

Stokes 方程的数值解法

王禹涵 2200010866

2024 年 11 月 26 日

1 基本理论

本文介绍 Navier-Stokes 方程

$$\begin{cases} -\Delta u + \nabla p = f \\ \operatorname{div} u = 0 \\ \frac{\partial u}{\partial n}, \frac{\partial v}{\partial n} \text{ 于边界上已知, } \mathbf{n} \text{ 为边界外法向} \end{cases}$$

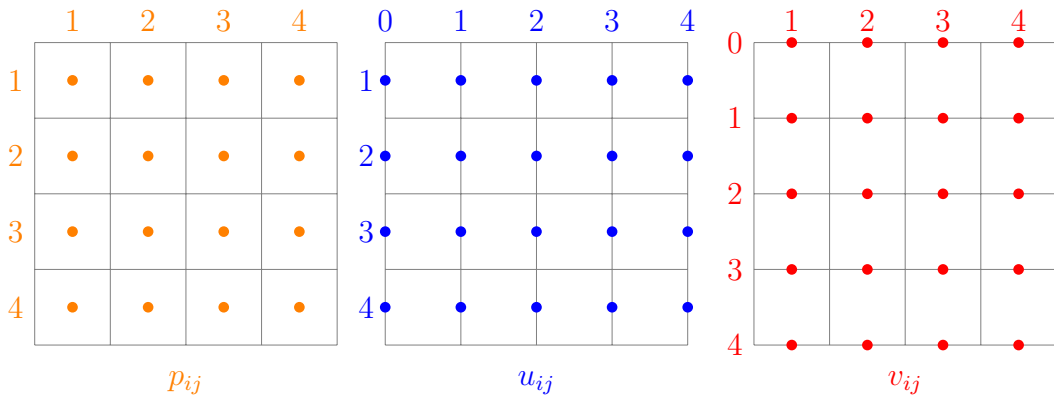
的几种数值解法:

1.1 MAC 交错网格

MAC, Marker and Cell 网格是按如下方式对 (u, v, p) 做的离散近似:

- (1) 首先把待求解区域 Ω 进行 N 等分得到 $N \times N$ 的网格, 左上角为 $(0, 0)$, 水平轴 i 轴, 竖直轴 j 轴。
- (2) 以每个网格的中心点为压力 p 的取样点, 第 i 列、 j 行处的取样点记为 p_{ij} , $1 \leq i, j \leq N$ 。
- (3) 以每条竖直线段的中点为水平速度 u 的取样点, 第 i 列、 j 行处的取样点记为 u_{ij} , $0 \leq i \leq N, 1 \leq j \leq N$ 。
- (4) 以每条水平线段的中点为数值速度 v 的取样点, 第 i 列、 j 行处的取样点记为 v_{ij} , $1 \leq i \leq N, 0 \leq j \leq N$ 。

以 $N = 4$ 为例:



把 u, v, p 按如下方式排列:

$$U = [u_{11}, \dots, u_{14}, \dots, u_{31}, \dots, u_{34}]^T \in \mathbb{R}^{(N-1)N}$$

$$V = [v_{11}, \dots, v_{13}, \dots, v_{41}, \dots, v_{43}]^T \in \mathbb{R}^{N(N-1)}$$

$$P = [p_{11}, \dots, p_{14}, \dots, p_{41}, \dots, p_{44}]^T \in \mathbb{R}^{N^2}$$

关于 U, P 的方程: $A_1 U + B_1 P = F_1$, 其中:

$$A_1 = \frac{1}{h^2} \begin{bmatrix} X & -I_{4 \times 4} \\ -I_{4 \times 4} & X & -I_{4 \times 4} \\ & -I_{4 \times 4} & X \end{bmatrix} \in M_{(N-1)N \times (N-1)N}(\mathbb{R}), \quad X = \begin{bmatrix} 3 & -1 & & \\ -1 & 4 & -1 & \\ & -1 & 4 & -1 \\ & & -1 & 3 \end{bmatrix} \in M_{N \times N}(\mathbb{R})$$

而

$$B_1 = \frac{1}{h} \begin{bmatrix} -I_{4 \times 4} & I_{4 \times 4} & & \\ & -I_{4 \times 4} & I_{4 \times 4} & \\ & & -I_{4 \times 4} & I_{4 \times 4} \end{bmatrix} \in M_{N(N-1) \times N^2}(\mathbb{R})$$

同样的, 关于 V, P 的方程: $A_2 V + B_2 P = F_2$, 其中:

$$A_2 = \frac{1}{h^2} \begin{bmatrix} S & -I_{3 \times 3} \\ -I_{3 \times 3} & R & -I_{3 \times 3} \\ & -I_{3 \times 3} & R & -I_{3 \times 3} \\ & & -I_{3 \times 3} & S \end{bmatrix}, \quad S = \begin{bmatrix} 3 & -1 & \\ -1 & 3 & -1 \\ & -1 & 3 \end{bmatrix}, \quad R = \begin{bmatrix} 4 & -1 & \\ -1 & 4 & -1 \\ & -1 & 4 \end{bmatrix}$$

