x+\Delta x)\frac{\partial u}{\partial x}(x+\Delta x,t)-k(x)\frac{\partial u}{\partial x}(x,t)\right]\Delta t$$

$\Rightarrow$

$$\frac{c(x)\cdot\rho(x)\cdot\Delta u}{\Delta t}=\frac{1}{\Delta x}\left[k(x+\Delta x)\frac{\partial u}{\partial x}(x+\Delta x,t)-k(x)\frac{\partial u}{\partial x}(x,t)\right]$$

取极限 $\Delta t,\Delta x\to 0$, 可得:

$$c(x)\rho(x)\frac{\partial u}{\partial t}(x,t)=\frac{\partial}{\partial x}\left(k(x)\frac{\partial u}{\partial x}(x,t)\right)$$

如果 k, c, ρ 均为常数，则上述方程可写为：

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

其中 $a = \frac{k}{c \cdot \rho}$.

为了保证适定性，还需给出适当的初始时刻的温度分布 $u(x, 0)$ 以及边界条件 $u(0, t)$ 和 $u(l, t)$

一维热传导方程：

$$
\begin{aligned}
&\frac{\partial u}{\partial t}=a\frac{\partial^2 u}{\partial x^2},\ \ 0<x<l,\ \ t>0,\\
&u(0,t)=g_0(t),\ \ t>0,\\
&u(l,t)=g_1(t),\ \ t>0,\\
&u(x,0)=u_0(x),\ \ 0<x<l.
\end{aligned}
\tag{3}
$$

若直棒内部还有热源$f(x)$，则方程为：

$$
\begin{aligned}
&\frac{\partial u}{\partial t}=a\frac{\partial^2 u}{\partial x^2}+f(x),\ \ 0<x<l,\ \ t>0,\\
&u(0,t)=g_0(t),\ \ t>0,\\
&u(l,t)=g_1(t),\ \ t>0,\\
&u(x,0)=u_0(x),\ \ 0<x<l.
\end{aligned}
\tag{4}
$$

下面，针对如下一维热传导方程初边值问题(抛物方程)

$$\begin{cases} \frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} + f(x), & (x,t) \in G, \\ u(x,0) = \phi(x), & 0 < x < l, \\ u(0,t) = u(l,t) = 0, & 0 \le t \le T, \end{cases} \tag{5}$$

在均匀网格剖分下，介绍有限差分法。

回顾有限差分法的步骤：

1. 网格剖分
2. 导数的差分离散
3. 初边值条件处理

网格剖分

取空间步长和时间步长为 $h = \frac{l}{N}, \tau = \frac{T}{M}$
分别对空间变量 x 所属的区间 $[0, l]$ 和时间变量 t 所属的区间 $[0, T]$ 做如下均匀剖分：

$$0 = x_0 < x_1 < \cdots < x_N = l, \quad 0 = t_0 < t_1 < \cdots < t_M = T$$

其中 $x_i = ih,\ t_k = k\tau$.

用两族平行直线 $x = x_j\ (0, 1, \cdots, N)$ 和 $t = t_k\ (k = 0, 1, \cdots, M)$ 将矩形域 $\bar{G}$ 分割成矩形网格，网格节点集合为

$$\bar{G}_h = G_h \cup \Gamma_h = \{(x_j, t_k) : 0 \le j \le N; 0 \le k \le M\}$$

其中

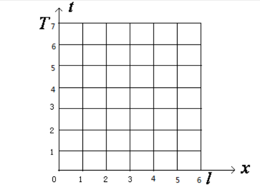
$$G_h = \{(x_j, t_k):\ 0 < j < N;\ 0 < k \le M\}$$

为网格内节点集合.

$$\Gamma_h = \{(x_j, t_k):\ j = 0, N; k = 1, \cdots, M\} \bigcup \{(x_j, t_0):\ j = 0, \cdots, N\}$$

为网格边界节点集合.

下图为当 $N = 6$ 和 $M = 7$ 时 (它们分别是沿 x 和 t 方向的剖分段数) 的矩形网格剖分图.



导数的差分离散与初边值条件处理

用 u_j^k 表示差分解在网格节点 (x_j, t_k) 处的分量, 下面采用逐层计算 的思想建立差分方程 (格式).

设从第 0 个时间层到第 $k \geq 0$ 个时间层的差分解分量

$$u_j^i,\ i = 0, \cdots, k;\ j = 0, \cdots, N$$

已经求得.

导数的差分离散与初边值条件处理

下面建立第 $k+1$ 个时间层(简称当前层)上的差分方程 (格式)

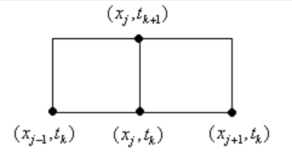
$m \geq 2$ 层差分格式: 若格式中仅涉及当前层以及第 $k+1-l$ 个时间层的差分解分量

$$u_j^l,\ j = 0, \cdots, N;\ \ l = 1, \cdots, m-1$$

下面首先介绍三种常用的二层格式.

(一) 向前差分格式

考虑当前层上的任意内节点 (x_j, t_{k+1}), 规定其差分格式涉及的模板点为



对 (x_j, t_k) 处的微分方程

$$\frac{\partial u}{\partial t}\Big|_{(x_j,t_k)} = a\frac{\partial^2 u}{\partial x^2}\Big|_{(x_j,t_k)} + f(x_j) \tag{6}$$

通常有两种近似方法.

