偏微分方程数值解法

椭圆型方程求解

刘 瑜

March 24, 2017

Outline

椭圆型方程描述的是平衡问题，属于纯粹的边值问题.
在很多情形中，椭圆型方程是非定常问题的极限情形.
比如描述的问题增加时间维度，即增加时间导数项 ∂t 或 $\partial^2/\partial t^2$，即可转化为时间演化问题.
椭圆型方程的求解方法在抛物型和双曲型方程的求解中亦能用到.

Outline

考虑两点边值问题

$$\frac{d}{dx}K(x)\frac{dU}{dx} = r(x), \quad 0 < x < L$$
$$U(0) = U_0, U(L) = U_L$$

其中 $K(x), r(x)$ 是可微的，因而上述方程等价为

$$K\frac{d^2U}{dx^2} + \frac{dK}{dx}\frac{dU}{dx} = r$$

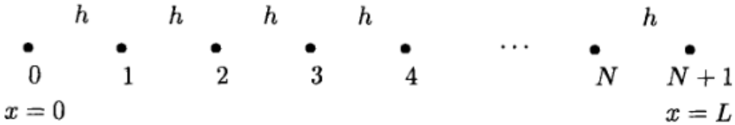

Figure: 一维网格节点

对于内点，采用二阶中心差分离散二阶和一阶导数

$$K_i \frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} + (dK/dx)_i \frac{U_{i+1} - U_{i-1}}{2h} = r_i$$

整理可得

$$\left(\frac{K_i}{h^2} + \frac{dK/dx}{2h} \right) U_{i+1} - 2\frac{K_i}{h^2} U_i + \left(\frac{K_i}{h^2} - \frac{dK/dx}{2h} \right) U_{i-1} = r_i$$

或者，可以表示为

$$A_i U_{i-1} + B_i U_i + C_i U_{i+1} = D_i, \quad i = 1, 2, \cdots, N$$

边界条件处理

Dirichlet 条件：$U(0)=U_0, U(L)=U_L$

$$B_1U_1+C_1U_2=r_1-A_1U_0$$
$$A_NU_{N-1}+B_NU_N=r_N-C_NU_L$$

Neumann 条件：$dU/dx(0)=a, dU/dx(L)=b$

采用一阶差分近似边界导数

$$\frac{U_1-U_0}{h}=a, or \quad U_0=U_1-ah$$
$$\frac{U_{N+1}-U_N}{h}=b, or \quad U_{N+1}=U_N+hb$$

二阶近似，分别采用前向、后向差分

$$\frac{-3U_0+4U_1-U_2}{2h}=a$$
$$\frac{3U_{N+1}-4U_N+U_{N-1}}{2h}=b$$

Shadow node 方法

(1) 节点 N 位于右边界 $x = L$，节点 $N+1$ 向边界外延拓一个网格

二阶中心差分格式

$$\frac{U_{N+1} - U_{N-1}}{2h} = b$$

(2) 边界位于 N 和 $N+1$ 节点中间

$$\frac{U_{N+1} - U_N}{h} = b$$

方程和边界条件离散后得到了三对角的线性方程组，对方程组的求解：
(1) 直接求逆
(2) 追赶法 (Thomas 算法)

Outline

- 椭圆型方程的差分离散格式
  - 两点边值问题
  - Laplace 方程
  - Possion 方程
- 线性方程组的迭代解法
  - 简单迭代算法
- 基本迭代法理论分析
  - 基本理论
  - 基本迭代法构造

考虑定解问题

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad 0 < x, y < 1$$
$$u(x,0) = u(0,y) = u(1,y) = 0, u(x,1) = \sin \pi x.$$

五点格式

$$\frac{u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}-4u_{i,j}}{h^2}=0$$

其截断误差为

$$T_{i,j}=\frac{h^2}{12}(u_{xxxx}+u_{yyyy})|_{i,j}+O(h^4).$$

因此，差分方程的精度为 2 阶. 差分方程可以改写为

$$4u_{i,j}=u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}$$

九点格式

$$\frac{1}{6h^2}[4u_{i+1,j}+4u_{i-1,j}+4u_{i,j+1}+4u_{i,j-1}$$
$$+u_{i-1,j-1}+u_{i-1,j+1}+u_{i+1,j-1}+u_{i+1,j+1}-20u_{i,j}]=0$$