而

$$B_2 = \frac{1}{h} \begin{bmatrix} Q & & & \\ & Q & & \\ & & Q & \\ & & & Q \end{bmatrix}, \quad Q = \begin{bmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \end{bmatrix}$$

其中,

$$Q^T Q = \begin{bmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 1 \end{bmatrix}, \quad Q Q^T = \begin{bmatrix} 2 & -1 & \\ -1 & 2 & -1 \\ & -1 & 2 \end{bmatrix}$$

就不难看出:

$$A_1 = I_{3 \times 3} \otimes Q^T Q + Q Q^T \otimes I_{4 \times 4}, \quad A_2 = Q^T Q \otimes I_{3 \times 3} + I_{4 \times 4} \otimes Q Q^T$$

并且:

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} Q \otimes I_{4 \times 4} \\ I_{4 \times 4} \otimes Q \end{bmatrix}$$

引理 1.1. (1) $Q Q^T$ 的全部特征值是:

$$\{2(1 - \cos \frac{k}{N} \pi)\}_{k=1:N-1}$$

(2) $Q^T Q$ 的全部特征值是:

$$\{2(1 - \cos \frac{k}{N} \pi)\}_{k=1:N-1} \cup \{0\}$$

证明. (1) All you need is attention. 注意到 $\lambda_k = 2(1 - \cos \frac{k}{N} \pi)$ 对应的特征向量是

$$u_k = [\sin(\frac{k\pi}{N}), \sin(\frac{2k\pi}{N}), \dots, \sin(\frac{(N-1)k\pi}{N})]^T$$

代入验证即可。

(2) 首先 $[1, \dots, 1]^T \in \ker Q^T Q$, 所以有一个零特征值。事实上, $\text{im}(Q) = \mathbb{R}^{N-1}$ (你总能根据差分结果通过求和构造出原像), 所以 $\dim \ker Q = 1$, 并且 $\text{rank}(Q^T) = N - 1$, 所以 $\ker Q = \ker Q^T Q = \langle [1, \dots, 1]^T \rangle$, $[1, \dots, 1]^T$ 是单特征值。然后,

$$Q^T Q v = \lambda v \Rightarrow (Q Q^T)(Q v) = \lambda(Q v) \Rightarrow \lambda \in \lambda(Q Q^T)$$

假设 $Q^T Q$ 存在某个特征子空间的维数至少为 2, 亦即存在线性无关的 v_1, v_2 使得 $Q \langle v_1, v_2 \rangle = \langle v_k \rangle$, 那么说明 $\ker Q \subset \langle v_1, v_2 \rangle$, 矛盾! 所以 $Q^T Q$ 全是单特征值, 那么剩下 $N - 1$ 个特征值取遍 $Q Q^T$ 的所有特征值。□

2 DGS 方法

2.1 DGS 迭代过程

Algorithm 1 DGS 迭代

Require: $k \geq 0$, $[X^k, P^k]$ (1) 松弛速度方程:

$$X^{k+\frac{1}{2}} = X^k + \hat{A}^{-1}(F - BP^k - AX^k)$$

(2) 松弛变形后的连续性方程:

$$\delta q = \hat{A}_p^{-1}(-D - BX^{k+\frac{1}{2}})$$

(3) 更新速度和压力:

$$X^{k+1} = X^{k+\frac{1}{2}} + B\delta q$$

$$P^{k+1} = P^k - B^T B \delta q$$

return $[X^{k+1}, P^{k+1}]$

给定初始值 $X_0 = [U_0^T, V_0^T]^T$, P_0 , 令 $k = 0$, $A = D_A - L_A - U_A$, DGS 迭代过程定义如下:

(1) 先对速度方程 $AX = F - BP$ 进行一次 Gauss-Seidel 迭代:

$$[U_{k+\frac{1}{2}}^T, V_{k+\frac{1}{2}}^T]^T = X_{k+\frac{1}{2}} = X_k + (D_A - L_A)^{-1}(F - BP_k - AX_k)$$

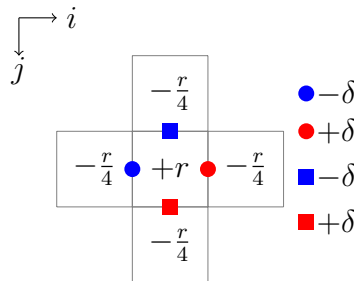
(2) 每个单元格处的数值散度计算公式: 记单元格 (i, j) 为左上角 $(i-1, j-1)$ 的单元格, 而目前该单元格边界上的速度是 $u_{i-1,j}, u_{i,j}, v_{i,j-1}, v_{i,j}$ (一开始是 $X_{k+\frac{1}{2}}$, 后来可能变化), 那么其散度的计算公式为:

$$r_{ij} := -\frac{u_{i,j} - u_{i-1,j}}{h} - \frac{v_{i,j} - v_{i,j-1}}{h}$$

(3) 对于内部单元格 (左上角 $(i-1, j-1)$, $2 \leq i, j \leq N-1$), 按如下方式更新内部速度和压力:

