
数值代数大作业

高以恒

2200010851@stu.pku.edu.cn

cpu: M1 Pro

1 问题描述

考虑 Stokes 方程

$$\begin{cases} -\Delta \vec{u} + \nabla p = \vec{F}, & (x, y) \in (0, 1) \times (0, 1) \\ \operatorname{div} \vec{u} = 0, & (x, y) \in (0, 1) \times (0, 1) \end{cases} \quad (1)$$

边界条件为

$$\begin{aligned} \frac{\partial u}{\partial \vec{n}} &= b, & y = 0, & \quad \frac{\partial u}{\partial \vec{n}} = t, & y = 1, \\ \frac{\partial v}{\partial \vec{n}} &= l, & x = 0, & \quad \frac{\partial v}{\partial \vec{n}} = r, & x = 1, \\ u &= 0, & x = 0, 1, & \quad v = 0, & y = 0, 1 \end{aligned}$$

其中 $\vec{u} = (u, v)$ 是速度场, p 是压力场, $\vec{F} = (f, g)$ 是外力场, \vec{n} 是单位外法向量。利用交错网格上的 MAC 格式离散方程 (1), 可得如下线性方程组

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} \vec{F} \\ 0 \end{pmatrix} \quad (2)$$

具体形式见附录。

在该例中, 外力为

$$\begin{aligned} f(x, y) &= -4\pi^2(2\cos(2\pi x) - 1)\sin(2\pi y) + x^2, \\ g(x, y) &= 4\pi^2(2\cos(2\pi y) - 1)\sin(2\pi x). \end{aligned}$$

Stokes 方程 (1) 的解析解为

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

其中压力 p 的取值可能有任意常数的平移。

2 Task1

分别取 $N = 64, 128, 256, 512, 1024, 2048$, 以 DGS 为磨光子, 用基于 V-cycle 的多重网格方法求解离散问题 (2), 停机标准为 $\|r_h\|_2/\|r_0\|_2 \leq 10^{-8}$ 。对不同的 $v1, v2, L$, 比较 V-cycle 的次数和 CPU 时间, 并计算误差

$$e_N = h \left(\sum_{j=1}^N \sum_{i=1}^N \left(u_{i,j-\frac{1}{2}} - u(x_i, y_{j-\frac{1}{2}}) \right)^2 + \sum_{j=1}^{N-1} \sum_{i=1}^{N-1} \left(v_{i-\frac{1}{2},j} - v(x_{i-\frac{1}{2}}, y_j) \right)^2 \right)^{\frac{1}{2}}$$

2.1 DGS 迭代法和 V-cycle 多重网格方法

给定初始值 $U_0, P_0 = p_0, f, g$, 迭代格式如下: 1. 用 Gauss-Seidel 迭代法求解 $AU^{(k+1)} = F - B^T P^{(k)}$, 得到 $U^{(k+1/2)}$; 2. 更新内部速度和压力:

$$\begin{aligned} U^{(k+1)} &= U^{(k+1/2)} + B(B^T B)^{-1}(g - BU^{(k+1/2)}), \\ P^{(k+1)} &= P^{(k)} - (g - BU^{(k+1/2)}). \end{aligned} \quad (3)$$

Algorithm 1 V-cycle 多重网格方法

Require: $U_0, P_0, f, g, v1, v2, N, L$

Ensure: 解 U_h 和 P_h

```

1: 初始化残差  $r_h = f - A_h U_h$ 
2: while  $\|r_h\|_2/\|r_0\|_2 > 10^{-8}$  do
3:   对  $U_h$  和  $P_h$  进行  $v1$  次 DGS 更新
4:   if  $L == N$  then
5:     return  $U_h, P_h$ 
6:   else
7:     将残差  $r_h$  限制到粗网格:  $r_{2h} = R(r_h)$ 
8:     初始化粗网格解:  $U_{2h} = 0, P_{2h} = 0$ 
9:     递归调用 V-cycle( $U_{2h}, P_{2h}, r_{2h}, f_{2h}, g_{2h}, v1, v2, 2h, L+1$ )
10:    将粗网格解插值回细网格:  $U_h = U_h + I(U_{2h}), P_h = P_h + I(P_{2h})$ 
11:   end if
12:   对  $U_h$  和  $P_h$  进行  $v2$  次 DGS 更新
13:   更新残差:  $r_h = f - A_h U_h$ 
14: end while