其截断误差为

$$T_{i,j}=\frac{h^2}{12}(u_{xxxx}+2u_{xxyy}+u_{yyyy})|_{i,j}+O(h^4)=\frac{h^2}{12}\Delta^2u|_{i,j}+O(h^4).$$

注意到，对于 Laplace 方程

$$\Delta u=0,\quad \Delta^2u=0,$$

因此差分方程的精度为 4 阶. 差分方程可以改写为

$$\begin{aligned}20u_{i,j}=&4u_{i+1,j}+4u_{i-1,j}+4u_{i,j+1}+4u_{i,j-1}\\&+u_{i-1,j-1}+u_{i-1,j+1}+u_{i+1,j-1}+u_{i+1,j+1}\end{aligned}$$

取网格间距 $\Delta x = \Delta y = h = 1/3$，由五点格式可以得到如下方程组

$$
\begin{aligned}
u_{0,1} + u_{1,0} + u_{2,1} + u_{1,2} - 4u_{1,1} = 0 \\
u_{1,1} + u_{2,0} + u_{3,1} + u_{2,2} - 4u_{2,1} = 0 \\
u_{0,2} + u_{1,1} + u_{2,2} + u_{1,3} - 4u_{1,2} = 0 \\
u_{1,2} + u_{2,1} + u_{3,2} + u_{2,3} - 4u_{2,2} = 0
\end{aligned}
$$

其中边界上的值 $u_{0,1}, u_{1,0}, u_{2,0}, u_{3,1}, u_{0,2}, u_{1,3}, u_{3,2}, u_{2,3}$ 已知.
内部网格点上的值，按照自然排序，记为
$\mathbf{x} = (u_{1,1}, u_{2,1}, u_{1,2}, u_{22})^T$. 方程组表示为矩阵的形式 $\mathbf{Ax} = \mathbf{b}$ 其中 $\mathbf{A} = \begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} u_{0,1} + u_{1,0} \\ u_{2,0} + u_{3,1} \\ u_{0,2} + u_{1,3} \\ u_{2,3} + u_{3,2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}.$

由九点格式得到

$$
\begin{aligned}
20u_{1,1} - 4(u_{2,1} + u_{1,2}) - u_{2,2} &= u_{0,2} + u_{0,0} + u_{2,0} + 4(u_{0,1} + u_{1,0}) \\
20u_{2,1} - 4(u_{1,1} + u_{2,2}) - u_{1,2} &= u_{1,0} + u_{3,0} + u_{3,2} + 4(u_{2,0} + u_{3,1}) \\
20u_{1,2} - 4(u_{1,1} + u_{2,2}) - u_{2,1} &= u_{0,3} + u_{0,1} + u_{2,3} + 4(u_{0,2} + u_{1,3}) \\
20u_{2,2} - 4(u_{2,1} + u_{1,2}) - u_{1,1} &= u_{1,3} + u_{3,1} + u_{3,3} + 4(u_{3,2} + u_{2,3})
\end{aligned}
$$

表示为矩阵形式

$$
\mathbf{A} = \begin{bmatrix} 20 & -4 & -4 & -1 \\ -4 & 20 & -1 & -4 \\ -4 & -1 & 20 & -4 \\ -1 & -4 & -4 & 20 \end{bmatrix},
$$

$$
\mathbf{b} = \begin{bmatrix} u_{0,2} + u_{0,0} + u_{2,0} + 4(u_{0,1} + u_{1,0}) \\ u_{1,0} + u_{3,0} + u_{3,2} + 4(u_{2,0} + u_{3,1}) \\ u_{0,3} + u_{0,1} + u_{2,3} + 4(u_{0,2} + u_{1,3}) \\ u_{1,3} + u_{3,1} + u_{3,3} + 4(u_{3,2} + u_{2,3}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{5\sqrt{3}}{2} \\ \frac{5\sqrt{3}}{2} \end{bmatrix}.
$$