- (i) 计算散度 r_{ij} , 并且令 $\delta := \frac{r_{ij}h}{4}$ 。
- (ii) 更新单元格边界上的速度: 水平轴上水平坐标大、竖直轴上竖直坐标大的那两个 $+\delta$, 剩下两个 $-\delta$ 。
- (iii) 更新本单元格及周围四个单元格的压力: 本单元格的压力 $+r_{ij}$, 其余四格压力均 $-\frac{r_{ij}}{4}$ 。

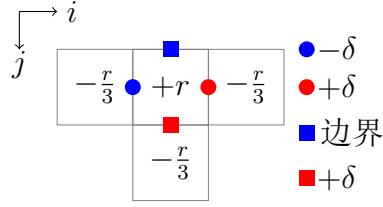
图解:



(4) 对于非四个角落的边界单元格, 按如下方式更新内部速度和压力:

- (i) 计算散度 r_{ij} , 并且令 $\delta := \frac{r_{ij}h}{3}$ 。
- (ii) 更新单元格边界上的速度: 水平轴上水平坐标大、竖直轴上竖直坐标大的那些 $+\delta$, 剩下那些 $-\delta$ 。
- (iii) 更新本单元格及周围三个单元格的压力: 本单元格的压力 $+r_{ij}$, 其余四格压力均 $-\frac{r_{ij}}{3}$ 。

以 $j = 1, 2 \leq i \leq N-1$ 为例的图解:



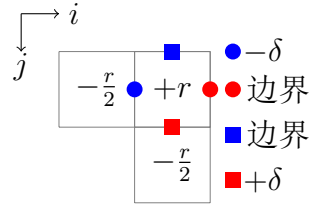
(5) 对于四个角落的单元格，按如下方式更新内部速度和压力：

(i) 计算散度 r_{ij} ，并且令 $\delta := \frac{r_{ij}h}{2}$ 。

(ii) 更新单元格边界上的速度：水平轴上水平坐标大、竖直轴上竖直坐标大的那些 $+\delta$ ，剩下那些 $-\delta$ 。

(iii) 更新本单元格及周围两个单元格的压力：本单元格的压力 $+r_{ij}$ ，其余四格压力均 $-\frac{r_{ij}}{2}$ 。

以 $j = 1, i = N$ 为例的图解：



(6) 单元格的更新顺序：红黑格迭代。将二维网格点依次做红黑记号，如图所示。在计算过程中，对未知量的值进行更新时，我们先更新红色节点，此时所使用的只是黑色节点的数据，然后再更新黑色节点，这时使用的是红色节点的数据。

2.2 使用 DGS 作为磨光子的 VCycle 多重网格方法

注意：虽然我们使用 VCycle 多重网格方法，使用 DGS 迭代作为磨光子来求解方程，那么在对残量方程进行限制或提升后，新方程组一般不满足数值散度为 0，新方程组将形如

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} X \\ P \end{bmatrix} = \begin{bmatrix} F \\ D \end{bmatrix}, \quad D = [d_{ij}]_{i,j}, \quad \text{排列方式同 } P$$

此时残量的计算公式是：

$$r_{ij} := -\frac{u_{i,j} - u_{i-1,j}}{h} - \frac{v_{i,j} - v_{i,j-1}}{h} - d_{ij}$$

二、我们使用的提升算子是：

$$\begin{aligned} u_{2i,2j-1}^h &= u_{2i,2j}^h = u_{ij}^{2h} \\ u_{2i+1,2j-1}^h &= u_{2i+1,2j}^h = \frac{u_{ij}^{2h} + u_{i+1,j}^{2h}}{2} \\ v_{2i-1,2j}^h &= v_{2i,2j}^h = v_{ij}^{2h} \\ v_{2i-1,2j+1}^h &= v_{2i,2j+1}^h = \frac{v_{ij}^{2h} + v_{i,j+1}^{2h}}{2} \\ p_{2i-1,2j-1}^h &= p_{2i,2j-1}^h = p_{2i-1,2j}^h = p_{2i,2j}^h = p_{i,j}^{2h} \end{aligned}$$

使用的限制算子是：

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

由此，令

$$\begin{aligned} L_N &= \frac{1}{2} \begin{bmatrix} 1 & 1 & & & & \\ & & 1 & 1 & & \\ & & & & \ddots & \ddots \\ & & & & & 1 & 1 \end{bmatrix} \in M_{\frac{N}{2} \times N}(\mathbb{R}) \\ K_N &= \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 & & & \\ & & 1 & 2 & 1 & \\ & & & \ddots & \ddots & \ddots \\ & & & & 1 & 2 & 1 \end{bmatrix} \in M_{(\frac{N}{2}-1) \times (N-1)}(\mathbb{R}) \end{aligned}$$