方法一. 差商逼近导数 (直接差分方法)

对 (6) 中的导数用差商近似:

$$\frac{\partial u(x_j, t_k)}{\partial t} \approx \frac{u(x_j, t_{k+1}) - u(x_j, t_k)}{\tau} \quad \text{(向前差商)} \tag{7}$$

$$\frac{\partial^2 u(x_j, t_k)}{\partial x^2} \approx \frac{u(x_{j+1}, t_k) - 2u(x_j, t_k) + u(x_{j-1}, t_k)}{h^2} \quad \text{(二阶中心差商)} \tag{8}$$

方法二. 待定系数 + Taylor 展开

$$\begin{aligned}\frac{\partial u}{\partial t}\Big|_{(x_j,t_k)} - a\frac{\partial^2 u(x_j,t_k)}{\partial x^2} \quad \approx \quad & \alpha_{j,k+1}u(x_j,t_{k+1}) + \alpha_{j,k}u(x_j,t_k) \\ & +\alpha_{j-1,k}u(x_{j-1},t_k) + \alpha_{j+1,k}u(x_{j+1},t_k)\end{aligned}$$

要求截断误差的阶尽可能高.

利用二元函数的 Taylor 展开

$$\cdots\cdots$$

可得近似公式 (7) + (8).

记 $f_j = f(x_j)$，将 (7) 和 (8) 代入 (6), 则有关于 (x_j, t_{k+1}) 的向前差分格式,

$$\frac{u_j^{k+1} - u_j^k}{\tau} = a\frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h^2} + f_j \tag{9}$$

在 (9) 两边同乘 τ, 整理可得

$$u_j^{k+1} = ru_{j-1}^k + (1-2r)u_j^k + ru_{j+1}^k + \tau f_j \tag{10}$$

其中, $r = a\frac{\tau}{h^2}$ 称为网格比 (简称网比).

利用(10), 并对初边值条件进行处理, 有

$$\begin{cases} u_j^{k+1} = ru_{j-1}^k + (1-2r)u_j^k + ru_{j+1}^k + \tau f_j, \ j = 1, \cdots, N-1; \ k = 0, \cdots, M-1 \\ u_j^0 = u(x_j, 0) = \phi(x_j), \ j = 1, \cdots, N-1 \\ u_0^k = u(0, t_k) = 0, \ u_N^k = u(l, t_k) = 0, \ k = 0, 1, \cdots, M \end{cases} \tag{11}$$

令 $N-1$ 维 (第 k 个网格层) 差分解向量和右端向量

$$U^k=(u_1^k,u_2^k,\cdots,u_{N-1}^k)^T,\quad F=(f_1+ru_0^k,f_2,\cdots,f_{N-1}+ru_N^k)^T$$

则向前差分格式 (10) 的矩阵表示为：

$$U^{k+1}=A_0U^k+\tau F \tag{12}$$

其中

$$A_0=\begin{bmatrix} 1-2r & r & & & \\ r & 1-2r & r & & \\ & \ddots & \ddots & \ddots & \\ & & r & 1-2r & r \\ & & & r & 1-2r \end{bmatrix}_{(N-1)\times(N-1)}$$

由此可见:

1. 利用 (11) 可逐层求出差分解分量

   在 (11) 中, 取 $k=0$, 利用初值 $u_j^0=\phi(x_j)$ $(j=1,\cdots,N-1)$ 和边值 $u_0^0=u_N^0=0$, 可算出第一层的 u_j^1 $(j=1,\cdots,N-1)$;

   在 (11) 中, 取 $k=1$, 利用 u_j^1 $(j=1,\cdots,N-1)$ 和边值 $u_0^1=u_N^1=0$, 可算出第二层的 u_j^2 $(j=1,\cdots,N-1)$;

   $$\cdots\cdots$$

2. 向前差分格式 (11) 是一种显格式, 即为了求出差分解, 无需求解线性代数方程组.

引入向前差分算子 $L_h^{(1)}$，它满足

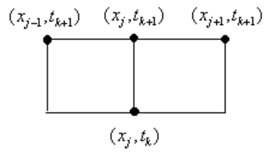
$$L_h^{(1)} u_j^k = \frac{u_j^{k+1} - u_j^k}{\tau} - a\frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h^2}.$$

可以证明向前差分格式 (9) 的截断误差为

$$R_j^k(u) = O(\tau + h^2).$$

(二) 向后差分格式

模板点为



对 (x_j, t_{k+1}) 处的微分方程

$$\frac{\partial u}{\partial t}\Big|_{(x_j,t_{k+1})} = a\frac{\partial^2 u}{\partial x^2}\Big|_{(x_j,t_{k+1})} + f(x_j) \tag{13}$$

采用直接差分方法进行近似.