计算线性方程组，得到
五点格式的解为：$\mathbf{x}=(\frac{\sqrt{3}}{16},\frac{\sqrt{3}}{16},\frac{3\sqrt{3}}{16},\frac{3\sqrt{3}}{16})^T\approx$
$(0.108253,0.108253,0.32476,0.32476)^T$,
九点格式的解为：$\mathbf{x}=(\frac{25}{154\sqrt{3}},\frac{25}{154\sqrt{3}},\frac{40}{77\sqrt{3}},\frac{40}{77\sqrt{3}})^T\approx$
$(0.0937257,0.0937257,0.299922,0.299922)^T$,
定解问题的精确解为

$$u(x,y)=\frac{\sin\pi x\sinh\pi y}{\sinh\pi}$$

从而 $\mathbf{x}$ 的精确解为
$\mathbf{x}=(\frac{\sqrt{3}\sinh\frac{\pi}{3}}{\sinh\pi},\frac{\sqrt{3}\sinh\frac{\pi}{3}}{\sinh\pi},\frac{\sqrt{3}\sinh\frac{2\pi}{3}}{\sinh\pi},\frac{\sqrt{3}\sinh\frac{2\pi}{3}}{\sinh\pi})^T\approx$
$(0.0936885,0.0936885,0.299857,0.299857)^T$.

如果 $h = 1/4$，由五点格式得到方程组

$$
\begin{aligned}
u_{0,1} + u_{1,0} + u_{2,1} + u_{1,2} - 4u_{1,1} &= 0 \\
u_{1,1} + u_{2,0} + u_{3,1} + u_{2,2} - 4u_{2,1} &= 0 \\
u_{2,1} + u_{3,0} + u_{4,1} + u_{3,2} - 4u_{3,1} &= 0 \\
u_{0,2} + u_{1,1} + u_{2,2} + u_{1,3} - 4u_{1,2} &= 0 \\
u_{1,2} + u_{2,1} + u_{3,2} + u_{2,3} - 4u_{2,2} &= 0 \\
u_{2,2} + u_{3,1} + u_{4,2} + u_{3,3} - 4u_{3,2} &= 0 \\
u_{0,3} + u_{1,2} + u_{2,3} + u_{1,4} - 4u_{1,3} &= 0 \\
u_{1,3} + u_{2,2} + u_{3,3} + u_{2,4} - 4u_{2,3} &= 0 \\
u_{2,3} + u_{3,2} + u_{4,3} + u_{3,4} - 4u_{3,3} &= 0
\end{aligned}
$$

写成矩阵形式

$$\mathbf{A} = \begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix}$$

$$
\mathbf{b}=\begin{bmatrix} u_{0,1}+u_{1,0} \\ u_{2,0} \\ u_{3,0}+u_{4,1} \\ u_{0,2} \\ 0 \\ u_{4,2} \\ u_{0,3}+u_{1,4} \\ u_{2,4} \\ u_{4,3}+u_{3,4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{\sqrt{2}}{2} \\ 1 \\ \frac{\sqrt{2}}{2} \end{bmatrix}, \quad \mathbf{x}=\begin{bmatrix} 1/224(6+5\sqrt{2}) \\ 1/112(5+3\sqrt{2}) \\ 1/224(6+5\sqrt{2}) \\ 1/16(1+\sqrt{2}) \\ 1/16(2+\sqrt{2}) \\ 1/16(1+\sqrt{2}) \\ 1/224(22+37\sqrt{2}) \\ 1/112(37+11\sqrt{2}) \\ 1/224(22+37\sqrt{2}) \end{bmatrix}.
$$

计算值 $u(1/2,1/2)\approx 0.213388$, 精确解 $u(1/2,1/2)\approx 0.199268$.
$u(1/4,1/4)=0.058353$, $u(1/3,1/3)\approx$
$2/3u(1/4,1.4)+1/3u(1/2,1/2)=0.110031$