便得到了水平速度的限制算子与提升算子：

$$I_{U,h}^{2h} = K_N \otimes L_N, \quad I_{U,2h}^h = 4I_{U,h}^{2h^T}$$

竖直速度的限制算子与提升算子：

$$I_{V,h}^{2h} = L_N \otimes K_N, \quad I_{V,2h}^h = 4I_{V,h}^{2h^T}$$

压力的限制算子与提升算子：

$$I_{P,h}^{2h} = L_N \otimes L_N, \quad I_{P,2h}^h = 4I_{P,h}^{2h^T}$$

三、实际上，考虑到下一层的系数矩阵

$$I_h^{2h} \begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} I_{2h}^h$$

的形式很奇怪，并不是与上一层的系数矩阵具有相同形式，所以实际的代码实现中我们采用与上一层的系数矩阵具有相同形式的矩阵来进行近似。数值实验结果表明，这样做并不影响收敛性。

2.3 数值试验结果

数值算例：区域 $\Omega = (0, 1) \times (0, 1)$ ，全部条件为：

$$\begin{cases} f(x, y) = -4\pi^2(2 \cos(2\pi x) - 1) \sin(2\pi y) + x^2 \\ g(x, y) = 4\pi^2(2 \cos(2\pi y) - 1) \sin(2\pi x) \\ b(x, 0) = -2\pi(1 - \cos(2\pi x)) \\ t(x, 1) = 2\pi(1 - \cos(2\pi x)) \\ l(0, y) = 2\pi(1 - \cos(2\pi y)) \\ r(1, y) = -2\pi(1 - \cos(2\pi y)) \end{cases}$$

真解为：

$$\begin{cases} u(x,y) = (1 - \cos(2\pi x)) \sin(2\pi y) \\ v(x,y) = -(1 - \cos(2\pi y)) \sin(2\pi x) \\ p(x,y) = \frac{x^3}{3} - \frac{1}{12} \end{cases}$$

对于 $N = 64, 128, 256, 512, 1024$ ，取初值 $P_0(x,y) = x + y$ ，采用 DGS 方法求解离散问题 (*)，停机标准为 $\frac{\|r_h\|_2}{\|r_0\|_2} < 10^{-8}$ 。将程序的 CPU 用时、数值解与真解的误差 e_N 、误差收敛阶 $r = \log e_N / \log N$ 列表和作图如下：其中

$$e_N = \frac{1}{N} \sqrt{\sum_{i,j} |U[i,j] - u(x_i,y_j)|^2 + \sum_{i,j} |V[i,j] - v(x_i,y_j)|^2}$$

N	用时 (s)	误差 e_N	收敛阶 r
64	0.0093	0.0015	-1.5635
128	0.0114	3.7364×10^{-4}	-1.6266
256	0.0355	9.3408×10^{-5}	-1.6733
512	0.1920	2.3359×10^{-5}	-1.7096
1024	1.3544	5.8488×10^{-6}	-1.7386
2048	7.4119	1.4479×10^{-6}	-1.7624

将 e_N 作于离散对数图中：

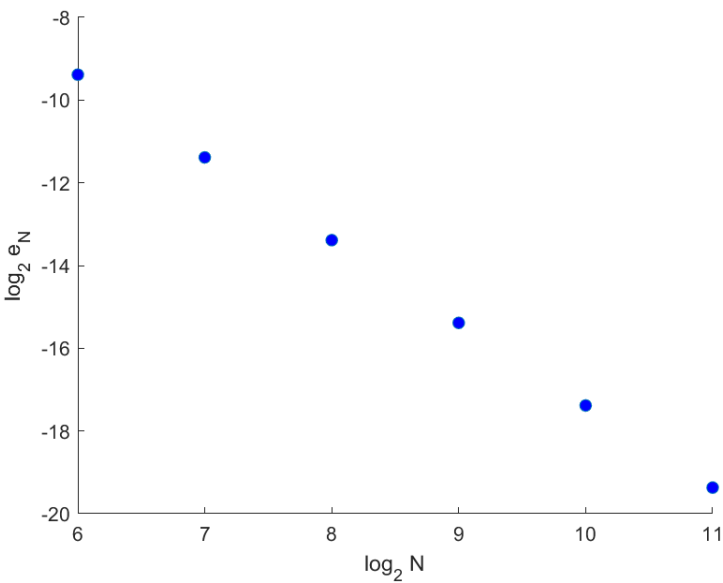


图 1: Caption

使用 Matlab 计算得到其回归直线的斜率是：

$$k = -1.9972$$

可见，误差的收敛速度为二次的。

3 Uzawa 方法

3.1 Uzawa 方法

Uzawa 方法定义为如下的迭代过程:

给定 P_0 , 令 $k = 0$;

(1) 求解 $AX_{k+1} = F - BP_k$;