对 (13) 中的偏导数用差商近似：

$$\frac{\partial u(x_j, t_{k+1})}{\partial t} \approx \frac{u(x_j, t_{k+1}) - u(x_j, t_k)}{\tau} \quad \text{(向后差商)} \tag{14}$$

$$\frac{\partial^2 u(x_j, t_{k+1})}{\partial x^2} \approx \frac{u(x_{j+1}, t_{k+1}) - 2u(x_j, t_{k+1}) + u(x_{j-1}, t_{k+1})}{h^2} \text{(二阶中心差商)} \tag{15}$$

将 (14) 和 (15) 代入 (13), 则有关于 (x_j, t_{k+1}) 的向后差分格式,

$$\frac{u_j^{k+1} - u_j^k}{\tau} = a\frac{u_{j+1}^{k+1} - 2u_j^{k+1} + u_{j-1}^{k+1}}{h^2} + f_j \tag{16}$$

在 (16) 两边同乘 τ, 整理可得

$$-ru_{j-1}^{k+1} + (1+2r)u_j^{k+1} - ru_{j+1}^{k+1} = u_j^k + \tau f_j \tag{17}$$

利用(17), 并对初边值条件进行处理, 有

$$\begin{cases} -ru_{j-1}^{k+1} + (1+2r)u_j^{k+1} - ru_{j+1}^{k+1} = u_j^k + \tau f_j, j = 1, \cdots, N-1; k = 0, \cdots, M-1 \\ u_j^0 = u(x_j, 0) = \phi(x_j),\ j = 1, \cdots, N-1 \\ u_0^k = u(0, t_k) = 0,\ u_N^k = u(l, t_k) = 0,\ k = 0, 1, \cdots, M \end{cases} \tag{18}$$

向后差分格式 (17) 的矩阵表示为

$$A_1 U^{k+1} = U^k + \tau F \tag{19}$$

其中

$$A_1 = \begin{bmatrix} 1+2r & -r & & & \\ -r & 1+2r & -r & & \\ & \ddots & \ddots & \ddots & \\ & & -r & 1+2r & -r \\ & & & -r & 1+2r \end{bmatrix}_{(N-1)\times(N-1)}$$

由此可见：

1. 利用 (18) 可逐层求出差分解分量
2. 向后差分格式 (18) 是一种隐格式，即为了求解第 $k+1$ 层上的差分解分量，需求解线性代数方程组 (19). 注意 (19) 的系数矩阵 A_1 为三对角阵 (对角占优)，因此可以用"追赶法"进行求解，其运算复杂度为 $O(N)$.

引入向后差分算子 $L_h^{(2)}$, 它满足

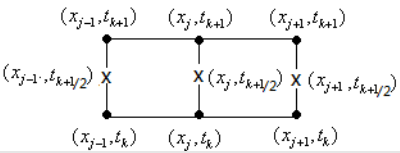
$$L_h^{(2)} u_j^k = \frac{u_j^{k+1} - u_j^k}{\tau} - a\frac{u_{j+1}^{k+1} - 2u_j^{k+1} + u_{j-1}^{k+1}}{h^2}$$

可以证明向后差分格式 (17) 的截断误差为

$$R_j^k(u) = O(\tau + h^2).$$

(三) 六点对称格式 (Crank-Nicholson格式)

对当前层上的任意内节点 (x_j, t_{k+1}), 其模板点为



对 $(x_j, t_{k+1/2})$ 处的微分方程

$$\frac{\partial u}{\partial t}\Big|_{(x_j, t_{k+1/2})} = a\frac{\partial^2 u}{\partial x^2}\Big|_{(x_j, t_{k+1/2})} + f(x_j) \tag{20}$$

采用直接差分方法进行近似.

对 (20) 中的偏导数用差商近似：

$$\frac{\partial u(x_j, t_{k+1/2})}{\partial t} \approx \frac{u(x_j, t_{k+1}) - u(x_j, t_k)}{\tau} \quad \text{(一阶中心差商)} \tag{21}$$

$$\frac{\partial^2 u(x_j, t_{k+1/2})}{\partial x^2} \approx \frac{u(x_{j+1}, t_{k+1/2}) - 2u(x_j, t_{k+1/2}) + u(x_{j-1}, t_{k+1/2})}{h^2}$$

由此并利用

$$u(x_{j+m}, t_{k+1/2}) \approx \frac{u(x_{j+m}, t_k) + u(x_{j+m}, t_{k+1})}{2}, \quad m = 0, \pm 1$$

可得

$$\frac{\partial^2 u(x_j, t_{k+1/2})}{\partial x^2} \approx \frac{1}{2}[\frac{u_{j+1}^{k+1} - 2u_j^{k+1} + u_{j-1}^{k+1}}{h^2} + \frac{u_{j+1}^{k} - 2u_j^{k} + u_{j-1}^{k}}{h^2}] \tag{22}$$

将 (21) 和 (22) 代入 (20), 则可得关于 (x_j, t_{k+1}) 的六点对称格式,

$$\frac{u_j^{k+1}-u_j^k}{\tau}=\frac{a}{2}[\frac{u_{j+1}^{k+1}-2u_j^{k+1}+u_{j-1}^{k+1}}{h^2}+\frac{u_{j+1}^k-2u_j^k+u_{j-1}^k}{h^2}]+f_j \tag{23}$$

在 (23) 两边同乘 τ, 整理可得

$$-\frac{r}{2}u_{j-1}^{k+1}+(1+r)u_j^{k+1}-\frac{r}{2}u_{j+1}^{k+1}=\frac{r}{2}u_{j-1}^k+(1-r)u_j^k+\frac{r}{2}u_{j+1}^k+\tau f_j \tag{24}$$

利用(24), 并对初边值条件进行处理, 有

$$\begin{cases} -\frac{r}{2}u_{j-1}^{k+1}+(1+r)u_j^{k+1}-\frac{r}{2}u_{j+1}^{k+1}=\frac{r}{2}u_{j-1}^{k}+(1-r)u_j^{k}+\frac{r}{2}u_{j+1}^{k}+\tau f_j \\ \qquad\qquad\qquad\qquad j=1,\cdots,N-1;\, k=0,\cdots,M-1 \\ u_j^0=u(x_j,0)=\phi(x_j),\ j=1,\cdots,N-1 \\ u_0^k=u(0,t_k)=0,\ u_N^k=u(l,t_k)=0,\ k=0,1,\cdots,M \end{cases} \tag{25}$$

注: 六点对称格式也可通过将向前、向后差分格式 (9) 和 (16) 做算术平均得到.

