# 数值代数大作业：Stokes 方程的交错网格离散与迭代求解

**姓名：** 樊泽羲
**学号：** 2200010816
**日期：** 2026年1月1日

## 1. 问题背景与离散格式

### 1.1 Stokes 方程

考虑定义在单位正方形区域 $\Omega = (0,1) \times (0,1)$ 上的二维 Stokes 方程：

$$\begin{cases} -\Delta \vec{u} + \nabla p = \vec{F}, & (x,y) \in \Omega \\ \nabla \cdot \vec{u} = 0, & (x,y) \in \Omega \end{cases} \tag{1}$$

其中 $\vec{u} = (u,v)^T$ 为速度场，$p$ 为压力场，$\vec{F} = (f,g)^T$ 为外力项。边界条件为混合 Dirichlet-Neumann 边界条件，具体如下：

$$\begin{aligned} &u = 0, \quad x = 0,1; \quad v = 0, \quad y = 0,1 \\ &\frac{\partial u}{\partial y} = b|_{y=0}, \quad \frac{\partial u}{\partial y} = a|_{y=1} \\ &\frac{\partial v}{\partial x} = l|_{x=0}, \quad \frac{\partial v}{\partial x} = r|_{x=1} \end{aligned} \tag{2}$$

其中 $\vec{u} = (u,v)$ 为速度，$p$ 为压力，$\vec{F} = (f,g)$ 为外力，$\vec{n}$ 为外法向方向. 在区域 $\Omega = (0,1) \times (0,1)$ 上，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 方程的真解为

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

### 1.2 交错网格离散 (MAC Scheme)

采用交错网格（Staggered Grid）进行有限差分离散，以克服棋盘格压力模态不稳定性。将区域 $\Omega$ 划分为 $N \times N$ 的网格，步长 $h = 1/N$。变量定义位置如下：

- **压力** $p_{i,j}$：定义在单元中心 $(x_{i-1/2}, y_{j-1/2})$。
- **速度** $u_{i,j}$：定义在单元垂直边中点 $(x_i, y_{j-1/2})$。
- **速度** $v_{i,j}$：定义在单元水平边中点 $(x_{i-1/2}, y_j)$。

利用中心差分格式，动量方程在内部节点 $(i,j)$ 处的离散形式为：

$$\begin{aligned} -\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} - \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} + \frac{p_{i+1,j} - p_{i,j}}{h} &= f_{i,j} \\ -\frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{h^2} - \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{h^2} + \frac{p_{i,j+1} - p_{i,j}}{h} &= g_{i,j} \end{aligned} \tag{3}$$

连续性方程（不可压条件）在单元中心 $(i,j)$ 处的离散形式为：

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

对于 Neumann 边界条件，采用影子节点（Ghost Point）法处理。例如在 $y=0$ 处，$\frac{\partial u}{\partial y}=b \Rightarrow \frac{u_{i,1}-u_{i,0}}{h}=b$，由此消去离散方程中的影子变量 $u_{i,0}$。

## 1.3 线性代数方程组

离散后的系统可写为如下鞍点问题：

$$\mathcal{A}\mathbf{x}=\mathbf{b} \quad \Rightarrow \quad \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix}\begin{pmatrix} U \\ P \end{pmatrix}=\begin{pmatrix} F \\ 0 \end{pmatrix} \tag{5}$$

其中 $U$ 包含所有内部速度未知量，$P$ 包含所有压力未知量。

- $A=\mathrm{diag}(A_u,A_v)$ 为对称正定（SPD）的离散拉普拉斯算子。
- $B$ 为离散梯度算子，$B^T$ 为离散负散度算子。
- 由于压力解相差一个常数，需施加约束 $\sum_{i,j}p_{i,j}=0$ 以保证解的唯一性。

误差度量采用离散 $L^2$ 范数：

$$e_N=\left[h^2\left(\sum_{i,j}|u_{i,j}-u_{exact}(x_i,y_j)|^2+\sum_{i,j}|v_{i,j}-v_{exact}(x_i,y_j)|^2\right)\right]^{1/2} \tag{6}$$

---

# 2. 问题一：基于 V-cycle 的多重网格方法

## 2.1 算法原理：DGS 磨光子

对于耦合系统，我们采用 **分布式 Gauss-Seidel (Distributive Gauss-Seidel, DGS)** 作为磨光子。DGS 的核心思想是在保持残量平滑的同时，解耦动量方程与连续性方程的更新。

记当前残量为 $r_u,r_v,r_p$。DGS 迭代包含两个步骤：

1. **动量松弛 (Momentum Relaxation)**：
   固定压力 $p$，对速度 $u,v$ 进行 Gauss-Seidel 扫描：

$$u_{i,j}\leftarrow u_{i,j}+\omega\frac{h^2}{4}(r_u)_{i,j}, \quad v_{i,j}\leftarrow v_{i,j}+\omega\frac{h^2}{4}(r_v)_{i,j} \tag{7}$$

