

摘要

混合有限元是求解不可压流体方程的一种重要方法，它相对于一般的有限元方法，有很多的优点。在计算流体问题时，通常的有限元是把速度和压力分开来求解，先计算压力，然后对压力求导之后再计算速度，这样做会降低计算精度。而混合元方法是把速度和压力耦合在一起求解，从而提高计算精度。混合元涉及到两个有限元空间，这两个有限元空间需要满足 inf-sup 或 LBB 条件来保证混合有限元解的存在唯一性。因此这个条件对于求解非常的关键。

本文使用的是 $4P_1 - P_1$ 元，因为这种混合元自然满足 inf-sup 条件，保证了有限元解的存在唯一性，而且在速度和压力空间都只用线性元逼近。 $4P_1 - P_1$ 元是基于两层嵌套的网格结构，即速度网格和压力网格，而速度网格由压力网格加密一次得到。这种两层网格结构其实是一种特殊的四叉树，它有以下两个优点：

- (1) 这种树结构为有利于开发数值代数的快速求解器，即代数多重网格预处理；
- (2) 为自适应网格的求解提供了数据结构上的便利。

在邸教授 [28] 的工作基础之上，基于 $4P_1 - P_1$ 元，我们推广了求解不可压 Navier-Stokes 方程的移动网格有限元方法。这种方法在求解不可压流体时，不局限于周期性边界条件。我们的移动网格策略是：首先移动压力网格，然后通过树状结构来同步速度网格。为了检验这种方法可以求解更一般的流体问题，本文算例中采用的边界均是 Dirichlet 或者 Neumann 条件。

为了提高计算效率，我们结合 Elman 在 [38] 中对不可压 N-S 的 PCD 预处理方法以及已有的代数多重网格预处理策略，用移动网格有限元求解 N-S 方程组。特别是，在从旧网格到新网格的插值过程中，我们也使用了代数多重网格预处理策略。从而，我们提出了一种代数多重网格预处理策略，来加速移动网格有限元方法求解不可压 N-S 方程组。最后，我们在数值算例中展示这种预处理策略的高效性。

关键词： $4P_1 - P_1$ 元，树结构，移动网格方法，多重网格预处理，PCD 预处理。

Abstract

It is an important method for solving incompressible flow equations with mixed finite element method, and it has more advantages than general finite element method. When solving flow problems, velocity and pressure are separately solved, pressure solution is computed firstly, velocity solution is obtained after pressure derivation in general finite element method. This may reduce the calculation precision. While velocity and pressure are computed together in mixed finite element method, it will increase the calculation accuracy. It contains two finite element spaces, which need to satisfy inf-sup or LBB condition. The solutions of mixed finite element method are existed and unique if the inf-sup condition is satisfied, so this condition is very important for solving.

In this work, $4P_1 - P_1$ finite element pair is used, which naturally satisfies the inf-sup condition, so the existence and uniqueness of discrete finite element solutions is guaranteed, And linear approximation is for velocity and pressure finite element spaces. $4P_1 - P_1$ pair is based on two nested mesh structure, velocity mesh and pressure mesh. Velocity mesh can be obtained by globally refining pressure mesh. The structure of this two layer mesh is actually a special four-fork tree, it has two advantages as following:

- (1) This tree structure benefits for developing numerical fast solver, algebraic multi-grid precondition;
- (2) It provides convenience for adaptive mesh solving in data structure.

Based on the work of Prof. Di [28], we promote moving mesh finite element method for solving incompressible Navier-Stokes equations based on the $4P_1 - P_1$ finite element pair. This method is not restricted to period boundary condition in solving incompressible flow. Our moving strategy is: Firstly, we move the pressure mesh; next, we synchronize the velocity mesh by tree structure. For checking this method in solving general flow problem, Dirichlet or Neumann conditions are imposed on the boundaries of numerical experiments in this work.

ABSTRACT

For improving the calculation efficiency, we solve Navier-Stokes equations with moving mesh finite element method, combining the work of Elman [38] that the PCD preconditioning and current AMG preconditioning strategy. Especially, in the process of the interpolation from old mesh to new mesh, we also apply AMG preconditioning strategy. Then we propose an AMG preconditioning strategy, for accelerating the solving incompressible Navier-Stokes equations with moving mesh finite element method. Finally, we show efficiency of the preconditioning strategy by numerical experiments.

Keywords: $4P_1 - P_1$ pair, moving mesh method, AMG preconditioning, PCD preconditioning.

目 录

摘要	i
Abstract	iii
目录	v
第 1 章 绪论	1
1.1 引言	1
1.2 研究背景	2
1.3 论文的组织	3
第 2 章 预备知识	5
2.1 混合有限元	5
2.2 几何遗传树结构	7
2.3 移动网格方法	9
2.4 预处理方法	11
2.4.1 针对Stokes方程	11
2.4.2 针对Navier-Stokes方程	19
第 3 章 基于$4P_1 - P_1$元求解 Navier-Stokes 方程的移动网格有限元方法	23
3.1 数据结构	24
3.2 混合元近似	24
3.2.1 流体方程	24
3.2.2 弱形式	26
3.3 预处理策略	29
3.4 移动网格策略	31

目 录

3.4.1	获取控制函数	31
3.4.2	获取新的逻辑网格	32
3.4.3	物理网格的移动方向	32
3.4.4	散度为 0 的插值	33
3.5	数值算例	36
3.5.1	碰撞流	36
3.5.2	喷射流	37
3.5.3	流体经过一个凹形台阶	38
3.5.4	圆柱绕流	43
第 4 章	不可压 Navier-Stokes 方程的代数多重网格预处理的移动网格有限元方法	47
4.1	数值离散	48
4.1.1	时间方向的离散	48
4.1.2	空间离散	51
4.2	快速 Krylov 求解	52
4.3	移动网格策略	55
4.3.1	网格移动的流程	55
4.3.2	用 AMG 预处理求解 (4.48)	56
4.4	数值算例	58
4.4.1	方腔驱动流	58
4.4.2	向后台阶流	59
4.4.3	圆柱绕流	62
参考文献	65	

表 格

3.1	用移动网格计算碰撞流的误差, $\nu = 1.0$ 。	36
3.2	用移动网格计算碰撞流的散度和涡量误差, $\nu = 1.0$ 。	36
4.1	方腔流: 对比用 AMG 和 ILU 预处理来求解 (4.22) 时, GMRES 的迭代步数, $Re = 2000$ 。	61
4.2	方腔流: 对比用不同的预处理矩阵 M_S^* , 求解 (4.48) 时, GMRES 平均迭代次数, $Re = 2000$ 。	61
4.3	圆柱绕流: 对比用不同的预处理矩阵 F_p 和 M_S^* 分别求解 (4.22) 和 (4.48) 时, GMRES 的平均迭代次数, $\alpha = 5, Re = 240$ 。	63

插 图

2.1 左: 压力 p 单元, \circ 为 p 单元上的自由度; 右: 4个速度 v 单元, \bullet 表示 v 单元上的自由度.	8
2.2 全局加密一次的网格 9	
2.3 它的几何遗传树 9	
2.4 几何遗传树结构 9	
3.1 左: p 单元; 右: 与压力单元对应的 4 个速度 v 单元 29	
3.2 左: $m = \frac{1}{G_1}$ 的等高线和网格移动方向; 右: 压力等高线和速度的流速线, $\alpha = 0.002, \beta = 2.0$ 37	
3.3 速度 L^2 误差分布, 左: 均匀网格; 右: 移动网格 $\alpha = 0.002, \beta = 2.0$ 37	
3.4 网格对比, 左: 均匀网格 20×20 ; 右: 移动网格 $\alpha = 0.002, \beta = 2.0$ 38	
3.5 顶部: 移动网格, 底部: 涡量等高线, $t = 12s$ 39	
3.6 顶部: 移动网格, 底部: 涡量等高线, $t = 27s$ 40	
3.7 顶部: 移动网格; 底部: 涡量等高线, $t = 0.5s$ 41	
3.8 顶部: 移动网格; 底部: 涡量等高线, $t = 1s$ 41	
3.9 顶部: 移动网格; 底部: 涡量等高线, $t = 1.5s$ 42	
3.10 顶部: 移动网格; 底部: 涡量等高线, $t = 2s$ 42	
3.11 顶部: 压力等高线; 底部: 速度流速线, $t = 2s$ 43	
3.12 顶部: 均匀网格; 底部: 初始移动网格, 粘性系数 $\nu = 0.003$ 44	
3.13 顶部: 移动网格; 底部: 涡量等高线, $t = 24.5s$, 粘性系数 $\nu = 0.003$ 44	
3.14 顶部: 移动网格; 底部: 涡量等高线, $t = 23s$, 粘性系数 $\nu = 0.001$ 45	

4.1 全局加密一次的网格	52
4.2 它的几何遗传树	52
4.3 几何遗传树结构	52
4.4 方腔流: 左: 移动网格, 右: 涡量的等高线, 压力网格 20×20 , $Re = 2000$	59
4.5 方腔流, 速度的散度等高线, 左: 均匀网格, 右: 移动网格, 压 力网格 20×20 , $Re = 2000$	59
4.6 方腔流: 流速线, 压力网格 20×20 , $Re = 2000$	60
4.7 方腔流: 对比在 PCD 预处理中用不同的 F_p 来求解 (4.22) 时, GMRES 的迭代步数, 压力网格 20×20 , $Re = 2000$	60
4.8 向后台阶流, 上: 移动网格, 下: 流速线, $t = 100s$, 粘性系数 $\nu = 0.02$	61
4.9 向后台阶流: 对比不同的 M_S^* 求解 (4.48) 时, GMRES 随时间的 迭代次数, $\nu = 0.02$	62
4.10 圆柱绕流: 对比不同 α , 用 PCD 预处理求解 (4.22), GMRES 的 迭代次数, $Re = 240$	63
4.11 圆柱绕流, 上: 移动网格, 下: 涡量等高线, $t = 2s$, $Re = 240$	64
4.12 圆柱绕流, 上: 移动网格, 下: 涡量等高线, $t = 40s$, $Re =$ 240.	64

第 1 章 绪论

1.1 引言

在自然界中，很多物理现象都是满足一些自然规律，从而根据这些自然规律对物理现象进行建模。反过来通过对模型的分析和进一步研究，来更好的把握和利用自然规律。然而这些模型通常又是用偏微分方程来进行描述，比如说描述流体运动的 Navier-Stokes 方程组。这个方程组中是建立在三大物理守恒定律之上的，即质量守恒，动量守恒和能量守恒。N-S 方程组是以流体连续性为基础，在宏观上描述牛顿流体。在宏观尺度下，我们不考虑分子的结构和运动，用速度、压力、密度、温度这些宏观量以及他们在空间的导数以及他们自身的变化率来描述流体特性。由于 N-S 方程组是非线性的，寻找它的精确解是很困难的，因此通常我们借助计算机，用数值的方法去离散并求解 N-S 方程组从而使得在更多的领域中，数值计算得到很大的重视，比如说大气流动，海洋洋流运动，以及飞机汽车的外形流线设计，甚至是在微流中，如血液的流动等等。在实际的物理中，还有更为复杂的流体，比如说多相流(气液、固液两相流)。特别地，当流体的粘性非常小的时候，还会出现湍流现象。这些都要求我们在数值计算上能够提供更好的计算方法，能够高效的求解这些流体方程，并且使得计算出的数值解是科学的，是符合物理规律的。