(2) 更新压力 $P_{k+1} = P_k + \alpha(B^T U_{k+1})$;

(3) 如果误差小于给定的值, 那么停止迭代; 否则回到 (1)。

3.2 最优参数 $\alpha^* = 1$ 的选取

定理 3.1. 使得迭代矩阵 M 的谱半径最小的参数是

$$\alpha^* = \frac{2}{\lambda_{\min}(S) + \lambda_{\max}(S)}$$

其中 $S = B^T A^{-1} B$ 。

定理 3.2. 对于 $S = B^T A^{-1} B$, 成立

$$\lambda(S) = \{0\} \cup \{1, \dots, 1\}_{N^2-1 \text{ 重}}$$

证明. 首先我们来验证

$$AB = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix} \begin{bmatrix} Q \otimes I_N \\ I_N \otimes Q \end{bmatrix} = \begin{bmatrix} Q \otimes Q^T Q + Q Q^T Q \otimes I_N \\ Q^T Q \otimes Q + I_N \otimes Q Q^T Q \end{bmatrix}$$

和

$$BB^T B = \begin{bmatrix} Q \otimes I_N \\ I_N \otimes Q \end{bmatrix} \begin{bmatrix} Q^T \otimes I_N & I_N \otimes Q^T \end{bmatrix} \begin{bmatrix} Q \otimes I_N \\ I_N \otimes Q \end{bmatrix} = \begin{bmatrix} Q Q^T Q \otimes I_N + Q \otimes Q^T Q \\ Q^T Q \otimes Q + I_N \otimes Q Q^T Q \end{bmatrix}$$

从而我们就得到了一个重要的结果: $AB = BB^T B$ 。变形一下:

$$B^T B = (B^T A^{-1})AB = (B^T A^{-1} B)B^T B \Rightarrow (I - S)B^T B = 0$$

设 $Q^T Q$ 的全部特征值是 $\{0, \lambda_1, \dots, \lambda_{N-1}\}$, 设正交矩阵 P_1 把 $Q^T Q$ 正交对角化, 即 $P_1^T Q^T Q P_1 = D = \text{diag}\{0, \lambda_1, \dots, \lambda_{N-1}\}$ 。其中, $\lambda_k = 2(1 - \cos \frac{k\pi}{N})$, 记 $\lambda_0 = 0$ 。取正交矩阵 $P_2 = P_1 \otimes P_1$ 就得到:

$$P_2^T (B^T B) P_2 = (P_1^T \otimes P_1^T) (Q^T Q \otimes I_N + I_N \otimes Q^T Q) (P_1 \otimes P_1) = D \otimes I_N + I_N \otimes D = \tilde{D} = \text{diag}\{\lambda_i + \lambda_j\}_{i,j=0:N-1}$$

所以 \tilde{D} 的对角元中只有一个 0, 其余全部非零。我们令 $K = P_2^T S P_2$, 由上就有 $(I - K)\tilde{D} = 0$, 代入 e_2, \dots, e_{N^2} , 得到:

$$(I - K)e_k = 0, \quad k = 2, \dots, N^2$$

换言之, K 有 $N^2 - 1$ 重特征值 1, 而显然 $[1, \dots, 1]^T$ 在 S 的零空间中, 于是 S 的全部特征值就是:

$$\{0\} \cup \{1, \dots, 1\}_{N^2-1 \text{ 重}}$$

□

但是，如果考虑这个零特征值，我们就会发现迭代过程是不收敛的。我们可以这么想：如果 P 是符合方程的一个压力解，那么显然 $P + \langle [1, \dots, 1]^T \rangle = P + \ker S$ 都符合离散 N-S 方程！所以我们对压力的迭代，实际上需要限制在 $V/\ker S$ 上进行。刨去这个零空间，所以最优参数就是：

$$\alpha^* = \frac{2}{1+1} = 1$$

3.3 数值试验

数值算例：区域 $\Omega = (0, 1) \times (0, 1)$ ，全部条件为：

$$\begin{cases} f(x, y) = -4\pi^2(2\cos(2\pi x) - 1)\sin(2\pi y) + x^2 \\ g(x, y) = 4\pi^2(2\cos(2\pi y) - 1)\sin(2\pi x) \\ b(x, 0) = -2\pi(1 - \cos(2\pi x)) \\ t(x, 1) = 2\pi(1 - \cos(2\pi x)) \\ l(0, y) = 2\pi(1 - \cos(2\pi y)) \\ r(1, y) = -2\pi(1 - \cos(2\pi y)) \end{cases}$$

真解为：

$$\begin{cases} u(x, y) = (1 - \cos(2\pi x))\sin(2\pi y) \\ v(x, y) = -(1 - \cos(2\pi y))\sin(2\pi x) \\ p(x, y) = \frac{x^3}{3} - \frac{1}{12} \end{cases}$$