2. **散度校正 (Divergence Correction)**：
   为了消除散度残量 $r_{div}=-\nabla_h\cdot\vec{u}$，我们寻找修正量 $(\delta u,\delta v,\delta p)$。在 DGS 中，这是通过更新分布函数 $\phi$ 实现的：

$$\delta u=\partial_x\phi, \quad \delta v=\partial_y\phi, \quad \delta p=\Delta_h\phi \tag{8}$$

   在离散层面上，这意味着对每个单元 $(i,j)$，计算散度残量 $r_{i,j}$，然后进行如下分布更新：

$$\begin{aligned} &p_{i,j}\leftarrow p_{i,j}+r_{i,j} \\ &p_{nb}\leftarrow p_{nb}-w\cdot r_{i,j} \quad (\text{distribute to neighbors to mimic Laplacian}) \\ &u_{i,j}\leftarrow u_{i,j}-\frac{h}{4}r_{i,j}, \quad u_{i-1,j}\leftarrow u_{i-1,j}+\frac{h}{4}r_{i,j} \\ &v_{i,j}\leftarrow v_{i,j}-\frac{h}{4}r_{i,j}, \quad v_{i,j-1}\leftarrow v_{i,j-1}+\frac{h}{4}r_{i,j} \end{aligned} \tag{9}$$

   (注：具体系数取决于边界处理)

## 2.2 算法流程

采用标准几何多重网格 V-cycle：

- **限制算子** $I_h^{2h}$：全加权限制（Full Weighting）。
- **提升算子** $I_{2h}^h$：双线性插值。
- **粗网格求解**：当网格过小时，进行多次松弛。

### Algorithm 1: V-cycle DGS-MG

1. **Pre-smoothing**: Apply DGS smoother $\nu_1$ times on $A_h x_h = b_h$.
2. **Restriction**: Compute residual $r_h = b_h - A_h x_h$, restrict $r_{2h} = I_h^{2h} r_h$.
3. **Coarse Grid Correction**:
   - If on coarsest grid, solve $A_{2h} e_{2h} = r_{2h}$ directly.
   - Else, call $e_{2h} = \text{V-cycle}(A_{2h}, r_{2h})$.
4. **Prolongation**: Interpolate correction $e_h = I_{2h}^h e_{2h}$.
5. **Correction**: $x_h \leftarrow x_h + e_h$.
6. **Post-smoothing**: Apply DGS smoother $\nu_2$ times.

## 2.3 数值结果与分析

取停机标准 $\|r_h\|_2 / \|r_0\|_2 \leq 10^{-8}$。对比不同平滑次数配置下的表现。

**表 1：V-cycle 性能对比 (组 D: $\nu = 4, N/L = 4$)**

| $N$ | V-cycle 次数 | CPU 时间 (s) | 误差 $e_N$ | 收敛阶 |
|---|---|---|---|---|
| 64 | 14 | 0.0210 | 1.4951e-03 | - |
| 128 | 15 | 0.0555 | 3.7364e-04 | 2.00 |
| 256 | 16 | 0.2250 | 9.3399e-05 | 2.00 |
| 512 | 16 | 1.0780 | 2.3350e-05 | 2.00 |
| 1024 | 16 | 7.3337 | 5.8407e-06 | 2.00 |
| 2048 | 16 | 29.3548 | 1.4824e-06 | 2.00 |

**分析**：

1. **收敛阶**：误差数据 $e_N$ 严格遵循 $O(h^2)$，验证了离散格式的正确性。
2. **网格无关性**：迭代次数随 $N$ 增加基本保持恒定（约 16 次），体现了多重网格方法最优的计算复杂度。
3. **效率**：在 $N = 2048$ 规模下仅需 29.35秒，是所有测试方法中最快的。

---

# 3. 问题二：Uzawa 迭代法

## 3.1 算法原理

Uzawa 方法基于 Schur 补系统 $B^T A^{-1} B P = B^T A^{-1} F$。该方法将原问题解耦为两个子问题：

### Algorithm 2: Uzawa Iteration

1. Initialize $P_0$, set $k = 0$.
2. While $\|r\| > \text{tol}$:
   a. **Solve Velocity**: Find $U_{k+1}$ such that $AU_{k+1} = F - BP_k$.
   b. **Update Pressure**: $P_{k+1} = P_k + \alpha(B^T U_{k+1} - D)$.
   c. $k \leftarrow k + 1$.