在实际的计算过程中，我们通常会遇到，问题的解在某个局部的区域内有比较大的奇异性，或者是在流体在局部有比较细微的结构，比如涡。我们为了抓住流体局部的性质，保证在这个区域内的计算精度，通常要使用小尺度的网格。然而在全局都用同样尺度的网格，会使得我们的计算效率不高，甚至是一些计算上的浪费。特别是在高维的问题中，用自适应的网格，即局部需要小尺度网格，不需要精确计算的区域用相对比较大的网格，会在保证计算精度的条件下，极大的降低计算量。

如何快速的求解 N-S 方程，也是需要着重考虑的。因为方程本身是非线性的，如果用全隐格式的数值方法离散，需要用非线性迭代来求解，例如牛顿迭代，或者是将非线性项线性化。结合自适应网格后，会使得离散出的系数矩阵条件数会发生改变，特别当矩阵的阶数非常大的时候，从而给求解效率带来一

定的挑战。构建高效的求解方法，势在必行。多重网格方法可以使得矩阵求解的迭代次数跟矩阵的阶数是无关的，因此，在求解中用上多重网格方法，将会很大程度提高计算效率。

在本文中，在自适应网格上求解不可压 N-S 方程，并且研究方程求解的代数多重网格预处理方法。

1.2 研究背景

Navier-Stokes 方程组包括质量守恒、动量守恒和能量守恒方程，动量守恒方程即是由牛顿第二定律得到的。不可压的条件意味着质量守恒方程退化为速度散度为 0 的方程。但是散度为 0 不显示含有压力，因此这给不可压 N-S 方程带来了困难。数值求解不可压 N-S 方程的困难点之一是速度和压力要耦合在一起求解。在数值求解中，需要选取交错网格或者部分交错网格来满足，如 MAC 方法([45]) 和 ALE 方法([47])。这些方法有一定的局限性。在有限元分析中，只要满足 inf-sup (LBB) 条件 [43]，速度和压力的有限元空间就能保证散度为 0 的有限元近似收敛到散度为 0 的精确解。因此为了保证 inf-sup 条件的成立，要么在相同的网格上，速度和压力采取不同阶的近似，要么就采用不同的网格，同阶的近似。这就是混合有限元方法的想法详见 [18]。在混合元方法中，速度有限元空间和压力有限元空间要满足 inf-sup 条件，关于这个条件的验证可以参考([23]、[18]等著作)。我们知道的自然满足 inf-sup 条件的有限元对如 Taylor-Hood 元、Mini 元，RT 元等等，还有一些不满足 inf-sup 条件的有限元对如 $P_1 - P_1$ 、 $Q_1 - Q_1$ 、 $Q_1 - P_0$ 、 $P_1 - P_0$ 元。对于不满足 inf-sup 条件的有限元对，需要在压力空间施加约束进行稳定化，例如罚方法。有关的文献可以参考([22]、[75]、[52]、[11]等)。但是在工程计算中，当计算区域不规则比如说有一些拐角，或者数值解有一定的奇异性时，我们更倾向于使用低阶近似并且会用到自适应网格方法。自适应网格方法一般分为 h 自适应加密方法和移动网格方法(r 自适应)。在([24]、[33]、[10]) 中提到 h 自适应的 $P_2 - P_1$ 元。稳定化的 $P_1 - P_0$ 和 $P_1 - P_1$ 有限元方法结合自适应网格方法参考([77]、[78])。

Di 在 [28] 中将一种基于调和映射的移动网格有限元方法([56]、[57]) 应用到原始变量形式的不可压 N-S 方程的求解。调和映射应用到自适应网格生成是在一系列工作([74]、[31])的基础之上。Di 在 [28] 的基础之上，将移动网格有限元方法应用到求解多相流问题上([27]、[30])。除了上述移动网格方法，还有应

用比较广泛的由Huang 和Russell 发展的移动网格偏微分方程方法([49]、[48])。在这种方法中，物理方程是在逻辑网格上与网格控制方程进行联立求解，不利于程序块化。还有一些其它值得关注的移动网格方法见([4]、[71])等。

对不可压 N-S 方程组的用混合元离散会导出一个鞍点问题，针对鞍点问题的求解分为直接解法和迭代解法。在迭代的求解中又有很多种方法，例如 Uzawa 方法，罚方法，乘子方法，Krylov 子空间方法等等，可以参考 [5] 中的综述。如何为迭代方法提供好的预处理子，成为加速问题求解的关键。我们主要考虑 Krylov 子空间方法的块预处理、块三角预处理，参考 [38]。在块预处理或块三角预处理中，关键的是要找到Schur 补矩阵好的近似。Elman 在 [38] 中提出了两种近似 Schur 补矩阵的方法，压力对流扩散以及最小二乘交換子预处理。Boyle 在([12] [13])中，提出了一种基于多重网格迭代的压力对流扩散和最小二乘预处理方法。还有其他的预处理方法比如 ILU(incomplete LU) [8]等。多重网格方法也是求解不可压流体的重要方法，Xu 提出两格子方法([46]、[64]、[20]、[58])。Wabro [70] 等将代数多重网格应用到不可压流体方程的求解。维数分解方法 [6] 也是求解方法之一。用基于 $4P_1 - P_1$ 元的移动网格有限元方法来求解 N-S 方程，并且为方程求解提供高效的代数多重网格预处理是本文研究的关键。

1.3 论文的组织

第 2 章中介绍了一些研究中需要用到的方法和理论，包括混合元方法、网格的数据结构、移动网格方法以及与 Stokes 方程和 N-S 方程的预处理方法。在混合元方法中，以 Stokes 方程为例介绍了离散 inf-sup(LBB) 条件，以及混合元离散后，问题的存在唯一性。在网格的数据结构中，简述了几何遗传树的四叉树结构。在预处理方法中，我们通过介绍矩阵的特征值分析，引入常用的块预处理方法。

第 3 章中我们用一种满足 LBB 条件的混合元 $4P_1 - P_1$ 元，这种混合元的网格结构是建立在几何遗传树上，速度和压力分别用两套网格，但是速度网格由压力网格加密一次得到，通过几何遗传树来建立速度网格和压力网格间的 1 – 1 对应关系。这样我们可以压力速度全部用线性元来拼装矩阵。我们用这种有限元来结合移动网格来求解 N-S 方程。

第 4 章中我们研究了用代数多重网格迭代预处理方法来加速 N-S 方程的

求解，以及数值解在新旧网格守恒差值过程中方程的求解。在 N-S 方程的求解中，我们采用代数多重网格迭代的 PCD 预处理，并且在预处理矩阵求逆过程中，用几步代数多重网格迭代来代替精确求解，可以加速 N-S 方程的求解。在移动网格新旧网格的数值解守恒插值过程中，我们用压力刚度矩阵来近似压力 Schur 补矩阵，使计算效率得到很大的提高。

第 2 章 预备知识

我们把后文中用到的混合有限元方法，构建混合元所用的几何遗传树结构，移动网格方法，以及预处理方法，在本章中做简要描述和历史回顾。

2.1 混合有限元

混合有限元方法的专著可以参考 F.Brezzi [18]、[1]。二阶椭圆性问题的混合有限元方法就是基于 Hellinger-Reissner 变分原理的有限元方法。在流体中混合元方法是把速度和压力耦合在一起求解，因而精度得到提高，体现出相对于分开求解速度和压力有限元方法的优势。另外，对于不可压流体中的 Galenkin 逼近，只能采取混合有限元。在这一节中，我们以 Stokes 方程为例，来介绍混合元方法。Stokes 方程可以看做是 Navier-stokes 的简化，去掉了非线性项。Stokes 方程有两个需要求解的函数速度 \mathbf{u} 和压力 p 是耦合的，并且速度要满足质量守恒的条件($\nabla \cdot \mathbf{u} = 0$)。因此，需要我们用混合有限元方法来求解方程。为确保方程的解存在唯一，速度和压力所在的有限元空间必须满足 inf-sup 条件。如果有限元不满足 inf-sup(LBB) 条件，例如在四边形单元中的 $Q_1 - Q_1$ 、 $Q_1 - P_0$ ，三角形单元中的 $P_1 - P_1$ 、 $P_1 - P_0$ 元，还需要在压力空间上做稳定性。

我们考虑 Stokes 方程

$$-\nabla^2 \mathbf{u} + \nabla p = \mathbf{0}, \quad (2.1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2.2)$$

在 $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ 上的边界条件是

$$\mathbf{u} = \mathbf{w} \text{ 在 } \partial\Omega_D \text{ 上, } \frac{\partial \mathbf{u}}{\partial n} - \mathbf{n}p = \mathbf{s} \text{ 在 } \partial\Omega_N \text{ 上.} \quad (2.3)$$

它的弱形式为：找到 $(\mathbf{u}, p) \in (H_0^1(\Omega))^2 \times L_0^2(\Omega)$ 使得

$$\int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} - \int_{\Omega} p \nabla \cdot \mathbf{v} = \int_{\partial\Omega_N} \mathbf{s} \cdot \mathbf{v}, \quad \text{对所有的 } \mathbf{v} \in X, \quad (2.4)$$

$$\int_{\Omega} q \nabla \cdot \mathbf{u} = 0 \quad \text{对所有的 } q \in P, \quad (2.5)$$

其中 $X \in (H_0^1(\Omega))^2$, $P \in L_0^2(\Omega)$, $L_0^2 = \{q \in L^2(\Omega) | \int_{\omega} q dx = 0\}$ 。令

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v}, \quad (2.6)$$

$$b(\mathbf{u}, q) = - \int_{\Omega} \nabla \cdot \mathbf{u} q. \quad (2.7)$$

有限元空间 X, P 要满足 inf-sup 条件或者 LBB 条件，即

$$\gamma \|q\|_P \leq \sup_{\mathbf{v} \in X} \frac{b(\mathbf{v}, q)}{\|\mathbf{v}\|_X}, \quad \forall q \in P. \quad (2.8)$$

这个条件的验证由很多专著中给出，例如([18]、[16]、[23] 等)。我们不加证明地给出如下定理：

定理 2.1. 问题 (2.4) 和 (2.5) 的解存在唯一。

在研究用混合元去离散 Stokes 问题时，我们构造的有限维子空间 $(X_h, P_h) \subset (X, P)$ 要满足离散的 inf-sup 条件：

$$\gamma_0 \|q_h\|_{P_h} \leq \sup_{\mathbf{v}_h \in X_h} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{X_h}}, \quad \forall q_h \in P_h, \quad (2.9)$$

其中 β_0 是一个跟网格尺度无关的常量。Mini 元，RT 元以及 Taylor-Hood 元都是满足离散的 LBB 条件的。我们得到 (2.4) 和 (2.5) 对应的离散形式：寻找 $(\mathbf{u}_h, p_h) \in X_h \times P_h$ 使得

$$\int_{\Omega} \nabla \mathbf{u}_h : \nabla \mathbf{v}_h - \int_{\Omega} p_h \nabla \cdot \mathbf{v}_h = \int_{\partial \Omega_N} \mathbf{s} \cdot \mathbf{v}_h \quad \text{对所有的 } \mathbf{v}_h \in X_h^h, \quad (2.10)$$

$$\int_{\Omega} q_h \nabla \cdot \mathbf{u}_h = 0 \quad \text{d对所有的 } q_h \in P_h. \quad (2.11)$$