六点对称格式 (24) 的矩阵表示为

$$A_1 U^{k+1} = A_0 U^k + \tau F \tag{26}$$

其中

$$A_0 = \begin{bmatrix} 1-r & \frac{r}{2} & & & \\ \frac{r}{2} & 1-r & \frac{r}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{r}{2} & 1-r & \frac{r}{2} \\ & & & \frac{r}{2} & 1-r \end{bmatrix}_{(N-1)\times(N-1)}$$

$$A_1 = \begin{bmatrix} 1+r & -\frac{r}{2} & & & \\ -\frac{r}{2} & 1+r & -\frac{r}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{r}{2} & 1+r & -\frac{r}{2} \\ & & & -\frac{r}{2} & 1+r \end{bmatrix}_{(N-1)\times(N-1)}$$

由此可知:

1. 利用 u_j^0 和边值便可逐层求得 u_j^k.
2. 六点对称格式 (25) 也是一种隐格式.

引入六点对称差分算子 $L_h^{(3)}$, 它满足

$$L_h^{(3)} u_j^k = \frac{u_j^{k+1} - u_j^k}{\tau} - \frac{a}{2}[\frac{u_{j+1}^{k+1} - 2u_j^{k+1} + u_{j-1}^{k+1}}{h^2} + \frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h^2}]$$

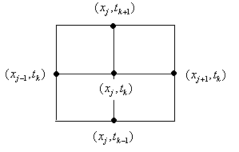
可以证明六点对称格式 (23) 的截断误差为

$$R_j^k(u) = O(\tau^2 + h^2).$$

下面介绍一种三层格式(多步法).

(四) Richardson格式

模板点为



对 (x_j, t_k) 处的微分方程

$$\frac{\partial u}{\partial t}\Big|_{(x_j,t_k)} = a\frac{\partial^2 u}{\partial x^2}\Big|_{(x_j,t_k)} + f(x_j) \tag{27}$$

采用直接差分方法进行近似.

对 (27) 中的偏导数用差商近似：

$$\frac{\partial u(x_j, t_k)}{\partial t} \approx \frac{u(x_j, t_{k+1}) - u(x_j, t_{k-1})}{2\tau} \quad \text{(一阶中心差商)} \tag{28}$$

$$\frac{\partial^2 u(x_j, t_k)}{\partial x^2} \approx \frac{u(x_{j+1}, t_k) - 2u(x_j, t_k) + u(x_{j-1}, t_k)}{h^2} \text{(二阶中心差商)} \tag{29}$$

将 (28) 和 (29) 代入 (27), 则有关于 (x_j, t_{k+1}) 的 Richardson 差分格式,

$$\frac{u_j^{k+1} - u_j^{k-1}}{2\tau} = a\frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h^2} + f_j \tag{30}$$

在 (30) 两边同乘 τ, 整理可得

$$u_j^{k+1} = 2r(u_{j+1}^k - 2u_j^k + u_{j-1}^k) + u_j^{k-1} + 2\tau f_j \tag{31}$$

利用(31), 并对初边值条件进行处理, 有

$$\begin{cases} u_j^{k+1} = 2r(u_{j+1}^k - 2u_j^k + u_{j-1}^k) + u_j^{k-1} + 2\tau f_j, \\ \qquad\qquad\qquad\qquad j = 1, \cdots, N-1; k = 0, \cdots, M-1 \\ u_j^0 = u(x_j, 0) = \phi(x_j), \ j = 1, \cdots, N-1 \\ u_0^k = u(0, t_k) = 0, \ u_N^k = u(l, t_k) = 0, \ k = 0, 1, \cdots, M \end{cases} \tag{32}$$

Richardson 差分格式 (31) 的矩阵表示为

$$U^{k+1} = A_1 U^k + U^{k-1} + 2\tau F \tag{33}$$

其中

$$A_1 = \begin{bmatrix} -4r & 2r & & & \\ 2r & -4r & 2r & & \\ & \ddots & \ddots & \ddots & \\ & & 2r & -4r & 2r \\ & & & 2r & -4r \end{bmatrix}_{(N-1)\times(N-1)}$$

由此可见:

1. 为了使计算能够逐层进行, 除初值 u_j^0 外, 还要用到 u_j^1, 这可以用前面介绍的两层格式计算.
2. Richardson 差分格式 (32) 是一种显格式.

引入 Richardson 差分算子 $L_h^{(4)}$, 它满足

$$L_h^{(4)} u_j^k = \frac{u_j^{k+1} - u_j^{k-1}}{2\tau} - a\frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h^2}$$

可以证明 Richardson 差分格式 (30) 的截断误差为

$$R_j^k(u) = O(\tau^2 + h^2).$$

习题 1 导出向前（向后）差分格式 (16), 六点对称格式 (23) 及 Richardson 差分格式 (30) 的截断误差.