由九点格式得到

$$4(u_{0,1}+u_{1,0}+u_{2,1}+u_{1,2})+u_{0,2}+u_{0,0}+u_{2,0}+u_{2,2}-20u_{1,1}=0$$
$$4(u_{1,1}+u_{2,0}+u_{3,1}+u_{2,2})+u_{1,2}+u_{1,0}+u_{3,0}+u_{3,2}-20u_{2,1}=0$$
$$4(u_{2,1}+u_{3,0}+u_{4,1}+u_{3,2})+u_{2,2}+u_{2,0}+u_{4,0}+u_{4,2}-20u_{3,1}=0$$
$$4(u_{0,2}+u_{1,1}+u_{2,2}+u_{1,3})+u_{0,3}+u_{0,1}+u_{2,1}+u_{2,3}-20u_{1,2}=0$$
$$4(u_{1,2}+u_{2,1}+u_{3,2}+u_{2,3})+u_{1,3}+u_{1,1}+u_{3,1}+u_{3,3}-20u_{2,2}=0$$
$$4(u_{2,2}+u_{3,1}+u_{4,2}+u_{3,3})+u_{2,3}+u_{2,1}+u_{4,1}+u_{4,3}-20u_{3,2}=0$$
$$4(u_{0,3}+u_{1,2}+u_{2,3}+u_{1,4})+u_{0,4}+u_{0,2}+u_{2,2}+u_{2,4}-20u_{1,3}=0$$
$$4(u_{1,3}+u_{2,2}+u_{3,3}+u_{2,4})+u_{1,4}+u_{1,2}+u_{3,2}+u_{3,4}-20u_{2,3}=0$$
$$4(u_{2,3}+u_{3,2}+u_{4,3}+u_{3,4})+u_{2,4}+u_{2,2}+u_{4,2}+u_{4,4}-20u_{3,3}=0$$

$$\mathbf{A} = \begin{bmatrix} 20 & -4 & 0 & -4 & -1 & 0 & 0 & 0 & 0 \\ -4 & 20 & -4 & -1 & -4 & -1 & 0 & 0 & 0 \\ 0 & -4 & 20 & 0 & -1 & -4 & 0 & 0 & 0 \\ -4 & -1 & 0 & 20 & -4 & 0 & -4 & -1 & 0 \\ -1 & -4 & -1 & -4 & 20 & -4 & -1 & -4 & -1 \\ 0 & -1 & -4 & 0 & -4 & 20 & 0 & -1 & -4 \\ 0 & 0 & 0 & -4 & -1 & 0 & 20 & -4 & 0 \\ 0 & 0 & 0 & -1 & -4 & -1 & -4 & 20 & -4 \\ 0 & 0 & 0 & 0 & -1 & -4 & 0 & -4 & 20 \end{bmatrix}$$

$$
\mathbf{b} = \begin{bmatrix} 4(u_{0,1} + u_{1,0}) + u_{0,2} + u_{0,0} + u_{2,0} \\ 4u_{2,0} + u_{1,0} + u_{3,0} \\ 4(u_{3,0} + u_{4,1}) + u_{2,0} + u_{4,0} + u_{4,2} \\ 4u_{0,2} + u_{0,3} + u_{0,1} \\ 0 \\ 4u_{4,2} + u_{4,1} + u_{4,3} \\ 4(u_{0,3} + u_{1,4}) + u_{0,4} + u_{0,2} + u_{2,4} \\ 4u_{2,4} + u_{1,4} + u_{3,4} \\ 4(u_{3,4} + u_{4,3}) + u_{2,4} + u_{4,2} + u_{4,4} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 + 2\sqrt{2} \\ 4 + \sqrt{2} \\ 1 + 2\sqrt{2} \end{bmatrix},
$$

$$
\mathbf{x} = \begin{bmatrix} (2601 + 1891\sqrt{2})/99176 \\ (3782 + 2601\sqrt{2})/99176 \\ (2601 + 1891\sqrt{2})/99176 \\ (144 + 113\sqrt{2})/2156 \\ (113 + 72\sqrt{2})/1078 \\ (144 + 113\sqrt{2})/2156 \\ (3(4101 + 4583\sqrt{2}))/99176 \\ (3(9166 + 4101\sqrt{2}))/99176 \\ (3(4101 + 4583\sqrt{2}))/99176 \end{bmatrix} \approx \begin{bmatrix} 0.0531911 \\ 0.0752235 \\ 0.0531911 \\ 0.140912 \\ 0.19928 \\ 0.140912 \\ 0.320108 \\ 0.452701 \\ 0.320108 \end{bmatrix}
$$

Table: 五点格式和九点格式精度比较

	5-points scheme	9-points scheme	analytical solution
(0.25,0.25)	0.058353	0.053191	0.0531871
(0.50,0.25)	0.082523	0.075223	0.0752178
(0.25,0.50)	0.150888	0.140912	0.1409041
(0.50,0.50)	0.213388	0.199280	0.1992681
(0.25,0.75)	0.331812	0.320108	0.3200991
(0.50,0.75)	0.469253	0.452701	0.4526881