求解 (2.10) 和 (2.11)，得到的离散数值解 (\mathbf{u}_h, p_h) ，如果 (\mathbf{u}, p) 是弱形式 (2.4) 和 (2.5) 的解，则我们可以得到对应的误差估计：

$$\|\mathbf{u} - \mathbf{u}_h\|_1 + \|p - p_h\|_0 \leq C \left\{ \inf_{\mathbf{v}_h \in X_h} \|\mathbf{u} - \mathbf{v}_h\|_1 + \inf_{q_h \in P_h} \|p - q_h\|_0 \right\}. \quad (2.12)$$

我们倾向于使用满足 LBB 条件的混合元 Taylor-Hood 元，如 $P_2 - P_1$ 、 $Q_2 - Q_1$ 元，即速度是二次元，压力是线性元。在实际的工程计算中，因为线

性元简单，所以比较受欢迎。因此我们设想，如果把稳定的 Taylor-Hood 元中的速度二次元换成线性元，同时又要保证速度和压力满足 inf-sup 条件，这就是 $P_1isoP_2P_1$ 的想法。如图 2.1 所示，因为压力是线性元，因此在一个三角形单元上有三个自由度，分布在三角形的三个顶点上。同样地，速度是二次元，在这个三角形单元个有 6 个自由度，分布在三角形的三个顶点和三边的中点上。如果这个三角形单元加密一次，即连接三边中点，因此得到四个小三角形。这样在每个小三角形上有分布着三个自由度。在 [9] 中证明了 $P_1isoP_2P_1$ 是满足 inf-sup 条件的。令 $X_h = X_{h/2}^1, P_h = P_h^1$ ，则

定理 2.2. 如果 Ω 是一个多边形，并且对所有 h 都有 $\Omega_h = \Omega$ ，如果 Ω_h 中所有的三角形都至少有一个顶点不在 Ω 的边界 Γ 上， X_h, P_h 是满足 LBB 条件的，那么问题 (2.10)、(2.11) 是有唯一解 $(\mathbf{u}_h, p_h) \in X_h \times (P_h/R)$ (p_h 除了一个常数外是确定的)。

定理的过程我们不再给出，[9] 中还给出了 $P_1isoP_2P_1$ 元的误差估计：

$$|\nabla(\mathbf{u} - \mathbf{u}_h)|_0 \leq hK(||\mathbf{u}||_{(H^2(\Omega))^2} + ||p||_{H^1(\Omega)/R}), \quad (2.13)$$

$$|\nabla(p - p_h)|_0 \leq K(||\mathbf{u}||_{(H^2(\Omega))^2} + ||p||_{H^1(\Omega)/R}), \quad (2.14)$$

其中 K 是与 h 无关的常数。但是需要注意到图 2.1 中，每个小三角形上的三个自由度，其实是大的三角形 6 个自由度中的其中 3 个。因此，在做速度单元和压力单元拼装的时候，需要先找到速度单元基函数所在的大单元，然后用大单元的基函数中正好位于当前速度单元上的基函数与压力单元上的基函数进行拼装。这会带来一些技术上的困难。如果图 2.1 右中每个小三角形上的自由度都是局部在一个单元上，这时跟 $P_1isoP_2P_1$ 拥有同样的网格结构，也满足 inf-sup 条件，但速度单元全部是线性元，因此称作 $4P_1 - P_1$ 元。在这种情况下，只要建立速度单元和压力单元之间的索引，速度单元与压力单元间的拼装变得容易很多。

2.2 几何遗传树结构

在上一节中，我们提到速度单元和压力单元之间的索引，而这个索引的建立利用了 [53] 中的几何遗传树结构。这种树结构是用来做 $h-$ 自适应，下面我们就介绍一下这种树结构。

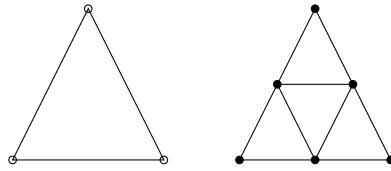


图 2.1: 左: 压力 p 单元, \circ 为 p 单元上的自由度; 右: 4个速度 v 单元, \bullet 表示 v 单元上的自由度.

在 $h-$ 自适应中, 两个耦合在一起的物理量(如上一节提到的速度 \mathbf{u} 和压力 p), 有可能在不同的地方出现奇异, 因此需要分别在两套网格上离散数值解。在拼装 $\int_{\Omega} \nabla \cdot \mathbf{u} p$ 这一项时, 因为 \mathbf{u} 和 p 在不同的网格上, 因此需要构建两套网格之间的联系。如果函数在相对于计算区域来说, 很小的区域内有奇异, 好的网格需要在这个奇异出现的地方网格非常集中。特别地, 如果最小的网格单元和最大的网格单元面积相差上百倍, 这时候建立两个网格之间对应的代价是非常昂贵的。为了创建高效的多网格自适应策略, [53] 中选取了几何遗传树, 具有遗传性的网格结构可以描述为点、线、面、体。并且针对无结构网格, 加密也是通过按层次来加密的。这种分层的数据结构可以提供简单的编程方法和快速的网格加密算法, 以及多重网格求解器。

我们以二维区域 Ω 上的一个三角剖分 \mathcal{T}_0 为例, 来介绍一下几何遗传树结构。为与有限元空间的单元区分开, 称三角剖分中的单元为几何单元。 τ 为 \mathcal{T}_0 的几何单元。三角剖分中所有几何性质都有属于的关系, 即如果一条边是一个三角形三边中的一条, 则称这条边属于这个三角形。同样的, 如果一个点是一条边两个顶点中的一个, 则称这个点属于这条边。三角剖分中的一个三角形可以被加密成四个小的三角形。在三角形加密的过程中, 三条边分别被分成了两部分。当三角剖分中所有的三角形都加密一次, 我们可以得到一个更密的三角剖分 \mathcal{T}_1 , 这就是 \mathcal{T}_0 的一次全局加密。经过逐次加密, 我们可以得到一系列的三角剖分 $\{\mathcal{T}_n\}$ 。如果 \mathcal{T}_n 中的一个三角形 τ_1 位于 \mathcal{T}_{n-1} 中的三角形 τ_0 中, 则 τ_1 被称作是 τ_0 的一个孩子。我们给 \mathcal{T}_0 中的所有三角形一个虚拟的父亲, 与所有从 \mathcal{T}_0 的三角形中派生出的所有四叉树组成了一个树的数据结构, 如图 2.4 所示。这棵树也被称为几何遗传树。用几何遗传树结构来构建 h -自适应的内容就不再详细介绍了。

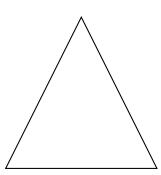


图 2.2: 全局加密一次的网格

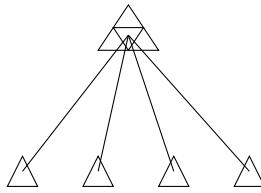
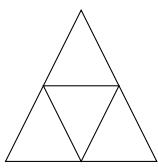


图 2.3: 它的几何遗传树

图 2.4: 几何遗传树结构

2.3 移动网格方法

移动网格方法常常用在数值计算中，寻找好的网格(跟具体问题有关)，并在网格上进行数值求解，使得在不增加自由度的前提下，尽可能的提高计算精度。移动网格方法包括网格的移动策略，以及在方程在网格上的数值求解算法。在 [19]、[73]、[68] 中系统地给出了移动网格方法的介绍。移动网格的策略是基于等分布原则(Equi-distribute Principle)，这是 de Boor 在 1974 年 [25] 提出的。等分布原则是根据一个能反映局部网格计算质量的指标，比如误差，移动网格节点使得这个指标在网格上均匀分布。如果在计算结果误差比较大的地方网格集中，尽量使得误差分布的比较均匀，从而提高计算精度。

我们不加证明地给出一维情形下的等分布原理，如下：

定理 2.3. 在一维空间中，对任意一个函数 $M = M(x) > 0, x \in [a, b]$ ，给定一个正整数 $N > 0$ ，将区间 $[a, b]$ 割分成 N 份 $[x_0, x_1], \dots, [x_{N-1}, x_N]$ ，使得在这些区间上

$$\int_{x_0}^{x_1} M(x)dx = \int_{x_1}^{x_2} M(x)dx = \dots = \int_{x_{N-1}}^{x_N} M(x)dx, \quad (2.15)$$

成立，那么这组剖分是存在唯一的。

如果函数 $M(x)$ 取成误差函数，或者是跟误差同阶的量，那么这组剖分使得误差在每个区间上都相等。通常我们称 M 为网格的控制函数。假设 Ω, Ω_c 分别为物理区域和计算区域，那么 Ω 与 Ω_c 之间的坐标变换可以用调和映射来构造。Dvinsky [31] 和 [14] 指出调和映射可以应用到自适应网格的生成中。调和映射的存在唯一性在 [44] 和 [63] 中给出，如果 Ω_c 满足曲率是非正的，并且 $\partial\Omega_c$ 在 Ω_c 下的度量是凸的，那么从 Ω 到 Ω_c 的调和映射是唯一的。如果把 Ω_c 选为凸区域，并且在 Ω_c 上用欧式度量，则上述两个条件都可以满足。通过求

解方程

$$\frac{\partial}{\partial x_i} \left(G^{ij} \frac{\partial \xi}{\partial x_j} \right) = 0, \quad (2.16)$$

可以得到从 Ω 到 Ω_c 的调和映射。其中 ξ 是逻辑区域的坐标， x_i 是物理区域的坐标 \mathbf{x} 的分量。而 $M = 1/G^{ij}$ 称为控制函数。

Li 在 [56] 中提出一种移动有限元方法，它是基于调和映射理论的。主要想法是先选取一个初始的逻辑区域，进而获得初始物理区域到初始逻辑区域之间的调和映射，在问题发展中，使得物理网格的对应的逻辑网格逐渐接近于调和映射。通过求解一个 Possion 方程

$$\begin{aligned} \Delta \xi &= 0, \\ \xi|_{\partial\Omega} &= \xi_b, \end{aligned} \quad (2.17)$$

来获得一个初始的逻辑网格 ξ^0 ，相当于控制函数全部取 1。这个初始逻辑网格获得之后，在后面的计算中保持不变。在将旧网格上的数值解插值到新网

Algorithm 2.1 网格重新分布

- 1: 给出物理网格 $\mathbf{x}^{(n)}$ 上的数值解 $\mathbf{u}^{(n)}$ 并通过求解 (2.17)，获得 ξ^* 。
 - 2: 判断 $\|\xi^* - \xi^{(0)}\|_{L^2}$ 是否小于容忍量 ϵ ，如果是迭代结束，否则，继续做 3 - 6。
 - 3: 用 $\xi^* - \xi^{(0)}$ 两者之差来计算物理网格的移动量 $\delta\mathbf{x}$ 。
 - 4: 利用 3 中 $\delta\mathbf{x}$ ，在网格 $\mathbf{x}^{(n)}$ 上将 $\mathbf{u}^{(n)}$ 更新到新网格上
 - 5: 更新网格 $\mathbf{x}^{(n)} \leftarrow \mathbf{x}^{(n)} + \eta\delta\mathbf{x}$
 - 6: 回到 1。
 - 7: 输出新网格 $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)}$ ，以及网格上的数值解 $\mathbf{u}^{(n)}$
-