```

注: 在计算粗网格上的 A_{2h}, B_{2h} 时, 按照课件上的方法, A_{2h}, B_{2h} 的形式如下:

$$\begin{pmatrix} A_{2h} & B_{2h} \\ B_{2h}^T & 0 \end{pmatrix} = I_h^{2h} \begin{pmatrix} A_h & B_h \\ B_h^T & 0 \end{pmatrix} I_{2h}^h$$

其中 I_h^{2h} 是限制算子, I_{2h}^h 是插值算子。但是这样做的计算代价比较大: 一方面要进行矩阵乘法, 另一方面得到的 A_{2h}, B_{2h} 的稀疏性比较差。因此, 在具体实现中我们直接

用粗网格上的 MAC 格式离散得到的稀疏矩阵去近似, 实际效果也是比较好的。如果想使用前一种方法, 我也提供了选项, 具体参见代码。

考虑到在粗网格上进行迭代的时间复杂度较低, 在实际实现中我让 $v2 = v_{20} + \log_2(N_0/N)$, 其中 v_{20} 是在初始网格上进行迭代的次数, N_0 是初始网格的大小。

2.2 并行加速

对 DGS, 两部分更新都可以并行化, 这里使用 OpenMP 对细网格 ($N > 8$) 进行并行加速。对 Gaussian-Seidel 迭代法, 使用红黑着色法, 将网格分为红色和黑色两部分, 分别更新。对压力和速度的更新, 可以简单的分成 8 个部分, 分别更新。

2.3 数值结果

对不同的迭代次数 $v1, v2$ 和底层网格大小 N , 得到的误差误差很小, 因此只给出 $v1 = 3, v2 = 1, N/L = 2$ 时的结果。

N	64	128	256	512	1024	2048	4096
误差	1.4951e-3	3.7363e-4	9.3399e-5	2.3349e-5	5.8372e-6	1.4593e-6	3.6480e-7

对不同迭代次数和底层网格大小的运行时间和 V-cycle 迭代次数比较:

表 1: $v1 = 1, v2 = 3, N/L = 2$

N	64	128	256	512	1024	2048
迭代次数	6	6	6	6	6	6
CPU 时间	0.0378	0.0396	0.0845	0.2574	0.8284	3.4668

注: 时间复杂度是 $O(N^2)$ 的, 对于 4096 的网格, 由于 16g 内存限制, 部分数据需要存储在硬盘上, 因此时间较长 (30s)。实测在 64g 内存的机器上, 8192 的网格也可以在 80s 左右完成。

表 2: $v1 = 1, v2 = 3, N/L = 4$

N	64	128	256	512	1024	2048
迭代次数	6	6	6	6	6	6
CPU 时间	0.0345	0.0581	0.1125	0.3260	0.9912	3.6560

表 3: $v1 = 1, v2 = 1, N/L = 2$

N	64	128	256	512	1024	2048
迭代次数	9	9	9	9	9	9
CPU 时间	0.0399	0.0407	0.0945	0.3062	0.9557	4.1706

表 4: $v1 = 1, v2 = 1, N/L = 4$

N	64	128	256	512	1024	2048
迭代次数	9	9	9	9	9	9
CPU 时间	0.0246	0.0429	0.0975	0.2850	1.0103	4.2779

表 5: $v1 = 2, v2 = 2, N/L = 2$

N	64	128	256	512	1024	2048
迭代次数	6	6	6	6	6	6
CPU 时间	0.0276	0.0375	0.0872	0.2694	0.8494	3.5884

表 6: $v1 = 2, v2 = 2, N/L = 4$

N	64	128	256	512	1024	2048
迭代次数	6	6	6	6	6	6
CPU 时间	0.0345	0.0581	0.1125	0.3260	0.9912	3.6560