对于 $N = 64, 128, 256, 512, 1024$ ，取初值 $P_0(x, y) = x + y$ ，采用 Uzawa 方法求解离散问题 (*)，停机标准为 $\frac{\|r_h\|_2}{\|r_0\|_2} < 10^{-8}$ 。将程序的 CPU 用时、数值解与真解的误差 e_N 、误差收敛阶 $r = \log e_N / \log N$ 列表和作图如下：其中

$$e_N = \frac{1}{N} \sqrt{\sum_{i,j} |U[i, j] - u(x_i, y_j)|^2 + \sum_{i,j} |V[i, j] - v(x_i, y_j)|^2}$$

N	用时 (s)	误差 e_N	收敛阶 r
64	0.0204	0.0015	-1.5635
128	0.1176	3.7363×10^{-4}	-1.6266
256	0.5869	9.3398×10^{-5}	-1.6733
512	2.1819	2.3349×10^{-5}	-1.7096
1024	10.0269	5.8372×10^{-6}	-1.7386
2048	93.4682	1.4593×10^{-6}	-1.7624

将 e_N 作于离散对数图中：

使用 Matlab 计算得到其回归直线的斜率是：

$$k = -2.0008$$

可见，误差的收敛速度为二次的。

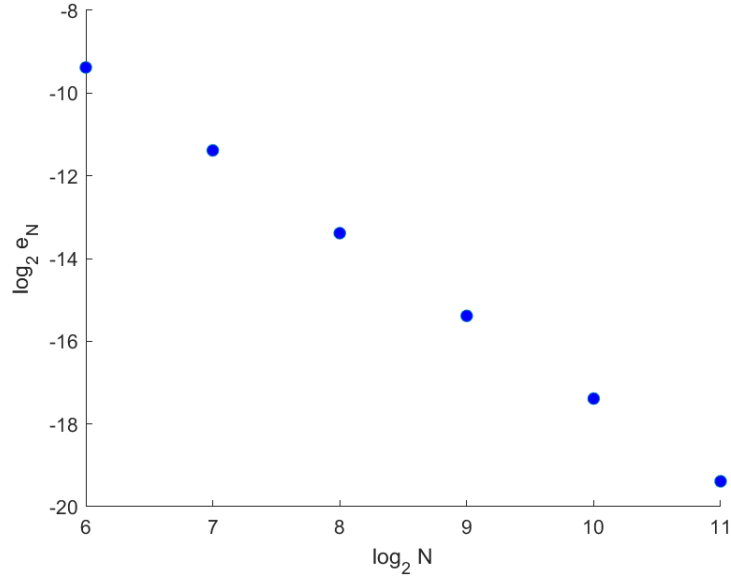


图 2: Caption

4 Inexact Uzawa 方法

4.1 Inexact Uzawa 方法

在 Uzawa 方法中, 第一步精确求解 $AX = F - BP$ 随着 N 的增大, 所需的时间、内存开销都在令人难以接受地上升。为了解决这一问题, 我们采用 Inexact Uzawa 方法: Inexact Uzawa 迭代与 Uzawa 迭代并无太大区别, 除了第一步不再精确求解方程 $AX = H - BP_k$ 的解, 而是选择求出一个近似解 \hat{X}_{k+1} 用于接下来的算法。在本文中, 我们近似求解方程的方法是采用 V-cycle 作为预优子的预优共轭梯度法。

那么我们需要近似求解到何种地步? 可以证明, 对于预先选定的充分小的 $\tau > 0$, 定义

$$\delta_k = A\hat{X}_{k+1} - H + BP_k$$

只要近似解满足

$$\|\delta_k\|_2 \leq \tau \|B^T \hat{X}_k\|$$

那么 Inexact Uzawa 迭代是收敛的。参数 τ 的选取需要在精确度和计算性能开销之间做出合理的抉择, 关于它的详细讨论会在下一小节中介绍。

Algorithm 2 Inexact Uzawa 迭代求解离散形式 (*)

Require: $P_0, k = 0, r_0 = \|H\|$

while $\|r_h\|/\|r_0\| > 10^{-8}$ **do**

 以 V-cycle 预优共轭梯度法近似求解方程 $AX = H - BP_k$ 得到近似解 \hat{X}_{k+1}

 更新压力 $P_{k+1} = P_k + \alpha B^T \hat{X}_{k+1}$

 计算误差 $r_h = H - A\hat{X}_{k+1} - BP$

$k = k + 1$

end while

return $[\hat{X}_k, P_k]$

Algorithm 3 V-cycle 预优共轭梯度法近似求解方程 $Ax = b$ **Require:** x, k_{max} $k = 0; r = b - Ax; \rho = r^T r$ **while** $\|r\| > \max\{10^{-8}\|b\|, \tau\|B^T \hat{U}_k\|\}$ 以及 $k < k_{max}$ **do** $k = k + 1$ 以对称 Gauss-Seidel 迭代作为磨光子, 利用 V-cycle 多重网格方法求解 $Az = r$ **if** $k = 1$ **then** $p = z; \rho = r^T z$ **else** $\tilde{\rho} = \rho; \rho = r^T z; \beta = \rho/\tilde{\rho}; p = z + \beta p$ **end if** $w = Ap; \alpha = \rho/\rho^T w; x = x + \alpha p; r = r - \alpha w$ **end while****return** x