格上的过程中，Li [55] 提出了一种保持数值解 L_2 范数的同伦方法。Di [27] 和 Wang [72] 应用代数多重网格方法加速 (2.17) 的求解。需要指出的是这种移动有限元方法将网格的移动和物理方程的求解分别开来，从而使得程序写起来非常方便。针对不同的物理问题，只要提供合适的控制函数，以及新网格更新数值解的部分程序即可。上面的算法 2.1 已经在写在了自适应有限元软件包 AFEPack([54]) 中。

2.4 预处理方法

我们将分别对 Stokes 方程和 Navier-Stokes 方程，介绍预处理的方法。首先我们先给出求解预处理的 MINRES 方法，MINRES 方法是 Krylov 子空间方法中求解对称不定系统的。我们考虑用求解

$$M^{-1}K\mathbf{x} = M^{-1}\mathbf{b}, \quad (2.18)$$

来代替 $K\mathbf{x} = \mathbf{b}$ ，其中 K 是对称不定矩阵， M 作为 K 的预处理矩阵，为了保持矩阵对称性质， M 是对称正定矩阵。通常 M 的选取要保证两点，一是能很好地近似矩阵 A ，一是要求解 M^{-1} 要容易。初始残差 $\mathbf{r}^{(0)} = \mathbf{b} - K\mathbf{x}^{(0)}$ ， k 次迭代 $\mathbf{x}^{(k)}$ 是在矩阵 $M^{-1}K$ 和 $M^{-1}\mathbf{r}^{(0)}$ 张成的 Krylov 子空间

$$\mathbf{x}^{(0)} + \mathcal{K}_k(M^{-1}, M^{-1}\mathbf{r}^{(0)}), \quad (2.19)$$

中。第 k 个残差向量满足

$$\mathbf{z}^{(k)} \in M^{-1}\mathbf{r}^{(0)} + \mathcal{K}_k(M^{-1}K, M^{-1}\mathbf{r}^{(0)}), \quad (2.20)$$

其中

$$\mathcal{K}_k(A, \mathbf{x}) := \text{span}\{\mathbf{x}, A\mathbf{x}, A^2\mathbf{x}, \dots, A^{k-1}\mathbf{x}\}. \quad (2.21)$$

我们给出预处理的 MINRES 迭代算法：在算法 2.2 中，预处理的部分在于求解 $M\mathbf{z}^{(j+1)} = \mathbf{v}^{(j+1)}$ 。在我们的实际计算中，我们主要通过实现 M^{-1} 的求解来达到预处理的过程。更多的关于 Krylov 子空间的知识可以参考 [61]、[69] 等。同样地，在求解非对称的不定矩阵的时候，如 Navier-Stokes 方程离散出的 Ossen 问题，这时候我们要用 GMRES(general minimal residual) 来求解，我们关注的是预处理部分，所以算法不再给出，详见 [62]。

2.4.1 针对 Stokes 方程

我们通过选取有限元空间基函数对 (2.10) 和 (2.11) 进行离散，可以得到一个线性方程组，我们定义这个线性方程组的系数矩阵为：

$$K = \begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & -C \end{bmatrix}, \quad (2.22)$$

Algorithm 2.2 预处理的 MINRES 方法

```

1:  $\mathbf{v}^{(0)} = \mathbf{0}, \mathbf{w}^{(0)} = \mathbf{0}, \mathbf{w}^{(1)} = \mathbf{0}, \gamma = 0$ 
2: 选取  $\mathbf{u}^{(0)}$ , 计算  $(\mathbf{1}) = \mathbf{b} - K\mathbf{x}$ 
3: 求解  $M\mathbf{z}^{(1)} = \mathbf{v}^{(1)}$ , 令  $\gamma_1 = \sqrt{\langle \mathbf{z}^{(1)}, \mathbf{z}^{(1)} \rangle}$ 
4: 令  $\eta = \gamma_1, s_0 = s_1 = 0, c_0 = c_1 = 1$ 
5: for  $j = 1 \rightarrow$  残差小于容忍量 do
6:    $\mathbf{z}^{(j)} = \mathbf{z}^{(j)}/\gamma_j$ 
7:    $\delta_j = \langle K\mathbf{z}^{(j)}, \mathbf{z}^{(j)} \rangle$ 
8:    $\mathbf{v}^{(j+1)} = K\mathbf{z}^{(j)} - (\delta_j/\gamma_j)\mathbf{v}^{(j)} - (\gamma_j/\gamma_{j-1})\mathbf{v}^{(j-1)}$ 
9:   求解  $M\mathbf{z}^{(j+1)} = \mathbf{v}^{(j+1)}$ 
10:   $\gamma_{j+1} = \sqrt{\langle \mathbf{z}^{(j+1)}, \mathbf{z}^{(j+1)} \rangle}$ 
11:   $\alpha_0 = c_j\delta_j - c_{j-1}s_j\gamma_j$ 
12:   $\alpha_1 = \sqrt{\alpha_0^2 + s\gamma_{j+1}^2}$ 
13:   $\alpha_2 = s_j\delta_j + c_{j-1}c_j\gamma_j$ 
14:   $\alpha_3 = s_{j-1}\gamma_j$ 
15:   $c_{j+1} = \alpha_0/\alpha_1; s_{j+1} = \gamma_{j+1}/\alpha_1$ 
16:   $\mathbf{w}^{(j+1)} = (\mathbf{z}^{(j)} - \alpha_3\mathbf{w}^{(j)})/\alpha_1$ 
17:   $\mathbf{u}^{(j)} = \mathbf{u}^{(j-1)} + c_{j+1}\eta\mathbf{w}^{j+1}$ 
18:   $\eta = -s_{j+1}\eta$ 
19:  计算残差
20: end for

```

其中如果用满足 LBB 条件的混合元做近似, 则 $C = 0$ 。因为本文中我们选取的是满足 LBB 条件的 $4P_1 - P_1$ 元, 所以我们下面均考虑 $C = 0$ 的情形。令 $\mathbf{v}_h \in X_h$ 和 $q_h \in P_h$, X_h 和 P_h 是在第一节中提到的有限元空间。我们先来定义 \mathbf{v}_h, q_h 的范数:

$$\|\nabla \mathbf{v}_h\| := \langle \mathbf{A}\mathbf{v}, \mathbf{v} \rangle^{1/2}, \quad (2.23)$$

$$\|q_h\| := \langle Qq, q \rangle^{1/2}, \quad (2.24)$$

其中 \mathbf{v} 和 q 是速度和压力变量在基函数 $\{\phi_j\}_{j=1}^{n_u}$ 和 $\{\psi_k\}_{k=1}^{n_p}$ 下的系数。这里的 \mathbf{A} 是离散的向量形式的刚度矩阵:

$$\mathbf{A} = [a_{ij}], \quad a_{ij} = \int_{\Omega} \nabla \phi_i : \nabla \phi_j, \quad i, j = 1, \dots, n_u, \quad (2.25)$$

并且 Q 是压力质量矩阵:

$$Q = [q_{kl}], \quad q_{kl} = \int_{\Omega} \psi_k \psi_l, \quad k, l = 1, \dots, n_p. \quad (2.26)$$

我们用标准的分量形式的向量基函数, 在二维问题情形中

$$\langle A\mathbf{v}, \mathbf{v} \rangle \equiv \langle A\mathbf{v}_x, \mathbf{v}_x \rangle + \langle A\mathbf{v}_y, \mathbf{v}_y \rangle, \quad (2.27)$$

其中 A 是 $n \times n$ 的拉普拉斯矩阵。我们 $B\mathbf{A}^{-1}B^T$ 为压力 Schur 补矩阵。我们考虑 $Q^{-1}B\mathbf{A}^{-1}B^T$ 的特征值的范围。我们先考虑边界全部是 Dirichlet 边界的流体问题, 我们先给出一些定义。

定义 2.1. 对于一系列的三角网格 $\{\mathcal{T}_h\}$, 如果存在一个最小的角度 $\theta_* > 0$, 使得 \mathcal{T}_h 中的所有单元满足 $\theta_T > \theta_*$, 那么 \mathcal{T}_h 称作是形正则的。

命题 2.1. 对于 \mathbb{R}^2 空间中的具有形正则性质的一个剖分, \mathbf{P}_1 是建立在这个剖分上的一个近似, 那么质量矩阵 Q 以如下的形式近似数量单位矩阵:

$$ch^2 \leq \frac{\langle Q\mathbf{v}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \leq Ch^2, \quad (2.28)$$

对所有的 $\mathbf{v} \in \mathbb{R}^n$ 。这里 $\underline{h} = \min_{\Delta_k \in \mathcal{T}_h} h_k$ 并且 $h = \max_{\Delta_k \in \mathcal{T}_h} h_k$ 。常数 c 和 C 都是跟 \underline{h} 和 h 无关的。

定义 2.2 (拟一致剖分). 对于一系列的三角网格 $\{\mathcal{T}_h\}$, 如果存在着一个常数 $\rho > 0$ 使得对于 $\{\mathcal{T}_h\}$ 中的每一个网格, 都有 $\underline{h} \geq \rho h$, 那么我们就称 $\{\mathcal{T}_h\}$ 是拟一致的。

对于 \mathbb{R}^2 空间的一个拟一致剖分, 并且它的单元都是形正则的, 那么界限 (2.28) 可以简化为:

$$ch^2 \leq \frac{\langle Q\mathbf{v}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \leq Ch^2, \quad \text{对所有的 } \mathbf{v} \in \mathbb{R}^n. \quad (2.29)$$

(2.29) 是跟空间的维数有关的。对于 \mathbb{R}^3 中的一个区域, 进行拟一致剖分, 则这个剖分上的所有四面体单元, 对应有界:

$$ch^3 \leq \frac{\langle Q\mathbf{v}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \leq Ch^3, \quad \text{对所有的 } \mathbf{v} \in \mathbb{R}^n. \quad (2.30)$$

注 2.1. 如果剖分是拟一致的, 那么对于任意近似 $\mathbf{P}_m, m \geq 2$, 有界性 (2.29) 都成立, 但是 c 和 C 都跟 m 有关。

定理 2.4. 对于 $\partial\Omega = \partial\Omega_D$ 的流体问题, 用满足 LBB 条件的混合元近似并且 \mathbb{R}^2 上的拟一致剖分是形正则的, 则压力 Schur 补矩阵 $B\mathbf{A}^{-1}B^T$ 谱等价于压力质量矩阵 Q :

$$\gamma^2 \leq \frac{\langle B\mathbf{A}^{-1}B^T\mathbf{q}, \mathbf{q} \rangle}{\langle Q\mathbf{q}, \mathbf{q} \rangle} \leq 1 \quad \text{对所有的 } \mathbf{q} \in \mathbb{R}^{n_p} \text{ 使得 } \mathbf{q} \neq \mathbf{1}. \quad (2.31)$$

inf-sup 常量 γ 非 0, 并且跟网格尺度 h 无关。并且压力 Schur 补矩阵的条件数满足

$$\kappa(B\mathbf{A}^{-1}B^T) \leq C/(c\gamma^2), \quad (2.32)$$

其中 c 和 C 是 (2.29) 中的常数。

证明. 我们先来证明下界。离散 inf-sup 稳定性条件的代数表示:

$$\gamma \leq \min_{q_h \neq \text{constant}} \max_{v_h \neq \mathbf{0}} \frac{|(q_h, \nabla \cdot v_h)|}{\|\nabla v_h\| \|q_h\|}. \quad (2.33)$$