习题 2 将向前差分格式和向后差分格式作加权平均, 得到如下格式:

$$\frac{u_j^{k+1}-u_j^k}{\tau}=\frac{a}{h^2}\left[\theta\left(u_{j+1}^{k+1}-2u_j^{k+1}+u_{j-1}^{k+1}\right)+(1-\theta)\left(u_{j+1}^k-2u_j^k+u_{j-1}^k\right)\right] \tag{34}$$

其中 $0\leqslant\theta\leqslant 1$. 试计算截断误差, 并证明当 $\theta=\frac{1}{2}-\frac{1}{12r}$ 时, 截断误差的阶最高($O(\tau^2+h^4)$).

注: 除了以上四种差分格式外, 还可以作出许多逼近 (5) 的差分格式, 但并不是每一个差分格式都是可用的. 衡量一个差分格式是否经济适用, 主要由以下几个方面的因素决定:

- 计算简单.

  显格式 (计算最简单): 向前差分格式, Richardson 差分格式;

  隐格式 (若系数矩阵为三对角矩阵, 计算也简单): 向后差分格式, Richardson 差分格式.

- 收敛性和收敛速度.

  截断误差阶为 $O(\tau^2 + h^2)$: 六点对称格式, Richardson 格式;

  截断误差阶为 $O(\tau + h^2)$: 向前差分格式, 向后差分格式.

- 稳定性.

首先考察 Richardson 差分格式是否按初值稳定.

取关于空间变量 x 所属区域 $[0, l]$ 上的剖分段数 $N = 2M$, 记差分解序列 (或差分解网函数) $\{u_j^k\}$, $\{v_j^k\}$ 分别满足

$$\begin{cases} u_j^{k+1} = 2r(u_{j+1}^k - 2u_j^k + u_{j-1}^k) + u_j^{k-1} + 2\tau f_j \\ \qquad\qquad\qquad j = 1, \cdots, N-1; k = 0, \cdots, M-1 \\ u_j^0 = u(x_j, 0) = 0, \ j = 1, \cdots, N-1 \\ u_0^k = u(0, t_k) = 0, \ u_N^k = u(l, t_k) = 0, \ k = 0, 1, \cdots, M \end{cases}$$

和

$$\begin{cases} v_j^{k+1} = 2r(v_{j+1}^k - 2v_j^k + v_{j-1}^k) + v_j^{k-1} + 2\tau f_j \\ \qquad\qquad\qquad j = 1, \cdots, N-1; k = 0, \cdots, M-1 \\ v_j^0 = v(x_j, 0) = \delta_{jM}\varepsilon, , \ j = 1, \cdots, N-1 \\ v_0^k = v(0, t_k) = 0, \ v_N^k = v(l, t_k) = 0, \ k = 0, 1, \cdots, M \end{cases}$$

令内节点 (x_j, t_k) 处的误差分量 $e_j^k = v_j^k - u_j^k$, 则误差序列 $\{e_j^k\}$ 满足如下差分方程:

$$\begin{cases} e_j^{k+1} = 2r(e_{j+1}^k - 2e_j^k + e_{j-1}^k) + e_j^{k-1} \\ \qquad\qquad\qquad j = 1, \cdots, N-1; k = 0, \cdots, M-1 \\ e_j^0 = \delta_{jM}\varepsilon, , \; j = 1, \cdots, N-1 \\ e_0^k = 0, \; e_N^k = 0, \; k = 0, 1, \cdots, M \end{cases}$$

假设整个计算的过程均是精确的, 且设$e_j^{-1} = 0$, 则当 $r = 1/2$ 时通过计算可知初始误差的传递情况如下表所示:

k \ j	$M-3$	$M-2$	$M-1$	M	$M+1$	$M+2$	$M+3$
0	0	0	0	ε	0	0	0
1	0	0	ε	-2ε	ε	0	0
2	0	ε	-4ε	7ε	-4ε	ε	0
3	ε	-6ε	17ε	-24ε	17ε	-6ε	ε
4	-8ε	31ε	-68ε	89ε	-68ε	31ε	-8ε
5	49ε	-144ε	273ε	388ε	273ε	-144ε	49ε
6	-260ε	641ε	-1096ε	1311ε	-1096ε	641ε	-260ε

从上表可知, 误差随着 $k \to \infty$ 无限增长, 所以该差分格式是不稳定的. 实际上对于任何 $r > 0$ 都有类似的现象, 所以该格式是绝对不稳定的.

由此可知, 虽然 Richardson 格式是显格式, 且其截断误差的阶为 $O(\tau^2 + h^2)$, 但从稳定性方面来看, 它是不可用的.