其中, 对称 G-S 迭代指的是对于方程 $Ax = r$, 把 A 分解为 $A = D - L - U$, D 对角, L 严格下三角, U 严格上三角。那么:

$$x_{k+\frac{1}{2}} = (D - L)^{-1}(r + Ux_k)$$

$$x_{k+1} = (D - U)^{-1}(r + Lx_{k+\frac{1}{2}})$$

由于 A 对称正定, 所以它是收敛的。

4.2 Inexact Uzawa 方法的收敛性讨论以及参数 τ 的选取

迭代方程:

$$A\hat{X}_{k+1} + BP_k = H + \delta_k$$

$$P_{k+1} = P_k + \alpha B^T \hat{X}_{k+1}$$

设 X, P 是方程组的真解, 即 $AX + BP = H, B^T X = 0$, 定义误差

$$e_k = X - \hat{X}_k, \epsilon_k = P - P_k$$

则有:

$$A(\hat{X}_k - \hat{X}_{k+1}) = B(P_k - P_{k-1}) + \delta_{k-1} - \delta_k = \alpha BB^T \hat{X}_k + \delta_{k-1} - \delta_k$$

$$\Rightarrow e_{k+1} - e_k = \alpha A^{-1} BB^T \hat{X}_k + A^{-1} \delta_{k-1} - A^{-1} \delta_k$$

$$\Rightarrow B^T(e_{k+1} - e_k) = \alpha B^T A^{-1} BB^T \hat{X}_k + B^T A^{-1}(\delta_{k-1} - \delta_k)$$

由于 $B^T X = 0$, 所以 $B^T A^{-1} BB^T X = 0$, 加到等号右边的第一项, 得到:

$$B^T(e_{k+1} - e_k) = -\alpha SB^T e_k + B^T A^{-1}(\delta_{k-1} - \delta_k)$$

亦即

$$B^T e_{k+1} = (I - \alpha S) B^T e_k + B^T A^{-1}(\delta_{k-1} - \delta_k)$$

再运用预设条件 $\|\delta_k\| \leq \tau \|B^T \hat{X}_k\| = \tau \|B^T e_k\|$ 就有了:

$$\begin{aligned} \|B^T e_{k+1}\| &\leq \|I - \alpha S\| \|B^T e_k\| + \|B^T A^{-1}\| (\|\delta_{k-1}\| + \|\delta_k\|) \\ &\leq \|I - \alpha S\| \|B^T e_k\| + \tau \|B^T A^{-1}\| (\|B^T e_{k-1}\| + \|B^T e_k\|) \end{aligned}$$

记 $\rho = \|I - \alpha S\|$, $\eta = \tau \|B^T A^{-1}\|$, 就有了 $\|B^T e_k\| \leq \beta_k$, 其中

$$\beta_{k+1} = (\rho + \eta)\beta_k + \eta\beta_{k-1}, \quad \forall k \geq 2$$

特征根是

$$r_{1,2} = \frac{\rho + \eta}{2} (1 \pm \sqrt{1 + 4\eta/(\rho + \eta)^2})$$

所以就有了

$$\beta_k = C_1 r_1^k + C_2 r_2^k$$

由于 $|r_1| > |r_2|$, 所以得到了如下的关于收敛速度的定理:

定理 4.1. 设 $\rho = \|I - \alpha S\|$, $\eta = \tau \|B^T A^{-1}\|$, 定义

$$\hat{r} = \frac{\rho + \eta}{2} (1 + \sqrt{1 + 4\eta/(\rho + \eta)^2})$$

那么就有

$$\|B^T e_k\| \leq C \hat{r}^k$$

其中 C 是一个常数。

接下来, 代入 $\rho = 1 - \alpha$, $0 < \alpha < 1$ 和 $\hat{r} < 1$, 我们就得到了:

$$\hat{r} < 1 \Leftrightarrow 2\eta + \rho < 1 \Leftrightarrow \eta < \frac{\alpha}{2} \Leftrightarrow \tau < \frac{\alpha}{2\|B^T A^{-1}\|}$$

引理 4.1. 我们有以下等式成立:

$$\|B^T A^{-1}\| = \sqrt{\frac{1}{2N^2(1 - \cos \frac{\pi}{N})}} \gtrsim \frac{1}{\pi}$$

证明. 既然 B 的像空间是满的, 所以我们有

$$\|B^T A^{-1}\|^2 = \max_{v \neq 0} \frac{v^T A^{-1} B B^T A^{-1} v}{v^T v} = \max_{Bw \neq 0} \frac{w^T B^T A^{-1} B B^T A^{-1} B w}{w^T B^T B w} = \max_{Bw \neq 0} \frac{\|Sw\|^2}{w^T B^T B w}$$