特别的, 运用 (2.24) 中的矩阵范数, 并且有 $|(q_h, \nabla \cdot v_h)| = (\mathbf{q}, B\mathbf{v})$, 我们有:

$$\begin{aligned} \gamma &\leq \min_{q \neq \mathbf{1}} \max_{v \neq \mathbf{0}} \frac{|\langle \mathbf{q}, B\mathbf{v} \rangle|}{\langle \mathbf{A}\mathbf{v}, \mathbf{v} \rangle^{1/2} \langle Q\mathbf{q}, \mathbf{q} \rangle^{1/2}} \\ &= \min_{q \neq \mathbf{1}} \frac{1}{\langle Q\mathbf{q}, \mathbf{q} \rangle^{1/2}} \max_{w = \mathbf{A}^{1/2}v \neq \mathbf{0}} \frac{|\langle \mathbf{q}, B\mathbf{A}^{-1/2}\mathbf{w} \rangle|}{\langle \mathbf{w}, \mathbf{w} \rangle^{1/2}} \\ &= \min_{q \neq \mathbf{1}} \frac{1}{\langle Q\mathbf{q}, \mathbf{q} \rangle^{1/2}} \max_{w \neq \mathbf{0}} \frac{|\langle \mathbf{A}^{-1/2}B^T q, \mathbf{w} \rangle|}{\langle \mathbf{w}, \mathbf{w} \rangle^{1/2}} \\ &= \min_{q \neq \mathbf{1}} \frac{\langle \mathbf{A}^{-1/2}B^T q, \mathbf{A}^{-1/2}B^T q \rangle^{1/2}}{\langle Q\mathbf{q}, \mathbf{q} \rangle^{1/2}} \\ &= \min_{q \neq \mathbf{1}} \frac{\langle B\mathbf{A}^{-1}B^T\mathbf{q}, \mathbf{q} \rangle^{1/2}}{\langle Q\mathbf{q}, \mathbf{q} \rangle^{1/2}}. \end{aligned} \quad (2.34)$$

因为它的极值在 $\mathbf{w} = \pm \mathbf{A}^{-1/2}B^T\mathbf{q}$ 时取到。因此, 我们可以看到

$$\gamma^2 = \min_{q \neq \mathbf{1}} \frac{\langle B\mathbf{A}^{-1}B^T\mathbf{q}, \mathbf{q} \rangle}{\langle Q\mathbf{q}, \mathbf{q} \rangle}, \quad (2.35)$$

则下界得证。现在我们来证明上界, 上界是由微分算子的性质得到的。首先, 应用 Cauchy-Schwarz 不等式, 我们有

$$|(q_h, \nabla \cdot v_h)| \leq \|q_h\| \|\nabla \cdot v_h\|. \quad (2.36)$$

第二，注意到 $\mathbf{v}_h \in \mathcal{H}_0^1$ （因为 $\partial\Omega = \partial\Omega_D$ ），将 \mathbf{v}_h 带入到下列等式中：

$$\int_{\Omega} \nabla \mathbf{w} : \mathbf{v} \equiv \int_{\Omega} (\nabla \cdot \mathbf{w})(\nabla \cdot \mathbf{v}) + \int_{\Omega} (\nabla \times \mathbf{w}) : (\nabla \times \mathbf{v}), \quad (2.37)$$

其中这个等式是对任意属于 Sobolev 空间

$$\mathcal{H}_0^1 = \{\mathbf{u} \in \mathcal{H}^1(\Omega) | \mathbf{u} = \mathbf{0} \text{ 在 } \partial\Omega \text{ 上}\}, \quad (2.38)$$

中的任意函数 \mathbf{w}, \mathbf{v} 都成立的，因此我们有

$$\|\nabla \mathbf{v}_h\|^2 = \|\nabla \cdot \mathbf{v}_h\|^2 + \|\nabla \times \mathbf{v}_h\|^2 \geq \|\nabla \cdot \mathbf{v}_h\|^2. \quad (2.39)$$

最后，结合 (2.36) 和 (2.39)，我们得到

$$\frac{|(q_h, \nabla \cdot \mathbf{v}_h)|}{\|\nabla \mathbf{v}_h\| \|q_h\|} \leq 1. \quad (2.40)$$

将上式展开成矩阵形式，如推导 (2.35) 中类似，可以得到 (2.31) 中的上界。运用 (2.29) 中的质量矩阵条件数的有界性，我们可以得到 Schur 补矩阵的条件数有界

$$\gamma^2 \frac{\langle Q\mathbf{q}, \mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{q} \rangle} \leq \frac{\langle B\mathbf{A}^{-1}B^T\mathbf{q}, \mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{q} \rangle} \leq \frac{\langle Q\mathbf{q}, \mathbf{q} \rangle}{\langle \mathbf{q}, \mathbf{q} \rangle}. \quad (2.41)$$

□

注 2.2. 在 \mathbb{R}^3 中的拟一致剖分上的四面体单元，如果是稳定的混合元近似，那么 Schur 补矩阵的有界性同样成立。质量矩阵的有界性由 (2.30) 中给出。

同样地，对于边界中包含 Neumann 条件的流体问题，特征值的界在命题 2.2 中给出。

命题 2.2. 对任意 $\int_{\partial\Omega_N} ds \neq 0$ 的流体问题，用一个一致稳定的混合元近似去离散，并且这个混合元近似是建立在 \mathbb{R}^d 上的一个形正则、拟一致的剖分上，那么压力 Schur 补矩阵满足

$$\gamma_N^2 \leq \frac{\langle B\mathbf{A}^{-1}B^T\mathbf{q}, \mathbf{q} \rangle}{\langle Q\mathbf{q}, \mathbf{q} \rangle} \leq \Gamma^2, \quad \text{对所有的 } \mathbf{q} \in \mathbb{R}^{n_p}. \quad (2.42)$$

其中常数 γ_N 是不为 0 的有界常数，并且跟网格尺度 h 无关， $\Gamma < \sqrt{d}$ d 是空间维数。

对于 $P_2 - P_1, Q_2 - Q_1$ 混合元近似，是满足 inf-sup 条件的，即存在一个不为 0 的 inf-sup 常量 δ ，使得

$$\delta \leq \min_{q_h \neq \text{constant}} \max_{\mathbf{v}_h \neq \mathbf{0}} \frac{|(q_h, \nabla \cdot \mathbf{v}_h)|}{\|\mathbf{v}_h\| \|\nabla q_h\|}. \quad (2.43)$$

我们是在 $L^2(\Omega)$ 空间中估计速度，在 $\mathcal{H}^1(\Omega)$ 中估计压力，这跟一般的 inf-sup 条件有些相反。我们先来定义速度质量矩阵 Q ，

$$Q = [q_{ij}], \quad q_{ij} = \int_{\Omega} \boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_j, \quad i, j = 1, \dots, n_u, \quad (2.44)$$

和压力的刚度矩阵 A_p ，

$$A_p = [a_{kl}], \quad a_{jl} = \int_{\Omega} \nabla \psi_k \cdot \psi_l, \quad i, j = 1, \dots, n_p. \quad (2.45)$$

那么按照 (2.35) 的推导，则 (2.43) 可以重新写为

$$\delta^2 = \min_{\mathbf{q} \neq \mathbf{1}} \frac{\langle BQ^{-1}B^T \mathbf{q}, \mathbf{q} \rangle}{\langle A_p \mathbf{q}, \mathbf{q} \rangle}. \quad (2.46)$$

矩阵 $BQ^{-1}B^T$ 通常被叫做一致压力泊松矩阵，详见 [42]。对于封闭流体的边界条件，这个矩阵是奇异的，其他情况下，是不奇异的。计算 $P_2 - P_1$ 近似的最大最小特征值，我们发现在均匀网格上，并且对封闭流体的边界条件，有

$$\frac{1}{2} < \frac{\langle BQ^{-1}B^T \mathbf{q}, \mathbf{q} \rangle}{\langle A_p \mathbf{q}, \mathbf{q} \rangle} \leq 1, \quad \text{对所有的 } \mathbf{q} \in \mathbb{R}^{n_p}, \mathbf{q} \neq \mathbf{1}. \quad (2.47)$$

我们考虑 (2.22) 的块对角预处理矩阵

$$M = \begin{bmatrix} \mathbf{P} & 0 \\ 0 & T \end{bmatrix}, \quad (2.48)$$

其中 $\mathbf{P} \in \mathbb{R}^{n_u \times n_u}$ 和 $T \in \mathbb{R}^{n_p \times n_p}$ 都是对称正定矩阵。我们要到 (2.22) 的一个好的近似，意味着我们要找到合适的 \mathbf{P} 近似 \mathbf{A} 并且近似 Schur 补 $B\mathbf{A}^{-1}B^T$ 的矩阵 T 。从定理 2.4 我们知道，压力的质量矩阵 Q 在合适的边界条件下是谱等价于 Schur 补矩阵的。因此我们可以选择 $T = Q$ ，并且先令 $\mathbf{P} = \mathbf{A}$ 。我们考虑广义的特征值问题

$$\begin{bmatrix} \mathbf{A} & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{A} & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix}, \quad (2.49)$$

通过简单的计算可以知道特征值 λ 位于

$$\left[\frac{1 - \sqrt{1 + 4\Gamma^2}}{2}, \frac{1 - \sqrt{1 + 4\gamma^2}}{2} \right] \cup \{1\} \cup \left[\frac{1 + \sqrt{1 + 4\gamma^2}}{2}, \frac{1 + \sqrt{1 + 4\Gamma^2}}{2} \right]. \quad (2.50)$$

其中 γ 和 Γ 是 (2.31) 和 (2.42) 中的上下界，跟网格尺度 h 无关的常量。因此，我们可以知道，如果用预处理的 MINRES 来求解的话，会很快速收敛，并且与离散的网格尺度没有关系。

在 [38] 中指出，在不影响 MINRES 收敛的前提下，可以考虑用比 $T = \text{diag}(Q)$ 来作为 Schur 补矩阵的近似。并给出了 $T = Q$ 和 $T = \text{diag}(Q)$ 的有界性

引理 2.1. 如果在 \mathbb{R} 上的一个剖分上，压力空间用 P_1 元近似，选取 $T = \text{diag}(Q)$ ，那么有

$$\theta^2 \leq \frac{\langle Q\mathbf{q}, \mathbf{q} \rangle}{\langle T\mathbf{q}, \mathbf{q} \rangle} \leq \Theta^2 \quad \text{对所有的 } \mathbf{q} \in \mathbb{R}^{n_p}, \quad (2.51)$$