接下来考察向前差分格式是否稳定

同样取关于空间变量 x 所属区域 $[0, l]$ 上的剖分段数 $N = 2M$, 记差分解序列 (或差分解网函数) $\{u_j^k\}$, $\{v_j^k\}$ 分别满足

$$\begin{cases} u_j^{k+1} = ru_{j-1}^k + (1-2r)u_j^k + ru_{j+1}^k + \tau f_j, \\ \qquad\qquad\qquad j = 1, \cdots, N-1; k = 0, \cdots, M-1 \\ u_j^0 = u(x_j, 0) = 0, \ j = 1, \cdots, N-1 \\ u_0^k = u(0, t_k) = 0, \ u_N^k = u(l, t_k) = 0, \ k = 0, 1, \cdots, M \end{cases}$$

和

$$\begin{cases} v_j^{k+1} = rv_{j-1}^k + (1-2r)v_j^k + rv_{j+1}^k + \tau f_j, \\ \qquad\qquad\qquad j = 1, \cdots, N-1; k = 0, \cdots, M-1 \\ v_j^0 = v(x_j, 0) = \delta_{jM}\varepsilon, , \ j = 1, \cdots, N-1 \\ v_0^k = v(0, t_k) = 0, \ v_N^k = v(l, t_k) = 0, \ k = 0, 1, \cdots, M \end{cases}$$

那么此时误差序列 $\{e_j^k\}$ 满足如下差分方程:

$$\begin{cases} e_j^{k+1} = re_{j-1}^k + (1-2r)e_j^k + re_{j+1}^k, \ j = 1, \cdots, N-1; k = 0, \cdots, M-1 \\ e_j^0 = \delta_{jM}\varepsilon, , \ j = 1, \cdots, N-1 \\ e_0^k = 0, \ e_N^k = 0, \ k = 0, 1, \cdots, M \end{cases}$$

同样取 $r = 1/2$, 初始误差的传递情况如下表所示:

k \ j	$M-3$	$M-2$	$M-1$	M	$M+1$	$M+2$	$M+3$
0	0	0	0	ε	0	0	0
1	0	0	0.5ε	0	0.5ε	0	0
2	0	0.25ε	0	0.5ε	0	0.25ε	0
3	0.125ε	0	0.375ε	0	0.375ε	0	0.125ε
4	0	0.25ε	0	0.375ε	0	0.25ε	0

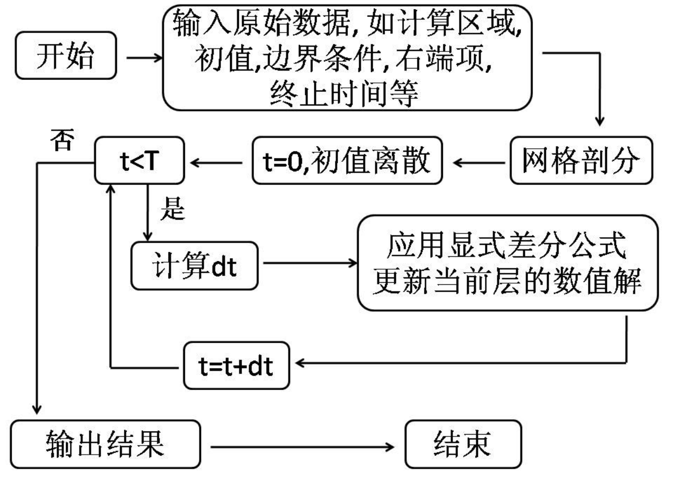
由上表可知, 误差逐渐衰减. 因此当 $r = 1/2$ 时向前差分格式是可取的.

考虑如下热传导问题差分离散的算法实现，

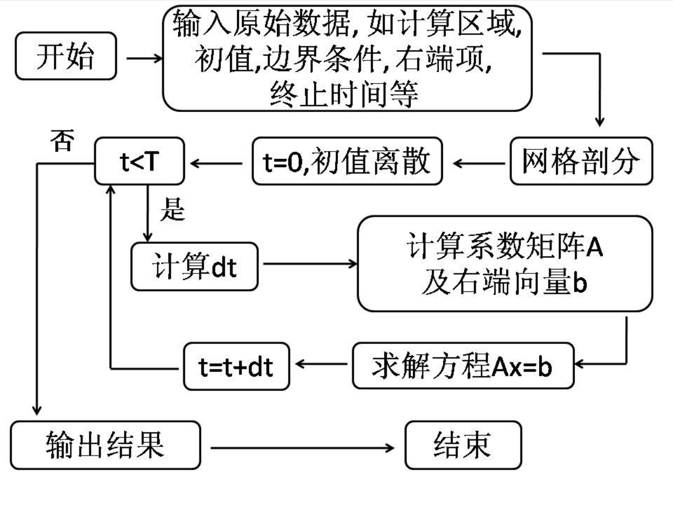
$$
\begin{aligned}
&\frac{\partial u}{\partial t} - a\frac{\partial^2 u}{\partial x^2} = f(x,t),\\
&u(L,t) = u_L(t), \quad u(R,t) = u_R(t),\\
&u(x,0) = u_0(x).
\end{aligned}
$$

首先给出算法的流程，时间离散方面分别考虑显式、隐式和Crank-Nicolson方法，然后给出实现算法的Matlab代码。

显式算法流程


开始
输入原始数据,如计算区域,初值,边界条件,右端项,终止时间等
网格剖分
t=0,初值离散
t<T
否
是
计算dt
应用显式差分公式更新当前层的数值解
t=t+dt
输出结果
结束

隐式算法流程


开始
输入原始数据，如计算区域，初值，边界条件，右端项，终止时间等
网格剖分
t=0,初值离散
t<T
否
是
计算dt
计算系数矩阵A及右端向量b
求解方程Ax=b
t=t+dt
输出结果
结束