Outline

- 椭圆型方程的差分离散格式
  - 两点边值问题
  - Laplace 方程
  - Possion 方程
- 线性方程组的迭代解法
  - 简单迭代算法
- 基本迭代法理论分析
  - 基本理论
  - 基本迭代法构造

考虑定解问题

$$
\begin{aligned}
\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad 0 < x, y < 1 \\
u(x, 0) = f_1(x), u(x, 1) = f_2(x) \\
u(0, y) = g_1(y), u(1, y) = g_2(y).
\end{aligned}
$$

五点格式

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

其截断误差为

$$T_{i,j}=(u_{xx}+u_{yy})|_{i,j}-f_{i,j}+\frac{h^2}{12}(u_{xxxx}+u_{yyyy})|_{i,j}+O(h^4).$$

因此，差分方程的精度为 2 阶. 差分方程可以改写为

$$u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}-4u_{i,j}=h^2f_{i,j}$$

九点格式

$$
\begin{aligned}
\frac{1}{6h^2}[4u_{i+1,j} + 4u_{i-1,j} + 4u_{i,j+1} + 4u_{i,j-1} \\
+ u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} + u_{i+1,j+1} - 20u_{i,j}] = f_{i,j}
\end{aligned}
$$

其截断误差为

$$
T_{i,j} = (u_{xx} + u_{yy})|_{i,j} - f_{i,j} + \frac{h^2}{12}(u_{xxxx} + 2u_{xxyy} + u_{yyyy})|_{i,j} + O(h^4).
$$

注意到，若 f 是调谐函数，即 $\Delta f = 0$，则

$$
\Delta^2 u = \Delta f = 0,
$$

因此差分方程的精度为 4 阶，否则只有二阶精度. 差分方程可以改写为

$$
\begin{aligned}
4(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) + u_{i-1,j-1} + u_{i-1,j+1} \\
+ u_{i+1,j-1} + u_{i+1,j+1} - 20u_{i,j} = 6h^2 f_{i,j}
\end{aligned}
$$

构造如下的差分格式

$$
\begin{aligned}
\frac{1}{6h^2}[4(&u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1})\\
&+u_{i-1,j-1}+u_{i-1,j+1}+u_{i+1,j-1}+u_{i+1,j+1}-20u_{i,j}]\\
&=f_{i,j}+\frac{h^2}{12}\Delta f(x_i,y_j)
\end{aligned}
$$

用五点差分格式离散 $\Delta f(x_i, y_j)$，则差分方程具有四阶精度.
差分方程为

$$
\begin{aligned}
4(u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1})&+u_{i-1,j-1}+u_{i-1,j+1}\\
&+u_{i+1,j-1}+u_{i+1,j+1}-20u_{i,j}\\
&=\frac{h^2}{2}(f_{i+1,j}+f_{i-1,j}+f_{i,j+1}+f_{i,j-1}+8f_{i,j})
\end{aligned}
$$

由五点差分格式离散具有 *Dirichlet* 边界条件的 *Possion* 方程得到的线性方程组为

$$\mathbf{A}_5\mathbf{x} = \mathbf{b}_5$$

其中

$$\mathbf{A}_5 = \begin{bmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & T \end{bmatrix}.$$

由九点差分格式离散具有 *Dirichlet* 边界条件的 *Possion* 方程得到的线性方程组为

$$\mathbf{A}_9\mathbf{x} = \mathbf{b}_9$$

其中

$$\mathbf{A}_9 = \begin{bmatrix} T_1 & -T_2 & & & \\ -T_2 & T_1 & -T_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -T_2 & T_1 & -T_2 \\ & & & -T_2 & T_1 \end{bmatrix}.$$

Outline

Jacobi 迭代

(1) 将线性方程组

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \quad i = 1, 2, \cdots, n,$$

改写为

$$x_i = \frac{1}{a_{ii}} \left(\sum_{j=1, j\neq i}^{n} -a_{ij}x_j + b_i \right) \quad i = 1, 2, \cdots, n$$

(2) 任意选择一组初始近似值 $x_1^{(0)}, x_2^{(0)}, \cdots, x_n^{(0)}$, 作为方程的第 0 次近似解.