成立，其中 $\theta = 1/\sqrt{2}$ 和 $\Theta = \sqrt{2}$ ，都与网格尺度 h 无关。

下面考虑如何更好的近似 \mathbf{A} 。我们首先考虑泊松问题变分问题：找到 $u \in \mathcal{H}_E^1$ ，使得

$$a(u, v) = l(v) \quad \text{对所有的 } v \in \mathcal{H}_{E_0}^1. \quad (2.52)$$

命题 2.3 (\mathcal{H} 正则性). 如果存在一个常数 C_Ω 使得对任意的 $f \in L_2(\Omega)$ ，都存在一个解 $u \in \mathcal{H}_E^1 \subset \mathcal{H}^2(\Omega)$ ，并且满足

$$\|u\|_{\mathcal{H}^2} \leq C_\Omega \|f\|, \quad (2.53)$$

那么则称变分问题 (2.52) 是 \mathcal{H}^2 正则的。

定理 2.5. 给定一个 \mathcal{H}^2 的问题和一个 \mathcal{R}^2 或 \mathcal{R}^3 上网格尺度为 h 的拟一致剖分，并且用一个 l 层多重网格 W 循环来求解 $A\mathbf{u} = \mathbf{f}$ 。那么存在一个最小的光滑步数 k_* ，它与 h 无关，并且存在一个有界的不为 1 的压缩因子 η_l ，也是与 h 无关的，那么我们有

$$\|\mathbf{u} - \mathbf{u}^{(i+1)}\|_A \leq \eta_l \|\mathbf{u} - \mathbf{u}^{(i)}\|_A. \quad (2.54)$$

其中 $\mathbf{u}^{(i)}$ 是 l 层多重网格迭代经过 i 次循环得到的。

定理 2.5 的证明见 [38]。

定理 2.6. 给定一个 \mathcal{H}^2 正则的问题和一个 \mathcal{R}^2 或 \mathcal{R}^3 上网格尺度为 h 的拟一致剖分，并且用一个对称的 l 层多重网格 V 循环来求解 $A\mathbf{u} = \mathbf{f}$ 。那么这个 l 层多重网格是用阻尼的雅克比光滑子，则存在一个压缩因子 $\eta_l < 1$ 使得

$$\|\mathbf{u} - \mathbf{u}^{(i+1)}\|_A \leq \rho_l \|\mathbf{u} - \mathbf{u}^{(i)}\|_A, \quad \rho_l \leq \rho_\infty := C/(C + k), \quad (2.55)$$

其中 C 是一个与 l 无关的常数，并且 $k = m$ 是光滑次数。

定理 2.6 的证明可以参考([15]、[16])。

引理 2.2. 在求解线性系统 $A\mathbf{u} = \mathbf{f}$ 中，不动点迭代

$$\mathbf{u}^{(i+1)} = (\mathbf{I} - \mathbf{P}^{-1}\mathbf{A})\mathbf{u}^i + \mathbf{P}^{-1}\mathbf{f} \quad (2.56)$$

的迭代误差满足

$$\|\mathbf{u} - \mathbf{u}^{(i+1)}\|_A \leq \rho \|\mathbf{u} - \mathbf{u}^{(i)}\|_A, \quad (2.57)$$

其中压缩比例 $\rho < 1$ ，那么

$$1 - \rho \leq \frac{\langle \mathbf{A}\mathbf{v}, \mathbf{v} \rangle}{\langle \mathbf{P}\mathbf{v}, \mathbf{v} \rangle} \leq 1 + \rho \quad \text{对所有的 } \mathbf{v}. \quad (2.58)$$

证明. 因为精确解 \mathbf{u} 满足

$$\mathbf{u} = (\mathbf{I} - \mathbf{P}^{-1}\mathbf{A})\mathbf{u} + \mathbf{P}^{-1}\mathbf{f},$$

(2.57) 可以表述为

$$\langle \mathbf{A}(\mathbf{I} - \mathbf{P}^{-1}\mathbf{A})\mathbf{e}, (\mathbf{I} - \mathbf{P}^{-1}\mathbf{A})\mathbf{e} \rangle \leq \rho^2 \langle \mathbf{A}\mathbf{e}, \mathbf{e} \rangle,$$

对任意的 $\mathbf{e} = \mathbf{u} - \mathbf{u}^{(i)}$ ，令 $\mathbf{v} = \mathbf{A}^{1/2}\mathbf{e}$ ，那么上式可以重新写成

$$\langle (\mathbf{I} - \mathbf{A}^{1/2}\mathbf{P}^{-1}\mathbf{A}^{1/2})\mathbf{v}, (\mathbf{I} - \mathbf{A}^{1/2}\mathbf{P}^{-1}\mathbf{A}^{1/2})\mathbf{v} \rangle \leq \rho^2 \langle \mathbf{v}, \mathbf{v} \rangle,$$

对所有的 \mathbf{v} 。因此

$$-\rho \langle \mathbf{v}, \mathbf{v} \rangle \leq \langle (\mathbf{I} - \mathbf{A}^{1/2}\mathbf{P}^{-1}\mathbf{A}^{1/2})\mathbf{v}, \mathbf{v} \rangle \leq \rho \langle \mathbf{v}, \mathbf{v} \rangle,$$

则我们可以得到

$$1 - \rho \leq \frac{\langle \mathbf{A}\mathbf{v}, \mathbf{v} \rangle}{\langle \mathbf{P}\mathbf{v}, \mathbf{v} \rangle} \leq 1 + \rho, \quad (2.59)$$

对所有的 \mathbf{v} 。 \square

结合定理 2.6, 定理 2.5 和引理 2.2, 可以选取合适的压缩比例因子, 用一个多重网格循环来代替 \mathbf{P} 。如果在预处理中我们选择 $\mathbf{P} = \mathbf{A}$, 并且用一次多重网格 V 型或 W 型循环代替 P , 虽然 MINRES 迭代次数会额外增加一些, 但是是与 h 无关的。但是如果用对角块直接做预处理矩阵, MINRES 的迭代步数会随着 h 的减小而增加。

2.4.2 针对Navier-Stokes方程

用稳定的混合元近似 Navier-Stokes 方程, 会得到与 Stokes 方程中类似的线性系统, 我们考虑它的系数矩阵:

$$\begin{bmatrix} \mathbf{F} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix}, \quad (2.60)$$

其中 \mathbf{F} 包含对流项矩阵, 不是对称的。我们可以用 Krylov 子空间方法中的 GMRES 来求解线性系统。在本文中, 我们仅考虑 Ossen 系统, 即

$$\begin{bmatrix} \nu\mathbf{A} + \mathbf{N} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}, \quad (2.61)$$

其中 \mathbf{A} 是速度空间的向量形式的刚度矩阵, 以及 \mathbf{N} 是向量形式的对流矩阵, 分别定义如下:

$$\mathbf{A} = [\mathbf{a}_{ij}], \quad \mathbf{a}_{ij} = \int_{\Omega} \nabla \phi_i \cdot \phi_j, \quad (2.62)$$

$$\mathbf{N} = [\mathbf{n}_{ij}], \quad \mathbf{n}_{ij} = \int_{\Omega} (\mathbf{u}_h \cdot \phi_i) \cdot \phi_j. \quad (2.63)$$

Elman 在 [38] 中指出, 在求解 (2.61) 时, 用块三角矩阵做预处理比块对角矩阵做预处理, 迭代次数会减半, 详见([40]、[36])。我们选择用块三角矩阵

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_F & \mathbf{B}^T \\ 0 & -\mathbf{M}_S \end{bmatrix}, \quad (2.64)$$

作为预处理矩阵, 与对角块预处理矩阵相比, 只增加了矩阵 \mathbf{B}^T 与向量乘积的运算量。我们先假设 $\mathbf{M}_F = \mathbf{F}$, 对于这一块矩阵的求解, 可以用代数多重网格求解。代数多重网格内容可以参考著作([17]、[60])。下面我们着重来找到 Schur 补预处理矩阵 \mathbf{M}_S 有效的选择。

注 2.3. 我们可以类似 Stokes 方程, 用压力的质量矩阵 Q 作为 Schur 补预处理矩阵。在([36]、[51]) 中, 指出对雷诺数 $Re < 10$ 的情况, 迭代收敛率是跟网格无关的, 但是随着 Reynolds 的增大, 收敛率会恶化。

下面我们考虑两种近似 Schur 补矩阵 $B\mathbf{F}^{-1}B^T$ 的方法, 压力对流扩散预处理(PCD precondition) 和最小二乘交换子预处理(LSC precondition)。我们先引入速度空间和压力空间的对流扩散算子

$$\begin{aligned}\mathcal{L} &= -\nu \nabla^2 + \mathbf{w}_h \cdot \nabla, \\ \mathcal{L}_p &= -(\nu \nabla^2 + \mathbf{w}_h \cdot \nabla)_p.\end{aligned}$$

我们考虑对流扩散算子和散度算子的交换子:

$$\mathcal{E} = \nabla \cdot (-\nu \nabla^2 + \mathbf{w}_h \cdot \nabla) - (\nu \nabla^2 + \mathbf{w}_h \cdot \nabla)_p \nabla \cdot, \quad (2.65)$$

这个交换子在某种意义上是比较小的。我们先来看压力对流扩散预处理(PCD precondition)。令 $Q^{-1}B$ 是负的散度算子的矩阵表现形式, $\mathbf{Q}^{-1}B^T$ 是离散梯度算子的矩阵表现形式。则 (2.65) 可以写成离散矩阵形式

$$\mathcal{E}_h = (Q^{-1}B)(\mathbf{Q}^{-1}\mathbf{F}) - (Q^{-1}F_p)(Q^{-1}B), \quad (2.66)$$

其中 \mathbf{Q} 是速度质量矩阵, \mathbf{F} 和 F_p 分别是速度和压力空间的对流扩散矩阵。

F_p 为:

$$F_p = [f_{p,ij}], f_{p,ij} = \nu \int_{\Omega} \nabla \psi_j \cdot \nabla \psi_i + \int_{\Omega} (\mathbf{w}_h \cdot \nabla \psi_j) \psi_i. \quad (2.67)$$

对 (2.66) 两边乘上 $\mathbf{F}^{-1}B^T$, 我们可以得到 Schur 补矩阵的一个近似:

$$B\mathbf{F}^{-1}B^T \approx QF_p^{-1}B\mathbf{Q}^{-1}B^T. \quad (2.68)$$

从 (2.46) 和 (2.47) 中我们得知, $B\mathbf{Q}^{-1}B^T$ 是跟压力空间的刚度矩阵是谱等价的。因此我们可以得到压力对流扩散预处理: 把 (2.64) 中的 M_S 取为

$$M_S := QF_p^{-1}A_p. \quad (2.69)$$

预处理方法需要实现矩阵求逆的操作

$$M_S^{-1} = A_p^{-1}F_pQ^{-1}. \quad (2.70)$$

在这个过程中包括求解一个泊松问题，一个质量矩阵求解还有一个矩阵向量乘法。在封闭流体(如方腔流)问题中，压力刚度矩阵 A_p 是跟 $B\mathbf{Q}^{-1}B^T$ 是谱等价的。在求解入流/出流问题时，对压力空间刚度矩阵 A_p 和压力对流扩散矩阵 F_p 不做任何处理，是 [50] 中的方法。[39] 指出，在有入流/出流边界的问题中，需要对压力空间刚度矩阵 A_p 和压力对流扩散矩阵 F_p 做边界条件处理，来提高计算效率。不失一般性地令 $\mathbf{s} = \mathbf{0}$ ，假设压力空间 M^h 是 C^0 的，则 (2.10) 可以写成

$$\int_{\Omega} \nabla \mathbf{u}_h : \nabla \mathbf{v}_h + \int_{\Omega} \nabla \cdot \mathbf{v}_h = \int_{\Omega_N} p_h (\mathbf{v}_h \cdot \mathbf{n}), \quad \text{对所有的 } \mathbf{v}_h \in X_0^h. \quad (2.71)$$