也就是令 w 取遍所有在 B 的核中投影为 0 的向量, 此时根据上一节的结果有 $\|Sw\| = \|w\|$, 并且

$$\|B^T A^{-1}\|^2 = \max_w \frac{w^T w}{w^T B^T B w} = \lambda_{\min, \neq 0}(B^T B)^{-1}$$

再运用一开始的结果, 根据 $\lambda(B^T B) = \{\lambda_i + \lambda_j\}_{i,j}$, $\lambda_i \in \lambda(Q^T Q) = \{2(1 - \cos \frac{k\pi}{N})\}_{k=0:N-1}$, 就得到了:

$$\|B^T A^{-1}\| = \sqrt{\frac{1}{2N^2(1 - \cos \frac{\pi}{N})}} \gtrsim \frac{1}{\pi}$$

□

上述运用 matlab 验算也成立。这样, 就得到了 τ 的选取的定理:

定理 4.2. 使得 Inexact Uzawa 收敛的一个合适的参数组是

$$0 < \alpha < 1, \quad 0 < \tau < \frac{\alpha\pi}{2}$$

譬如, $\alpha = 0.5$, $\tau = 0.7$ 。

4.3 数值结果

数值算例：区域 $\Omega = (0, 1) \times (0, 1)$ ，全部条件为：

$$\begin{cases} f(x, y) = -4\pi^2(2\cos(2\pi x) - 1)\sin(2\pi y) + x^2 \\ g(x, y) = 4\pi^2(2\cos(2\pi y) - 1)\sin(2\pi x) \\ b(x, 0) = -2\pi(1 - \cos(2\pi x)) \\ t(x, 1) = 2\pi(1 - \cos(2\pi x)) \\ l(0, y) = 2\pi(1 - \cos(2\pi y)) \\ r(1, y) = -2\pi(1 - \cos(2\pi y)) \end{cases}$$

真解为：

$$\begin{cases} u(x, y) = (1 - \cos(2\pi x))\sin(2\pi y) \\ v(x, y) = -(1 - \cos(2\pi y))\sin(2\pi x) \\ p(x, y) = \frac{x^3}{3} - \frac{1}{12} \end{cases}$$

对于 $N = 64, 128, 256, 512, 1024$ ，取初值 $P_0(x, y) = x + y$ ，采用 Inexact Uzawa 方法求解离散问题 (*)，停机标准为 $\frac{\|r_h\|_2}{\|r_0\|_2} < 10^{-8}$ 。将程序的 CPU 用时、数值解与真解的误差 e_N 、误差收敛阶 $r = \log e_N / \log N$ 列表和作图如下：其中

$$e_N = \frac{1}{N} \sqrt{\sum_{i,j} |U[i, j] - u(x_i, y_j)|^2 + \sum_{i,j} |V[i, j] - v(x_i, y_j)|^2}$$

N	用时 (s)	误差 e_N	收敛阶 r
64	0.0204	0.0015	-1.5635
128	0.1324	3.7363×10^{-4}	-1.6266
256	0.5778	9.3398×10^{-5}	-1.6733
512	1.2351	2.3349×10^{-5}	-1.7096
1024	4.2679	5.8372×10^{-6}	-1.7386
2048	10.1457	1.4593×10^{-6}	-1.7624

将 e_N 作于离散对数图中：

使用 Matlab 计算得到其回归直线的斜率是：

$$k = -1.9987$$

可见，误差的收敛速度为二次的。

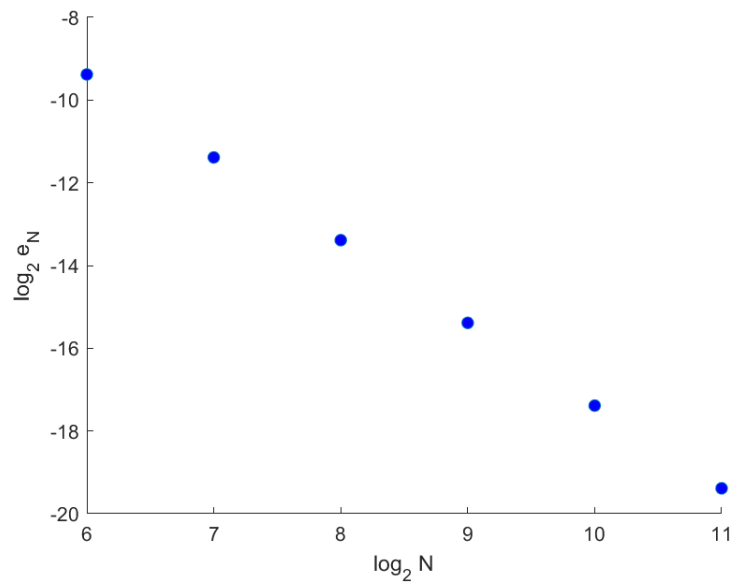


图 3: Caption