(3) 依次使 $k = 1, 2, 3, \cdots$, 用公式

$$x_i^{(k)} = \frac{1}{a_{ii}} \left(\sum_{j=1, j\neq i}^{n} -a_{ij}x_j^{(k-1)} + b_i \right) \quad i = 1, 2, \cdots, n$$

求出方程的第 k 次近似解，直至满足

$$\max_{1\le i\le n} |x_i^{(k)} - x_i^{(k-1)}| < \epsilon$$

其中 ϵ 是预先给定的允许误差.

考虑 $Possion$ 方程的五点差分格式

$$u_{ii} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - h^2 f_{i,j}}{4}$$

Jacobi 迭代公式表示为

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

对于前例中的 Laplace 方程，用 Jacobi 迭代求解

$$
\begin{aligned}
u_{11}^{(k+1)} &= \frac{u_{21}^{(k)} + u_{12}^{(k)} + u_{01} + u_{10}}{4} \\
u_{21}^{(k+1)} &= \frac{u_{11}^{(k)} + u_{22}^{(k)} + u_{20} + u_{31}}{4} \\
u_{12}^{(k+1)} &= \frac{u_{22}^{(k)} + u_{11}^{(k)} + u_{13} + u_{02}}{4} \\
u_{22}^{(k+1)} &= \frac{u_{12}^{(k)} + u_{21}^{(k)} + u_{23} + u_{32}}{4}
\end{aligned}
$$

取初始估计值 $(u_{11}^{(0)}, u_{21}^{(0)}, u_{12}^{(0)}, u_{22}^{(0)})^T = (0,0,0,0)^T$.

Table: Jacobi 迭代过程

k	u_{11}	u_{22}
0	0	0
1	0	0.216506
2	0.0541266	0.270633
3	0.0811899	0.297696
4	0.0947215	0.311228
5	0.101487	0.321377
6	0.106562	0.323068
7	0.107407	0.323914
8	0.10783	0.324337
9	0.108042	0.324548
10	0.108147	0.324654
11	0.1082	0.324707
12	0.108227	0.324746

Gauss-Seidel 迭代

$$x_i^{(k)} = \frac{1}{a_{ii}} \left(-\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} + b_i \right) \quad i = 1, 2, \cdots, n$$

Table: Gauss-Seidel 迭代过程

k	u_{11}	u_{21}	u_{12}	u_{22}
0	0	0	0	0
1	0	0	0.216506	0.270633
2	0.0541266	0.0811899	0.297696	0.311228
3	0.0947215	0.101487	0.317994	0.321377
4	0.10487	0.106562	0.323068	0.323914
5	0.107407	0.10783	0.324337	0.324548
6	0.108042	0.108147	0.324654	0.324707
7	0.1082	0.108227	0.324733	0.324746
8	0.10824	0.108247	0.324753	0.324756
9	0.10825	0.108252	0.324758	0.324759
10	0.108252	0.108253	0.324759	0.324759
11	0.108253	0.108253	0.324759	0.324759
12	0.108253	0.108253	0.32476	0.32476
13	0.108253	0.108253	0.32476	0.32476

Successive Over Relaxation, SOR

$$x_i^{(k)} = \omega \frac{1}{a_{ii}} \left(-\sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} + b_i \right) + (1-\omega) x_i^{k-1}$$

其中 ω 为松弛因子，$1 < \omega < 2$.

Table: SOR 迭代过程 ($\omega = 1.071796770$)

k	u_{11}	u_{21}	u_{12}	u_{22}
0	0	0	0	0
1	0	0	0.232051	0.294229
2	0.0621778	0.0954988	0.310889	0.319817
3	0.104427	0.106819	0.323406	0.324368
4	0.107781	0.108125	0.324625	0.324717
5	0.108217	0.108241	0.324748	0.324756
6	0.10825	0.108252	0.324759	0.324759
7	0.108253	0.108253	0.324759	0.32476
8	0.108253	0.108253	0.32476	0.32476

Table: 迭代方法比较，$n = 2$, ϵ=1E-12

方法	迭代次数
Jacobi	39
G-S	21
SOR	13

问题

(1) 在什么情况下，迭代解能够逼近精确解 (迭代法的收敛性)?

(2) 迭代法的收敛速度取决于什么?

Outline

线性方程组

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$\mathbf{A}$ 是 $n \times n$ 方阵.