对比 (2.71) 和 (2.10)，要使弱形式有相同的解，需要使

$$p_h|_{\partial\Omega_N} = 0. \quad (2.72)$$

下面我们来证明 (2.72)，我们取测试函数 $\mathbf{v}_h = \boldsymbol{\phi}_l$ ，则压力梯度算子的弱形式为

$$\int_{\Omega} \nabla p_h \cdot \boldsymbol{\phi}_l = \sum_{m=1}^{n_p} \left(\int_{\Omega} \nabla \psi_m \cdot \boldsymbol{\phi}_l \right) \mathbf{p}_m = \sum_{m=1}^{n_p} G_{lm} \mathbf{p}_m = [G\mathbf{p}]_l, \quad (2.73)$$

其中 G 是一个离散的梯度算子。利用分部积分可得

$$\begin{aligned} [G\mathbf{p}]_l &= - \int_{\Omega} p_h \nabla \cdot \boldsymbol{\phi}_l + \int_{\Omega_N} p_h (\boldsymbol{\phi}_l \cdot \mathbf{n}) \\ &= \sum_{m=1}^{n_p} \left(- \int_{\Omega} \psi_m \nabla \cdot \boldsymbol{\phi}_l \right) \mathbf{p}_m + \sum_{m=1}^{n_p} \left(\int_{\Omega_N} \psi_m (\boldsymbol{\phi}_l \cdot \mathbf{n}) \right) \mathbf{p}_m \\ &= [B^T \mathbf{p}]_l + [R\mathbf{p}]_l, \end{aligned}$$

其中 $R = [r_{lm}]$, $r_{lm} = \int_{\partial\Omega_N} \psi_m (\boldsymbol{\phi}_l \cdot \mathbf{n})$ 要使 $[G\mathbf{p}]_l = \int_{\Omega} p_h \nabla \cdot \boldsymbol{\phi}_l = [B^T \mathbf{p}]_l$, 当且仅当 $[R\mathbf{p}]_l = 0$ 。即有 (2.72) 成立。

在 $\partial\Omega_D$ 上，压力需要满足 Neumann 条件 $\frac{\partial p_h}{\partial n} = 0$ 。下面我们来验证。我们考虑压力拉普拉斯算子

$$[A_p \mathbf{p}]_i = \sum_{j=1}^{n_p} [A_p]_{ij} \mathbf{p}_j = \sum_{j=1}^{n_p} \left(\int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j \right) \mathbf{p}_j = \int_{\Omega} \nabla p_h \cdot \nabla \psi_i. \quad (2.74)$$

我们对上式右端利用分部积分可得

$$[A_p \mathbf{p}]_i = \int_{\Omega} \nabla p_h \cdot \nabla \psi_i = \int_{\Omega} (-\nabla^2 p_h) \psi_i - \int_{\partial\Omega} (\nabla p_h \cdot \mathbf{n}) \psi_i. \quad (2.75)$$

对比 (2.74) 和 (2.75)，要使两者一致，需要 (2.75) 中边界积分为 0。因此需要满足

$$\frac{\partial p_h}{\partial n}|_{\partial\Omega_D} = 0. \quad (2.76)$$

对于封闭流体问题，在全部边界上都有 $\frac{\partial p_h}{\partial n}$ 成立。

压力对流扩散矩阵 F_p 需要在边界上施加 Robin 条件

$$-\nu \frac{\partial p_h}{\partial n} + (\mathbf{w}_h \cdot \mathbf{n})p_h = 0. \quad (2.77)$$

验证过程详见([38]，第九章)。

另外一种预处理方法是最小二乘交换子预处理(LSC preconditioning)，我们做如下简要介绍：

$$M_S = (B\mathbf{T}^{-1}B^T)(B\mathbf{T}^{-1}\mathbf{F}\mathbf{T}^{-1}B^T)^{-1}(B\mathbf{T}^{-1}B^T), \quad (2.78)$$

其中 $\mathbf{T} = \text{diag}(\mathbf{Q})$ 。结合块三角预处理矩阵 (2.64)，就得到了最小二乘交换子的预处理方法。实施这种预处理方法，需要对 M_S 完成求逆操作：

$$M_S^{-1} = (B\mathbf{T}^{-1}B^T)^{-1}(B\mathbf{T}^{-1}\mathbf{F}\mathbf{T}^{-1}B^T)(B\mathbf{T}^{-1}B^T)^{-1}, \quad (2.79)$$

其中包括，两个泊松问题的求解，以及关于 $B, B^T, \mathbf{F}, \mathbf{T}$ 的矩阵向量乘法。对比这两种预处理方法，PCD 预处理需要额外的生成两个矩阵 A_p 和 F_p 。LSC 预处理不需要额外生成矩阵，但是在每步中需要两次泊松求解。矩阵 $B\mathbf{T}^{-1}B^T$ 在网格移动后需要从新生成，这其中包含矩阵乘法，是比较耗时的，我们还是倾向使用 PCD 预处理。

第 3 章 基于 $4P_1 - P_1$ 元求解 Navier-Stokes 方程 的移动网格有限元方法

根据上一章，我们知道混合元是求解不可压 Navier-Stokes 方程的一种重要方法。为了保证解的存在唯一性，需要使速度和压力两个解所在的空间满足 LBB 条件。其中一种办法是使得速度空间相对压力空间来说，自由度足够多，例如 Mini 元，Taylor-Hood 元等等。另外一种方法是对压力空间施加一些约束，比如说稳定化的 $P_1 - P_1$ 元， $P_1 - P_0$ 元。为了减少计算量，提高计算效率，通常会运用自适应网格方法。然而，不满足 LBB 条件的有限元对应用到自适应网格方法上有一定的技术困难。综合以上考虑，我们选取稳定的 $P_1isoP_2P_1$ 元，它自然满足 LBB 条件，详细见 [9]。在 $P_1isoP_2P_1$ 元中，速度单元所在的网格可以有压单元的网格加密一次得到，由于速度单元网格上的 3 个基函数不在同一个速度单元上，因此即拼装压力单元和速度单元时有一定技术上的困难。

在本文的工作中，我们选取 $4P_1 - P_1$ 元，它与 $P_1isoP_2P_1$ 拥有相同的网格结构，自然也满足 inf-sup 条件，但需要指出的是 $P_1isoP_2P_1$ 元的速度单元上的基函数都局部的位于相同的速度单元内，并且全部是线性元。这给我们拼装散度块矩阵提供了便利：只要我们建立四个速度单元和大的压力单元之间的索引。索引的建立又依赖于两层网格的数据结构，用我们上一章提到的几何遗传树结构，可以建立两层网格间的对应关系。

基于这种几何遗传树的结构，我们可以很容易的建立四个小的速度单元与压力宏单元间的索引。同时在 [53] 中 h 自适应方法也是基于这种树结构。自适应网格可以减少计算量，同时可以研究流体局部小尺度上的现象，比如涡。同时移动网格网格方法也可以应用在 $4P_1 - P_1$ 元上，网格移动我们只移动压力网格，当压力网格移动完，速度网格由压力网格加密一次即可得到。但是需要注意的是，求解不可压的 Navier-Stokes 方程，在将旧网格的解插值到移动后的网格的过程，要满足散度为 0 的条件，这个工作由 [28] 中给出。

3.1 数据结构

$4P_1 - P_1$ 是基于两套不同的网格和两种有限元空间。速度网格可以由压力网格全局加密一次得到。网格的数据结构是基于 [53] 中的几何遗传树结构，如图 2.4 所示。一个宏压力单元对应这四个速度单元，通过遍历一次所有的速度单元，利用几何遗传树结构，就可以建立速度单元和压力单元的 1 – 1 对应。

在 AFEPack 中，速度网格和压力网格分别存储在两张非正则网格 irregularMeshV 和 irregularMeshP 上，这两张非正则网格都是建立在同一棵树上。非正则网格上，可以进行全局加密，局部加密以及疏化的操作，能进行这种操作得益于它的四叉树结构。所有的网格节点从根节点到叶子节点，都存储在非正则网格中。这里 irregularMeshV 是由 irregularMeshP 进行全局一次加密得到的，根据非正则网格的树结构，我们可以通过遍历 irregularMeshV 的全部活动单元(即叶子节点)，通过活动单元来找到它的父亲单元(即叶子节点的父亲节点)。这样我们建立了从速度网格单元到压力网格单元之间的单向索引。还是根据活动单元的父亲节点也是具有树结构的，因此，我们可以根据父子节点单元(压力单元)来找到它对应的四个儿子单元(即四个速度单元)，这样我们也建立了从压力单元到速度单元的单向索引。到这里，速度和压力间的单元已经建立了 1 – 1 索引，注意到，只要不产生新的网格单元，我们建立的索引不需要重新改。我们只需要一次构建索引就可以，不管网格如何移动，网格间的索引不会改变。建立索引的过程参见算法 3.1。

3.2 混合元近似

3.2.1 流体方程

在 Stokes 方程

$$\begin{aligned} -\nabla^2 \mathbf{u} + \nabla p &= \mathbf{f}, \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned} \tag{3.1}$$

的基础上加一个对流项，再加上时间项，我们可以得到一个时间发展的 Navier-Stokes 方程

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} - \nu \nabla^2 \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p &= \mathbf{f}, \\ \nabla \cdot \mathbf{u} &= 0. \end{aligned} \tag{3.2}$$

Algorithm 3.1 构建速度单元和压力单元间的索引

```
achieveiterator ← irregualerMeshV.beginActiveElement()
enditerator ← irregualerMeshV.endActiveElement()
while achieveiterator ≠ enditerator do
    int index-v-element ← achieveiterator→ index
    HElement < DIM, DIM>* parent ← activeiterator→ parent
    int index-p-element ← parent→ index
    int n-child ← parent→ n-child
    index-p2v[index-p-element].resize(n-child)
    while i ≥ 0 and i < n-child do
        HElement<DIM,DIM> *chi ← parent→ child[i]
        int index-v-element ← child→ index
        index-p2v[index-p-element][i] ← index-v-element
        index-v2p[index-v-element] ← index-p-element
    end while
end while
```

其中 $\nu > 0$ 是一个常数，称作动力学粘性系数。跟 Stokes 方程中类似， \mathbf{u} 表示流体速度， p 表示压力。对流项 $\mathbf{u} \cdot \nabla \mathbf{u} := (\mathbf{u} \cdot \nabla) \mathbf{u}$ ，是非线性项，这也使得 Navier-Stokes 方程的解不唯一，这也给我们的数值计算带来了一定的挑战。系统 (3.2) 的计算区域是 Ω ，可以是二维的或三维的。在边界 $\partial\Omega = \partial\Omega_D \cup \Omega_N$ 上的边界条件如下：

$$\begin{aligned} \mathbf{u} &= \mathbf{w}, && \text{on } \partial\Omega_D, \\ \nu \frac{\partial \mathbf{u}}{\partial n} - p &= \mathbf{0}, && \text{on } \partial\Omega_N, \end{aligned} \quad (3.3)$$