主测试 matlab 脚本程序

```
%% 一维热传导方程有限差分方法主测试脚本 main_test.m
%   依次测试:
%          向前差分
%          向后差分
%          六点对称格式
%   并可视化数值计算结果。
%
% 作者: 魏华祎 <weihuayi@xtu.edu.cn>

pde = model_data(); %模型数据结构体

% 向前差分格式
[X,T,U] = heat_equation_fd1d(100,10000,pde,'forward');
showvarysolution(X,T,U);% 以随时间变化方式显示数值解
showsolution(X,T,U); % 以二元函数方式显示数值解

% 向后差分格式
[X,T,U] = heat_equation_fd1d(100,100,pde,'backward');
showvarysolution(X,T,U);% 以随时间变化方式显示数值解
showsolution(X,T,U); % 以二元函数方式显示数值解
```

主测试 matlab 脚本程序 II

```
% 六点对称格式, 即 Crank-Nicholson 格式
[X,T,U] = heat_equation_fd1d(100,100,pde,'crank-nicholson');
showvarysolution(X,T,U);% 以随时间变化方式显示数值解
showsolution(X,T,U); % 以二元函数方式显示数值解
```

有限差分方法实现

```
function [X,T,U] = heat_equation_fd1d(NS,NT,pde,method)
%% HEAT_EQUATION_FD1D 利用有限差分方法计算一维热传导方程
%
%   输入参数:
%       NS 整型, 空间剖分段数
%       NT 整型, 时间剖分段数
%       pde 结构体, 待求解的微分方程模型的已知数据,
%                      如边界、初始、系数和右端项等条件
%       method 字符串, 代表求解所用离散格式
%           F 或 f 或 forward : 向前差分格式
%           B 或 b 或 backward : 向后差分格式
%           CN 或 cn 或 crank-nicholson 或 Crank-Nicholson :
%                      -- 六点对称格式( Crank-Nicholson 格式)
%   输出参数:
%       X 长度为 NS+1 的列向量, 空间网格剖分
%       T 长度为 NT+1 的行向量, 时间网格剖分
%       U (NS+1)*(NT+1) 矩阵, U(:,i) 表示第 i 个时间层网格部分上的数值解
%
%   作者: 魏华祎 <weihuayi@xtu.edu.cn>
```

有限差分方法实现 II

```
[X,h] = pde.space_grid(NS);
[T,tau] = pde.time_grid(NT);
N = length(X);M = length(T);
r = pde.a()*tau/h/h;
if r >= 0.5 && ismember(method,{'F','f','forward'})
    error('时间空间离散不满足向前差分的稳定条件! ')
end
U = zeros(N,M);
U(:,1) = pde.u_initial(X);
U(1,:) = pde.u_left(T);
U(end,:) = pde.u_right(T);
switch(method)
    case {'F','f','forward'}
        forward();
    case {'B','b','backward'}
        backward();
    case {'CN','cn','crank-nicholson','Crank-Nicholson'}
        crank_nicholson();
    otherwise
        disp(['Sorry,␣I␣do␣not␣know␣your␣', method]);
end
```

有限差分方法实现 III

```
%% 向前差分方法
    function forward()
        d = 1 - 2*ones(N-2,1)*r;
        c = ones(N-3,1)*r;
        A = diag(c,-1) + diag(c,1)+diag(d);
        for i = 2:M
            RHS = tau*pde.f(X,T(i));
            RHS(2) = RHS(2) + r*U(1,i-1);
            RHS(end-1) = RHS(end-1) + r*U(end,i-1);
            U(2:end-1,i)=A*U(2:end-1,i-1)+ RHS(2:end-1);
        end
    end
%% 向后差分方法
    function backward()
        d = 1 + 2*ones(N-2,1)*r;
        c = -ones(N-3,1)*r;
        A = diag(c,-1) + diag(c,1)+diag(d);
        for i = 2:M
            RHS = tau*pde.f(X,T(i));
            RHS(2) = RHS(2) + r*U(1,i);
            RHS(end-1) = RHS(end-1) + r*U(end,i);
```

有限差分方法实现 IV

```
            U(2:end-1,i)=A\(U(2:end-1,i-1)+ RHS(2:end-1));
        end
    end
%% 六点对称格式, 即 Crank_Nicholson 格式
    function crank_nicholson()
        d1 = 1 + ones(N-2,1)*r;
        d2 = 1 - ones(N-2,1)*r;
        c = 0.5*ones(N-3,1)*r;
        A1 = diag(-c,-1) + diag(-c,1)+diag(d1);
        A0 = diag(c,-1) + diag(c,1) + diag(d2);
        for i = 2:M
            RHS = tau*pde.f(X,T(i));
            RHS(2) = RHS(2) + 0.5*r*(U(1,i)+U(1,i-1));
            RHS(end-1) = RHS(end-1) + ...
                0.5*r*(U(end,i)+U(end,i-1));
            U(2:end-1,i)=A1\(A0*U(2:end-1,i-1)+ RHS(2:end-1));
        end
    end
end
```