表 7: $v1 = 3, v2 = 3, N/L = 2$

N	64	128	256	512	1024	2048
迭代次数	6	6	6	6	6	6
CPU 时间	0.0374	0.0490	0.1167	0.3090	0.9741	4.0411

表 8: $v1 = 3, v2 = 3, N/L = 4$

N	64	128	256	512	1024	2048
迭代次数	6	6	6	6	6	6
CPU 时间	0.0465	0.0473	0.1063	0.3131	1.0737	4.3676

更多结果见附录 5.3. 可以看到, $v1, v2$ 的取值不需要太大, N/L 的取值则为 2 时效果最好, 这也符合 V-cycle 多重网格方法的特点. 理论上, 算法的时间复杂度为 $O(N^2)$, 误差阶为 $O(h^2)$, 数值结果也符合这一点.

3 Task2

分别取 $N = 64, 128, 256, 512$, 以 Uzawa Iteration Method 求解离散问题 (2), 停机标准为 $\|r_h\|_2/\|r_0\|_2 \leq 10^{-8}$ 。计算误差 e_N 。

3.1 Uzawa 迭代法

Algorithm 2 Uzawa 迭代法

Require: U_0, P_0, f, N, α **Ensure:** 解 U_h 和 P_h

- 1: 初始化残差 $r_h = f - A_h U_h$
 - 2: **while** $\|r_h\|_2/\|r_0\|_2 > 10^{-8}$ **do**
 - 3: 利用 CG 方法求解 $AU_{k+1} = f - B^T P_k$
 - 4: 更新压力: $P_{k+1} = P_k + \alpha B U_{k+1}$
 - 5: 更新残差: $r_h = f - A_h U_h$
 - 6: **end while**
-

关于 α 的选取, 分析知 $B^T A^{-1} B$ 的特征值为 0 和 1, 因此 α 的最佳取值为 1. (见附录 5.2)

3.2 数值结果

对 CG 算法的精确度要求为 10^{-9} , 对 Uzawa 迭代法的精确度要求为 10^{-8} 。

N	64	128	256	512
迭代次数	2	2	2	2
CPU 时间	0.0401	0.0958	0.8966	5.9550
误差	1.4951e-3	3.7363e-4	9.3399e-5	2.3349e-5

4 Task3

分别取 $N = 64, 128, 256, 512, 1024, 2048$, 以 Inexact Uzawa Iteration Method 为迭代法求解离散问题 (2), 停机标准为 $\|r_h\|_2/\|r_0\|_2 \leq 10^{-8}$ 。其中以 V-cycle 多重网格方法为预条件子, 利用共轭梯度法求解每一步的子问题 $AU_{k+1} = F - B^T P_k$, 对不同的 $\alpha, \tau, \nu_1, \nu_2, L$, 比较外循环的迭代次数和 CPU 时间, 并计算误差。

4.1 Inexact Uzawa 迭代法和 V-cycle-PCG 算法

定义 $\delta_k = A\hat{U}_{k+1} - f + B^T P_k$, 若 $\|\delta_k\|_2 \leq \tau \|B^T \hat{U}_{k+1}\|_2$, 且 τ 充分小, 则上述方法收敛。

Algorithm 3 Inexact Uzawa 迭代法

Require: $U_0, P_0, f, g, N, \alpha, \tau, \nu_1, \nu_2, L$ **Ensure:** 解 U_h 和 P_h

- 1: 初始化残差 $r_h = f - A_h U_h$
 - 2: **while** $\|r_h\|_2 / \|r_0\|_2 > 10^{-8}$ **do**
 - 3: 利用 V-cycle-PCG 方法求解 $AU_{k+1} = f - B^T P_k$, 得到近似解 \hat{U}_{k+1} ,
 - 4: 更新压力: $P_{k+1} = P_k + \alpha B \hat{U}_{k+1}$
 - 5: 更新残差: $r_h = f - A \hat{U}_{k+1}$
 - 6: **end while**
-