- 矩阵分裂 $\mathbf{A} = \mathbf{M} - \mathbf{N}$，$\mathbf{M}^{-1}$ 存在.
- 令 $\mathbf{H} = \mathbf{M}^{-1}\mathbf{N}$，$\mathbf{d} = \mathbf{M}^{-1}\mathbf{b}$.
- 对于任意初值 $\mathbf{x}(0)_{n\times 1}$，构造线性迭代

$$\mathbf{x}^{(k+1)} = \mathbf{H}\mathbf{x}^{(k)} + \mathbf{d}, \quad k = 1, 2, \cdots .$$

- 如果 $\rho(\mathbf{H}) < 1$，则 $\mathbf{A}$ 非奇异，且

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad \forall \mathbf{x}(0)$$

Proof.
分裂 $\mathbf{A} = \mathbf{M} - \mathbf{N} = \mathbf{M}(\mathbf{I} - \mathbf{H})$，
$\rho(\mathbf{H}) < 1$，因此 $(\mathbf{I} - \mathbf{H})^{-1}$ 存在，故 $\mathbf{A}$ 非奇异.

$$\mathbf{x}^{(k)} = \mathbf{H}^{(k)}\mathbf{x}(0) + (\mathbf{I} + \mathbf{H} + \mathbf{H^2} + \cdots + \mathbf{H^{k-1}})\mathbf{d}$$

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = (\mathbf{I} - \mathbf{H})^{-1}\mathbf{d} = (\mathbf{I} - \mathbf{H})^{-1}\mathrm{M}^{\mathbf{-1}}\mathbf{b} = \mathbf{A}^{-\mathbf{1}}\mathbf{b} = \mathbf{x}$$

$\square$

$$\mathbf{x}^{(k+1)} = \mathbf{H}\mathbf{x}^{(k)} + \mathbf{d}$$

由上式

$$\mathbf{x} - \mathbf{x}^{(k+1)} = \mathbf{x} - \mathbf{x}^{(k)} + (\mathbf{I} - \mathbf{H})\mathbf{x}^{(k)} - \mathbf{M}^{-1}\mathbf{A}\mathbf{x}$$

因为 $\mathbf{M}^{-1}\mathbf{A} = (\mathbf{I} - \mathbf{H})$

$$\mathbf{x} - \mathbf{x}^{(k+1)} = \mathbf{H}(\mathbf{x} - \mathbf{x}^{(k)})$$

记 $\mathbf{x} - \mathbf{x}^{(k)} = \mathbf{e}^{(k)}$

$$\mathbf{e}^{(k+1)} = \mathbf{H}\mathbf{e}^{(k)}$$

$\mathbf{H} = \mathbf{M}^{-1}\mathbf{N}$ 称为迭代矩阵.

若 $\rho(\mathbf{H}) < 1$，则

$$\frac{\mathbf{e}^{(k+m)}}{\mathbf{e}^{(k)}} \approx \rho(\mathbf{H})^m$$

因此，误差下降 10^{-q}，近似需要 $m = -\frac{q}{\log_{10}\rho(\mathbf{H})}$ 次迭代.

平均收敛速度

$$R(\mathbf{H}^k) = -\frac{1}{k}\log_{10}\|\mathbf{H}\|$$

渐近收敛速度

$$\lim_{k\to\infty} R(\mathbf{H}^k) = -\log_{10}\rho(\mathbf{H}) \triangleq R_\infty(\mathbf{H})$$

今由迭代矩阵 $\mathbf{H}_1, \mathbf{H}_2$. 若 $\rho(\mathbf{H}_1) < \rho(\mathbf{H}_2) < 1$，则渐近收敛速度 $R_\infty(\mathbf{H}_1) > R_\infty(\mathbf{H}_2)$.

Outline

矩阵 A 分裂为

$$A = L + D + U$$

Jacobi 迭代的分裂矩阵

$$M = D, \quad N = -(L + U)$$

Jacobi 迭代表示为矩阵形式

$$\mathbf{x}^{(k+1)} = -D^{-1}(L + U)\mathbf{x}^{(k)} + D^{-1}\mathbf{b}$$

迭代矩阵 $H = -D^{-1}(L + U)$

Gauss-Seidel 迭代的分裂矩阵

$$M = D + L, \quad N = -U$$

Gauss-Seidel 迭代表示为矩阵形式

$$\mathbf{x}^{(k+1)} = -(D + L)^{-1} U\mathbf{x}^{(k)} + (D + L)^{-1}\mathbf{b}$$