可视化

```
function showvarysolution(X,T,U)
%% SHOWVARYSOLUTION 显示数值解随着时间的变化
%
% 输入参数:
%        X 长度为的列向量，空间网格剖分N
%        T 第度为的行向量，时间网格剖分M
%        U N*M 矩阵，U(:,i) 表示第  i 个时间层网格部分上的数值解
%
%  作者: 魏华祎 <weihuayi@xtu.edu.cn>

M = size(U,2);
figure
xlabel('X');
ylabel('U');
s = [X(1),X(end),min(min(U)),max(max(U))];
axis(s);
for i = 1:M
   plot(X,U(:,i));
   axis(s);
   pause(0.0001);
```

可视化 II

```
    title(['T=',num2str(T(i)),'␣时刻的温度分布'])
end


function showsolution(X,T,U)
%%  SHOWSOLUTION  以二元函数方式显示数值解
%
%  输入参数:
%         X 长度为的列向量, 空间网格剖分N
%         T 第度为的行向量, 时间网格剖分M
%         U N*M 矩阵, U(:,i) 表示第  i 个时间层网格部分上的数值解
%
%   作者: 魏华祎 <weihuayi@xtu.edu.cn>

[x,t] = meshgrid(X,T);
mesh(x,t,U');
xlabel('X');
ylabel('T');
zlabel('U(X,T)');

end
```

算例 1：我们利用有限差分法去求解

$$
\begin{aligned}
&\frac{\partial u}{\partial t} - k\frac{\partial^2 u}{\partial x^2} = 0,\\
&u(0,t) = 0, \quad u(1,t) = 0, \quad 0 \le t \le 0.1\\
&u(x,0) = e^{-\frac{(x-0.25)^2}{0.01}} + 0.1\sin(20\pi x), \quad 0 < x < 1.
\end{aligned}
$$

其中系数$k = 1$.

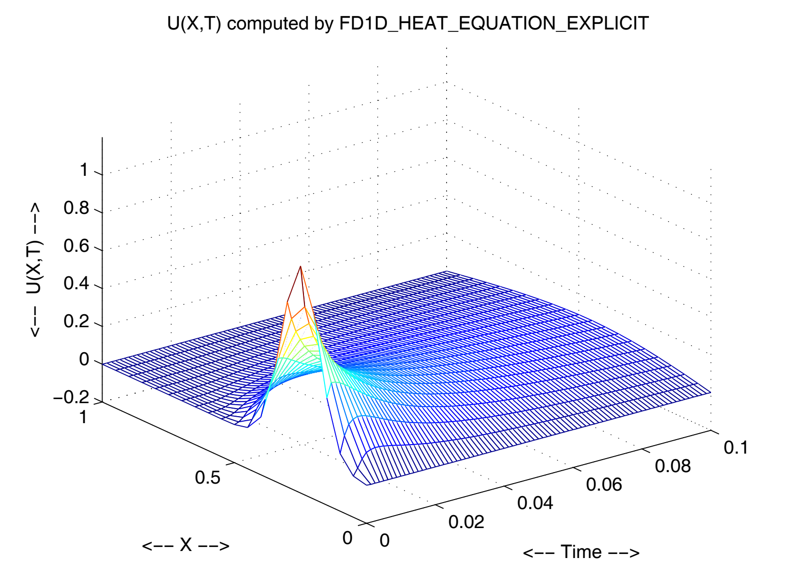
模型数据的 Matlab 实现如下：

```
function pde = model_data()
% MODEL_DATA 模型数据

pde = struct('u_initial',@u_initial,'u_left',@u_left,...
    'u_right',@u_right,'f',@f,'time_grid',@time_grid,...
    'space_grid',@space_grid,'a',@a);

    function [T,tau] = time_grid(NT)
        T = linspace(0,0.1,NT+1);
        tau = 0.1/NT;
```

```
    end
    function [X,h] = space_grid(NS)
        X = linspace(0,1,NS+1)';
        h = 1/NS;
    end
    function u = u_initial(x)
        u = exp(-(x-0.025).^2/0.01)+0.1*sin(20*pi*x);
    end
    function u = u_left(t)
        u = zeros(size(t));
    end
    function u = u_right(t)
        u = zeros(size(t));
    end
    function f = f(x,t)
        f = zeros(size(x));
    end
    function a = a()
        a = 1;
    end
end
```


U(X,T) computed by FD1D_HEAT_EQUATION_EXPLICIT
<-- U(X,T) -->
1
0.8
0.6
0.4
0.2
0
−0.2
1
0.5
0
<-- X -->
0
0.02
0.04
0.06
0.08
0.1
<-- Time -->

上机实验题目

用向前差分求解如下一维热传导方程，并观察最大模误差变化情况：

$$
\begin{aligned}
&\frac{\partial u}{\partial t} - a\frac{\partial^2 u}{\partial x^2} = f(x,t),\\
&u(0,t) = u_L(t), \quad u(1,t) = u_R(t),\\
&u(x,0) = u_0(x),\\
&x \in [0,1], t \in [0,0.1].
\end{aligned}
$$

其中 $a = 1$; 真解为 $u(x,t) = \sin(2\pi x)e^{10t}$

最大模误差定义如下：

$$
E = \max_{x_i, t_j} \left| u(x_i, t_j) - U(i,j) \right|
$$

上机实验题目 II

即所有网格点处数值解和真解误差绝对值的最大值，最大模误差 E 与时间步长 τ 和空间步长 h 满足如下关系：

$$E = o(\tau + h^2)$$

所以，当 τ 变为 $\frac{\tau}{4}$, h 变为 $\frac{h}{2}$ 时， E 应变为 $\frac{E}{4}$.