Algorithm 4 V-cycle-PCG 算法

Require: : x

- 1: $k = 0, r = b - Ax, \rho = r^T r$
 - 2: **while** $(\|r\| > \max(\epsilon \|b\|, \tau \|B^T \hat{U}_k\|)) \text{ and } k < K_m \text{ and } \alpha < K_m \alpha$ **do**
 - 3: $k = k + 1$
 - 4: 以 GS 迭代法为磨光子, 利用 V-cycle 多重网格方法求解 $Az = r$
 - 5: **if** $k = 1$ **then** $p = z; \rho = z^T r$
 - 6: **else** $\rho_{old} = \rho; \rho = z^T r; \beta = \rho / \rho_{old}; p = z + \beta p$
 - 7: **end if**
 - 8: $w = Ap; \alpha = \rho / (p^T w); x = x + \alpha p; r = r - \alpha w$
 - 9: **end while**
-

该部分我尝试了对称 GS 迭代和红黑 GS 迭代, 效果差不多, 因此并行实现。V-cycle 多重网格算法停机标准为 $\|r_h\|_2 / \|r_0\|_2 \leq 10^{-3} \text{ and } k \leq 3$

4.2 数值结果

误差:

N	64	128	256	512	1024	2048
误差	1.4951e-3	3.7363e-4	9.3399e-5	2.3349e-5	5.8372e-6	1.4593e-6

不同参数对运行时间和 V-cycle 迭代次数的影响见1。这里分别固定 $\alpha = 1, \tau = 10^{-3}$, ν_1, ν_2 , 改变另一组参数 ($N/L = 2, N = 2048$)

可以看到, 参数对改变对误差影响不大, 但是对迭代次数和时间影响较大。固定 ν_1, ν_2 , 改变 α, τ , 可以看到 $\alpha = 1$ 时效果最好, τ 取值对结果影响不大。

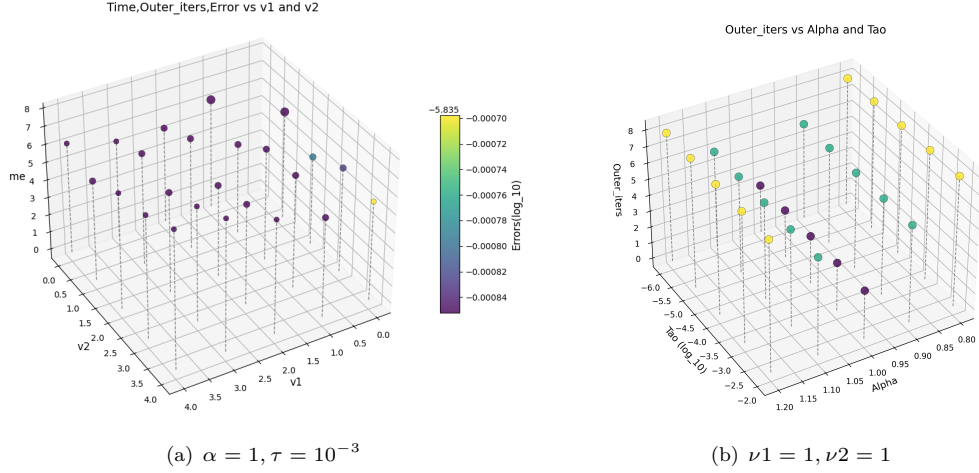


图 1: 注: 左图是 cpu 时间关于 v_1, v_2 的图, 右图是外循环迭代次数关于 α, τ 的图。左图点的大小与迭代次数成正比, 颜色表示误差大小。

最后以下给出几组组参数下的结果, 该算法时间复杂度为 $O(N^2)$, 误差阶为 $O(h^2)$, 数值结果也符合这一点。更多结果见附录。

$\alpha = 1, \tau = 10^{-3}, \nu_1 = 1, \nu_2 = 1, N/L = 2$:

N	64	128	256	512	1024	2048
CPU 时间	0.0698	0.0579	0.1445	0.4214	1.2661	5.0848
Inexact Uzawa 迭代次数	3	4	4	4	3	3
PCG 迭代次数	3 3 2	3 2 1 1	3 2 1 1	3 2 1 1	3 2 1	2 3 1

$\alpha = 1, \tau = 10^{-3}, \nu_1 = 1, \nu_2 = 1, N/L = 4$:

N	64	128	256	512	1024	2048
CPU 时间	0.1061	0.1597	0.3786	0.9168	2.7406	9.4641
Inexact Uzawa 迭代次数	4	4	4	4	4	4
PCG 迭代次数	5 15 2 1	4 14 2 1	4 12 2 1	3 11 2 1	3 9 2 1	3 8 2 1

$\alpha = 1, \tau = 10^{-3}, \nu_1 = 2, \nu_2 = 2, N/L = 2$:

N	64	128	256	512	1024	2048
CPU 时间	0.0657	0.0730	0.1736	0.4687	1.5007	4.3087
Inexact Uzawa 迭代次数	3	3	3	3	3	3
PCG 迭代次数	3 3 1	3 3 1	3 3 1	3 3 1	3 3 1	2 2 1

$\alpha = 1, \tau = 10^{-3}, \nu_1 = 2, \nu_2 = 2, N/L = 4$:

N	64	128	256	512	1024	2048
CPU 时间	0.1016	0.1409	0.2515	0.8003	1.9337	7.8004
Inexact Uzawa 迭代次数	4	4	4	4	3	3
PCG 迭代次数	3 8 2 1	3 6 2 1	3 5 1 1	3 3 2 1	3 4 1	3 3 1

5 Appendix

5.1 交错网格上的 MAC 格式

方程组 (2) 的系数矩阵 A, B 的具体形式如下:

其中 $A = \begin{pmatrix} A_u & \\ & A_v \end{pmatrix}, B = \begin{pmatrix} B_u \\ B_v \end{pmatrix}, A_u, A_v \in \mathbb{R}^{N(N-1) \times N(N-1)},$

$$\begin{aligned}
A_u &= \begin{pmatrix} A_1 & -\frac{1}{h^2}I & & & \\ -\frac{1}{h^2}I & A_2 & -\frac{1}{h^2}I & & \\ & -\frac{1}{h^2}I & A_2 & -\frac{1}{h^2}I & \\ & & \ddots & \ddots & \ddots \\ & & & -\frac{1}{h^2}I & A_2 & -\frac{1}{h^2}I \\ & & & & -\frac{1}{h^2}I & A_1 \end{pmatrix}, A_v = \begin{pmatrix} A_3 & -\frac{1}{h^2}I & & & \\ -\frac{1}{h^2}I & A_3 & -\frac{1}{h^2}I & & \\ & -\frac{1}{h^2}I & A_3 & -\frac{1}{h^2}I & \\ & & \ddots & \ddots & \ddots \\ & & & -\frac{1}{h^2}I & A_3 & -\frac{1}{h^2}I \\ & & & & -\frac{1}{h^2}I & A_3 \end{pmatrix} \\
A_1 &= \begin{pmatrix} \frac{3}{h^2} & -\frac{1}{h^2} & & & \\ -\frac{1}{h^2} & \frac{3}{h^2} & -\frac{1}{h^2} & & \\ & -\frac{1}{h^2} & \frac{3}{h^2} & -\frac{1}{h^2} & \\ & & \ddots & \ddots & \ddots \\ & & & -\frac{1}{h^2} & \frac{3}{h^2} & -\frac{1}{h^2} \\ & & & & -\frac{1}{h^2} & \frac{3}{h^2} & -\frac{1}{h^2} \\ & & & & & -\frac{1}{h^2} & \frac{3}{h^2} \end{pmatrix}, A_2 = \begin{pmatrix} \frac{4}{h^2} & -\frac{1}{h^2} & & & \\ -\frac{1}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} & & \\ & -\frac{1}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} & \\ & & \ddots & \ddots & \ddots \\ & & & -\frac{1}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} \\ & & & & -\frac{1}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} \end{pmatrix}, \\
A_3 &= \begin{pmatrix} \frac{3}{h^2} & -\frac{1}{h^2} & & & \\ -\frac{1}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} & & \\ & -\frac{1}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} & \\ & & \ddots & \ddots & \ddots \\ & & & -\frac{1}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} \\ & & & & -\frac{1}{h^2} & \frac{3}{h^2} \end{pmatrix}, B_u = \begin{pmatrix} H & & & \\ & H & & \\ & & \ddots & \\ & & & H \end{pmatrix}, \\
B_v &= \begin{pmatrix} -\frac{1}{h}I_{N \times N} & \frac{1}{h}I_{N \times N} & & & \\ & -\frac{1}{h}I_{N \times N} & \frac{1}{h}I_{N \times N} & & \\ & & \ddots & \ddots & \\ & & & -\frac{1}{h}I_{N \times N} & \frac{1}{h}I_{N \times N} \end{pmatrix}, H = \begin{pmatrix} -\frac{1}{h} & \frac{1}{h} & & & \\ & -\frac{1}{h} & \frac{1}{h} & & \\ & & \ddots & \ddots & \\ & & & -\frac{1}{h} & \frac{1}{h} \end{pmatrix}.
\end{aligned}$$