步骤 (a) 中 $A$ 为对称正定矩阵，使用共轭梯度法 (CG) 求解至高精度（$10^{-10}$）。由于 $B^T A^{-1} B$ 的谱半径 $\rho \approx 1$，最优步长取 $\alpha = 1$。

## 3.2 数值结果

取停机标准 $10^{-8}$。由于内层求解代价过高，仅测试至 $N = 512$。

**表 2：Uzawa 方法求解结果**

| $N$ | 外迭代次数 | CPU 时间 (s) | 误差 $e_N$ |
|---|---|---|---|
| 64 | 2 | 0.0326 | 1.4951e-03 |
| 128 | 2 | 0.8296 | 3.7363e-04 |
| 256 | 2 | 4.8331 | 9.3398e-05 |
| 512 | 2 | 29.5871 | 2.3349e-05 |

**分析：**
虽然外层迭代次数极少（2次），但由于内层需要精确求解 $A^{-1}$，计算时间随 $N$ 急剧增加。在 $N = 512$ 时，Uzawa 方法耗时 29.6s，而 V-cycle 仅需 1.1s。这表明经典 Uzawa 方法不适合求解大规模问题。

---

# 4. 问题三：Inexact Uzawa 迭代法

## 4.1 算法原理：MG-Preconditioned Inexact Uzawa

为了克服经典 Uzawa 内层求解昂贵的问题，采用非精确求解策略。

1. **非精确求解**：步骤 (a) 中 $AU_{k+1} \approx F - BP_k$ 仅需满足残量 $\|r_{inner}\| \leq \max(\varepsilon, \tau\|B^T U_k\|)$。
2. **预条件子**：为了加速内层 CG 的收敛，使用针对 Poisson 方程的 V-cycle 多重网格作为预条件子 $M^{-1}$。即在 PCG 算法中，计算 $z = M^{-1} r$ 等价于调用一次 `Poisson_MG_Vcycle(r)`。

### Algorithm 3: Inexact Uzawa with MG-PCG

1. Initialize $P_0$.
2. While $\|r_{total}\| > \text{tol}$:
   a. **Approximate Velocity**: Solve $AU_{k+1} \approx F - BP_k$ using **PCG**.
      * *Preconditioner*: One V-cycle sweep on $A_u, A_v$.
      * *Tol*: Adaptive tolerance based on divergence norm.
   b. **Update Pressure**: $P_{k+1} = P_k + \alpha(B^T U_{k+1})$.
   c. $k \leftarrow k + 1$.

## 4.2 数值结果与参数分析

对比不同内层容差 $\tau$、步长 $\alpha$ 及预条件质量（平滑次数 $\nu$）的影响。

**表 3：Inexact Uzawa 性能对比 ($N = 2048$)**

| 参数组 | 配置 $(\tau, \alpha, \nu)$ | 外迭代 | 内层 PCG 总步数 | CPU 时间 (s) |
|---|---|---|---|---|
| **组 2 (基准)** | $10^{-3}, 1.0, 4$ | 3 | 42 | **85.02** |
| **组 1 (弱预优)** | $10^{-3}, 1.0, 2$ | 4 | 120 | 149.11 |
| **组 3 (过松弛)** | $10^{-3}, 1.05, 2$ | 5 | 120 | 161.36 |
| **组 7 (高精度)** | $10^{-4}, 1.0, 4$ | 3 | 42 | 84.70 |

## 4.3 结果分析

1. **预条件效果**：对比组 1 ($\nu = 2$) 和组 2 ($\nu = 4$)，增加 MG 的平滑次数虽然增加了单次 V-cycle 的成本，但显著减少了内层 PCG 的迭代次数（从 120 降至 42），从而将总时间从 149s 优化至 85s。
2. **最优参数**：数值结果显示 $\alpha = 1.0$ 为最优步长。偏离该值会导致外层收敛变慢。
3. **求解能力**：相比经典 Uzawa 无法在合理时间内求解 $N = 2048$ 的问题，Inexact Uzawa 结合 MG 预条件成功将求解时间控制在 1.5 分钟以内。虽然慢于纯 V-cycle DGS 方法（约 30s），但其作为一种模块化更强的 Schur 补方法，表现出了极佳的鲁棒性。

---

# 5. 结论

本报告实现了基于交错网格的 Stokes 方程求解器，并对比了三种数值方法。结论如下：

1. **离散精度**：所有方法均验证了 MAC 格式在速度场上的二阶精度。
2. **效率对比**：
   - **DGS-MG (V-cycle)**：效率最高，具有 $O(N)$ 复杂度，是求解此类结构化网格问题的首选。
   - **Inexact Uzawa (MG-PCG)**：通过有效的预条件技术，克服了经典 Uzawa 的效率瓶颈，适用于需要解耦求解器的场景。
   - **Classical Uzawa**：仅适用于小规模问题验证，不具备大规模求解能力。