迭代矩阵 $H = -(D + L)^{-1} U$

SOR 迭代的分裂矩阵

$$M = D/\omega + L, \quad N = \left(\frac{1}{\omega} - 1\right) D - U$$

SOR 迭代表示为矩阵形式

$$\mathbf{x}^{(k+1)} = (I + \omega D^{-1} L)^{-1}[(1-\omega) I - \omega D^{-1} U]\mathbf{x}^{(k)} + (I + \omega D^{-1} L)^{-1} \omega D^{-1}\mathbf{b}$$

整理后

$$\mathbf{x}^{(k+1)} = (D/\omega + L)^{-1}[(\omega^{-1} - 1) D - U]\mathbf{x}^{(k)} + (I + \omega D^{-1} L)^{-1} \omega D^{-1}\mathbf{b}$$

迭代矩阵 $H = (D/\omega + L)^{-1}[(\omega^{-1} - 1) D - U]$

迭代法的收敛性

Theorem

若 A 严格对角占优，则 *Jacobi* 迭代和 *Gauss-Seidel* 迭代对任意的初值都收敛.

Theorem

SOR 迭代矩阵满足

$$\rho(H_{SOR}) \geq |\omega - 1|$$

从而 SOR 迭代收敛的一个必要条件是

$$0 < \omega < 2$$

Theorem

若 A 对称正定，则 *SOR* 迭代当且仅当 $0 < \omega < 2$ 时收敛.

当 $\omega = 1$ 时，SOR 迭代即为 Gauss-Seidel 迭代，因此当 A 正定，Gauss-Seidel 迭代收敛.

Laplace 方程五点差分格式

五点差分格式离散 Laplace 得到的线性方程组的 $A_{n^2\times n^2}$ 是正定矩阵.

Jacobi 迭代的迭代矩阵的谱半径

$$\rho(H_J)=\cos(\pi h)$$

Gauss-Seidel 迭代的迭代矩阵谱半径

$$\rho(H_{GS})=\cos^2(\pi h)$$

SOR 迭代的迭代矩阵的最优谱半径

$$\rho(H_{SOR})=\frac{1-\sin(\pi h)}{1+\sin(\pi h)}$$

当松弛因子

$$\omega_{opt}=1+\rho(H_{SOR})$$

取得.

以上格式中 $h=1/(n+1)$

求 $\rho(\mathbf{H}_J)$

$$-D^{-1}(L+U)\mathbf{x}=\lambda\mathbf{x}$$

整理得到

$$[\lambda D+(L+U)]x=0$$

因为是采用五点格式，上式写成分量形式

$$4\lambda x_{i,j}+(-x_{i,j-1}-xi-1,j-x_{i+1,j}-x_{i,j+1})=0$$

上式为差分方程，用分离变量法求其特征值. 令 $x_{i,j}=y_iz_j$，代入上式

$$4\lambda y_iz_j-y_i(z_{j-1}+z_{j+1})=z_j(y_{i+1}+y_{i-1})$$

方程两边同除以 y_iz_j，得到

$$\frac{y_{i-1}+y_{i+1}}{y_i}=4\lambda-\frac{z_{j-1}+z_{j+1}}{z_j}=\mu$$

$$y_{i-1} + y_{i+1} = \mu y_i$$
$$z_{j-1} + z_{j+1} = (4\lambda - \mu) z_j$$

$$\begin{bmatrix} 0 & 1 & & & & \\ 1 & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & & \ddots & \ddots & \ddots \\ & & & 1 & 0 & 1 \\ & & & & 1 & 0 \end{bmatrix} \mathbf{Y} = \mu \mathbf{Y}$$

可求得 $\mu = 2\cos s\pi h$. 同样可得 $(4\lambda - \mu) = 2\cos s\pi h$ 故

$$\lambda = \cos s\pi h, \quad s = 1, 2, \cdots, n-1$$

从而 $\rho(H_J) = \cos(\pi h)$.

对于前例当 $n=99$ 时，用三种迭代法进行了计算.

Table: 迭代方法比较，$n=99$, ϵ=1E-12

方法	迭代次数
Jacobi	46164
G-S	23180
SOR($\omega=1.9391$)	533

SOR 迭代法仅在特殊情形下才能求得最佳松弛因子. 对于一般的线性方程组，需要通过试验获得较好的收敛速率.