5.2 $B^T A^{-1} B$ 的特征值

$$B_{N-1} = \begin{pmatrix} 2 & -1 & 0 & \cdots & \\ -1 & 2 & -1 & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \\ \vdots & \cdots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}_{(N-1) \times (N-1)}, \quad R_N = \begin{pmatrix} 1 & -1 & 0 & \cdots & \\ -1 & 2 & -1 & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \\ \vdots & \cdots & -1 & 2 & -1 \\ \vdots & \cdots & \cdots & -1 & 1 \end{pmatrix}_{N \times N}$$

直接计算可得:

$$A = \begin{pmatrix} A_u & 0 \\ 0 & A_v \end{pmatrix}, \quad A_u = B_{N-1} \oplus R_N, \quad A_v = R_N \oplus B_{N-1}, \quad B^T B = R_N \oplus R_N$$

$$B = \begin{pmatrix} I \otimes H \\ H \otimes I \end{pmatrix}, \quad B_{N-1} = HH^T, \quad R_N = H^T H$$

其中 \oplus 为 Kronecker 直和, $A \oplus B = I \otimes A + B \otimes I$.

接下来通过 R_N, B_{N-1} 求 H 的 SVD 分解, 令

$$\lambda_k = 2 \sin\left(\frac{k\pi}{2N}\right) = \sqrt{2(1 - \cos(\frac{k\pi}{N}))}, \quad k = 1, 2, \dots, N-1$$

则容易验证 R_N 的特征值和特征向量为

$$\lambda_1^2, \dots, \lambda_{N-1}^2, 0$$

$$v_k = \frac{(\cos(\frac{(j-1/2)k\pi}{N}))_{j=1:N}^T}{\|\cdot\|}, \quad k = 1 : N-1, \quad v_N = (1, 1, \dots, 1)^T$$

其中 $\|\cdot\|$ 表示分子上的向量的模长, 同理可得 B_{N-1} 的特征值和特征向量为

$$\lambda_1^2, \dots, \lambda_{N-1}^2$$

$$u_k = \frac{(\cos(\frac{jk\pi}{N}))_{j=1:N-1}^T}{\|\cdot\|}, \quad k = 1 : N-1$$

故 $H = V \Sigma U^T$, 其中 $V = (v_1, \dots, v_{N-1}, v_N), U = (u_1, \dots, u_{N-1}), \Sigma = \text{diag}(\lambda_1, \dots, \lambda_{N-1}, 0)$. 考虑压力空间的一组基 $v_i \otimes v_j, i, j = 1, 2, \dots, N$, 若 $i \neq N, j \neq$

N ，则有

$$B^T A^{-1} B(v_i \otimes v_j) = B^T A^{-1} \begin{pmatrix} \lambda_j v_i \otimes u_j \\ \lambda_i u_i \otimes v_j \end{pmatrix} \quad (4)$$

$$= B^T \begin{pmatrix} \frac{\lambda_j}{\lambda_i^2 + \lambda_j^2} v_i \otimes u_j \\ \frac{\lambda_i}{\lambda_i^2 + \lambda_j^2} u_i \otimes v_j \end{pmatrix} \quad (5)$$

$$= \frac{\lambda_j^2}{\lambda_i^2 + \lambda_j^2} v_i \otimes v_j + \frac{\lambda_i^2}{\lambda_i^2 + \lambda_j^2} v_i \otimes v_j \quad (6)$$

$$= v_i \otimes v_j \quad (7)$$

类似的可以得到 i, j 有且只有一个等于 N 时, $B^T A^{-1} B(v_i \otimes v_j) = v_i \otimes v_j$, i, j 都等于 N 时, $B^T A^{-1} B(v_i \otimes v_j) = 0$ 。故 $B^T A^{-1} B$ 的特征值为 1 和 0, 特征向量为 $v_i \otimes v_j$, $i, j = 1, 2, \dots, N$ 。

5.3 DGS 迭代法和 V-cycle 多重网格方法数值结果

表 9: $v1 = 4, v2 = 4, N/L = 2$

N	64	128	256	512	1024	2048
迭代次数	5	5	5	5	5	5
CPU 时间	0.0569	0.0553	0.1227	0.3015	1.1306	4.2555

表 10: $v1 = 4, v2 = 4, N/L = 4$

N	64	128	256	512	1024	2048
迭代次数	5	5	5	5	5	5
CPU 时间	0.0341	0.0464	0.1047	0.3519	1.0612	4.5749

表 11: $v1 = 6, v2 = 6, N/L = 2$

N	64	128	256	512	1024	2048
迭代次数	5	5	5	5	5	5
CPU 时间	0.0548	0.0667	0.1355	0.4067	1.1734	4.8236

表 12: $v1 = 6, v2 = 6, N/L = 4$

N	64	128	256	512	1024	2048
迭代次数	5	5	5	5	5	5
CPU 时间	0.0625	0.0735	0.1410	0.3848	1.3356	5.7106

5.4 Inexact Uzawa 迭代法的参数选取

$\alpha = 1, \tau = 10^{-3}, \nu1 = 3, \nu2 = 3, N/L = 2$:

N	64	128	256	512	1024	2048
CPU 时间	0.0653	0.0810	0.1847	0.5133	1.6055	5.7464
Inexact Uzawa 迭代次数	2	2	2	2	2	2
PCG 迭代次数	3 3	3 3	3 3	3 3	3 3	3 2

$\alpha = 1, \tau = 10^{-3}, \nu_1 = 3, \nu_2 = 3, N/L = 4$:

N	64	128	256	512	1024	2048
CPU 时间	0.0702	0.0807	0.1848	0.4922	1.3745	5.7537
Inexact Uzawa 迭代次数	3	2	2	2	2	2
PCG 迭代次数	3 3 1	3 3	3 3	3 3	3 2	3 2

$\alpha = 1, \tau = 10^{-3}, \nu_1 = 4, \nu_2 = 4, N/L = 2$:

N	64	128	256	512	1024	2048
CPU 时间	0.0620	0.1023	0.2237	0.6076	1.6756	7.4759
Inexact Uzawa 迭代次数	2	2	2	2	2	2
PCG 迭代次数	3 3	3 3	3 3	3 3	3 2	3 2

$\alpha = 1, \tau = 10^{-3}, \nu_1 = 3, \nu_2 = 3, N/L = 4$:

N	64	128	256	512	1024	2048
CPU 时间	0.0836	0.0977	0.2279	0.5739	1.8969	8.2079
Inexact Uzawa 迭代次数	2	2	2	2	2	2
PCG 迭代次数	3 3 1	3 3	3 3	3 3	3 3	3 3

Hu [2024]

References

Jun Hu. V-cycle, dec 2024.