其中边界 $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ ， \mathbf{n} 表示边界的外法向。Dirichlet 边界根据边界上的速度和外法向的乘积，可以细分为：

$$\begin{aligned} \partial\Omega_+ &= x \text{ 在 } \partial\Omega \text{ 上 } |\boldsymbol{\omega} \cdot \mathbf{n}| > 0, && \text{出流边界,} \\ \partial\Omega_0 &= x \text{ 在 } \partial\Omega \text{ 上 } |\boldsymbol{\omega} \cdot \mathbf{n}| = 0, && \text{特征边界,} \\ \partial\Omega_- &= x \text{ 在 } \partial\Omega \text{ 上 } |\boldsymbol{\omega} \cdot \mathbf{n}| < 0, && \text{入流边界.} \end{aligned} \quad (3.4)$$

如果边界全部是 Dirichlet 边界条件，即 $\partial\Omega = \partial\Omega_D$ ，那么 Navier-Stokes 问题 (3.2) 和 (3.3) 的压力解除去一个常数外是唯一的。我们对 (3.2) 中的不可压约

束应用散度定理,

$$0 = \int_{\Omega} \nabla \cdot \mathbf{u} = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} = \int_{\Omega} \boldsymbol{\omega} \cdot \mathbf{n}. \quad (3.5)$$

即边界值要满足相容性条件

$$\int_{\partial\Omega_+} \boldsymbol{\omega} \cdot \mathbf{n} + \int_{\partial\Omega_-} \boldsymbol{\omega} \cdot \mathbf{n} = 0. \quad (3.6)$$

简单的讲, 就是说流入 Ω 的流体的体积, 要与流出的体积相等。这也是压力不唯一的原因。在处理入流/出流的问题时, 要注意保证这个相容性条件, 否则 Navier-Stokes 问题的解有可能不存在。一般情况下, 我们在出流边界设置自然条件, 相容性条件会自然满足, 因此这时候问题 (3.2) 和 (3.3) 的压力解是唯一的。

(3.2) 是 Navier-Stokes 的原始变量形式, 它的另外一组表现形式是流函数形式。如果区域 Ω 是单连通的, 那么不可压缩条件

$$\frac{\partial u_x}{\partial x} = -\frac{\partial u_y}{\partial y}, \quad (3.7)$$

意味着存在一个流函数 $\psi(x, y)$ 使得

$$u_x = \frac{\partial \psi}{\partial y}, \quad u_y = -\frac{\partial \psi}{\partial x}. \quad (3.8)$$

$\psi(x, y)$ 除去一个常数外被 \mathbf{u} 唯一确定。对 (3.2) 中的动量方程取 curl , 我们可以得到流函数形式:

$$-\nu \nabla^4 \psi + \frac{\partial \psi}{\partial y} \frac{\partial}{\partial x} (\nabla^2 \psi) - \frac{\partial \psi}{\partial x} \frac{\partial}{\partial y} (\nabla^2 \psi) = -\text{curl } \mathbf{f}. \quad (3.9)$$

但由于流函数形式 (3.9) 在三维空间没有一般形式, 所以流函数形式只在二维。我们还是研究 (3.2) 中的原始变量形式。

3.2.2 弱形式

我们定义解和检验空间如下:

$$\mathbf{H}_E^1 := \left\{ \mathbf{u} \in \mathcal{H}^1(\Omega)^d \mid \mathbf{u} = \mathbf{w} \text{ on } \partial\Omega_D \right\}, \quad (3.10)$$

$$\mathbf{H}_{E_0}^1 := \left\{ \mathbf{v} \in \mathcal{H}^1(\Omega)^d \mid \mathbf{u} = \mathbf{0} \text{ on } \partial\Omega_D \right\}, \quad (3.11)$$

其中 d 为空间维数。那么变分形式为：寻找 $(\mathbf{u}, p) \in (\mathbf{H}_E^1, L_2(\Omega))$ 使得

$$\int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} + \nu \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} + \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{v} - \int_{\Omega} p (\nabla \cdot \mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}, \quad (3.12)$$

$$\forall \mathbf{v} \in \mathbf{H}_{E_0}^1,$$

$$\int_{\Omega} q (\nabla \cdot \mathbf{u}) = 0, \quad (3.13)$$

$$\forall q \in L_2(\Omega),$$

其中 $\nabla \mathbf{u} : \nabla \mathbf{v} = \nabla u_x \cdot \nabla v_x + \nabla u_y \cdot \nabla v_y$ 。我们在构造混合元近似的时候要确保可解性条件

$$\int_{\Omega} p \nabla \cdot \mathbf{v} = 0, \quad \text{对任意的 } \mathbf{v} \in \mathbf{H}_{E_0}^1. \Rightarrow \begin{cases} p = \text{constant}, & \text{如果 } \partial\Omega = \partial\Omega_D. \\ p = 0, & \text{其他情况.} \end{cases} \quad (3.14)$$

(3.14) 可以由 inf-sup 条件推出。

$$\inf_{q \neq \text{constant}} \sup_{\mathbf{v} \neq 0} \frac{|(q, \nabla \cdot \mathbf{v})|}{\|\mathbf{v}\|_{H(\Omega)^1} \|q\|_{L(\Omega)^0}} \geq \gamma > 0. \quad (3.15)$$

假设 τ_h 是 Ω 上对压力网格的三角剖分，网格尺度 $h = \max_{T \in \tau_h} \text{diam}(T)$ ， T 为三角剖分 τ_h 的单元。对应地， $\tau_{\frac{h}{2}}$ 是对速度网格的三角剖分。基于 $\tau_{\frac{h}{2}}$ 和 τ_h 上的有限元空间 \mathbf{X}_E^h 和 P_h 满足

$$\mathbf{X}_E^h \subset \mathcal{H}_E, \quad P_h \subset L_2(\Omega).$$

那么 (3.12) 和 (3.13) 可以写成如下形式：寻找 $(\mathbf{u}_h, p_h) \in \mathbf{X}_E^h \times P^h$ 使得

$$\begin{aligned} & \int_{\Omega} \frac{\partial \mathbf{u}_h}{\partial t} \cdot \mathbf{v}_h + \nu \int_{\Omega} \nabla \mathbf{u}_h : \nabla \mathbf{v}_h \\ & + \int_{\Omega} (\mathbf{u}_h \cdot \nabla \mathbf{u}_h) \cdot \mathbf{v}_h - \int_{\Omega} p_h (\nabla \cdot \mathbf{v}_h) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_h, \quad \forall \mathbf{v}_h \in \mathbf{X}_0^h, \\ & \int_{\Omega} q_h (\nabla \cdot \mathbf{u}_h) = 0, \quad \forall q_h \in P^h. \end{aligned} \quad (3.16)$$

为了简便，我们时间离散上用显示 Euler 格式： $\forall (\mathbf{v}_h, q_h) \in \mathbf{X}_0^h \times P^h$

$$\begin{aligned} & \int_{\Omega} \frac{\mathbf{u}_h^{(n+1)} - \mathbf{u}_h^{(n)}}{\delta t} \cdot \mathbf{v} + \nu \int_{\Omega} \nabla \mathbf{u}_h^{(n+1)} : \nabla \mathbf{v}_h - \int_{\Omega} p_h^{(n+1)} (\nabla \cdot \mathbf{v}_h) = \int_{\Omega} (\mathbf{f} + \mathbf{u}_h^{(n)} \cdot \nabla \mathbf{u}_h^{(n)}) \cdot \mathbf{v}, \\ & \int_{\Omega} q_h \nabla \cdot \mathbf{u}_h^{(n+1)} = 0. \end{aligned} \quad (3.17)$$

令 $\{\phi_j\}_{j=1}^n$ 和 $\{\psi_k\}_{k=1}^m$ 分别为速度和压力的线性元基函数。则数值解 $\mathbf{u}_h^{(n+1)} = (u_{xh}^{(n+1)}, u_{yh}^{(n+1)}), p_h^{n+1}$ 可以写成如下形式：

$$u_{xh}^{(n+1)} = \sum_{j=1}^n u_j \phi_j, \quad u_{yh}^{(n+1)} = \sum_{j=1}^n v_j \phi_j, \quad p_h^{(n+1)} = \sum_{k=1}^m p_k \psi_k.$$

将 $u_{xh}^{(n+1)}, u_{yh}^{(n+1)}, p_h^{(n+1)}$ 带入离散弱形式 (3.17) 中，可以得到线性方程组

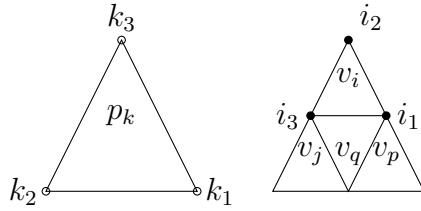
$$\begin{bmatrix} \frac{1}{dt}M + \nu A & 0 & B_x^T \\ 0 & \frac{1}{dt}M + \nu A & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ p \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ g \end{bmatrix}, \quad (3.18)$$

其中 M 是 $n \times n$ 的质量矩阵， A 是拉普拉斯矩阵，由以下形式：

$$\begin{aligned} A &= [a_{ij}], \quad a_{ij} = \int_{\Omega} \nabla \phi_i \cdot \nabla \phi_j; \\ M &= [m_{ij}], \quad m_{ij} = \int_{\Omega} \phi_i \phi_j; \\ B_x^T &= [bx_{ik}^T], \quad bx_{ik}^T = \int_{\Omega} \psi_k \frac{\partial \phi_i}{\partial x}; \\ B_y^T &= [by_{ik}^T], \quad by_{ik}^T = \int_{\Omega} \psi_k \frac{\partial \phi_i}{\partial y}; \\ f_x &= [f_i], \quad f_i = \int_{\Omega} \left(\frac{u_{xh}^{(n)}}{dt} - (u_{xh}^{(n)}, \frac{\partial u_{xh}^{(n)}}{\partial x} + u_{yh}^{(n)} \frac{\partial u_{xh}^{(n)}}{\partial y}) \right) \phi_i; \\ f_y &= [f_i], \quad f_i = \int_{\Omega} \left(\frac{u_{yh}^{(n)}}{dt} - (u_{xh}^{(n)} \frac{\partial u_{yh}^{(n)}}{\partial x} + u_{yh}^{(n)} \frac{\partial u_{yh}^{(n)}}{\partial y}) \right) \phi_i; \\ g &= [g_k], \quad g_i = 0. \end{aligned}$$

接下来我们着重解释一下散度块矩阵 $B_x B_y, B_x^T, B_y^T$ 的拼装。以 B_x^T 为例，令 Δ_{v_i} 为一个速度单元，我们可以通过上一节建立的单元素引本来找到对应的压力单元 Δ_{p_k} (如图 3.1)。 $\phi_{i_1}, \phi_{i_2}, \phi_{i_3}$ 是定义在速度单元 Δ_{v_i} 上的线性元基函数，下标 i_1, i_2, i_3 表示该顶点上的自由度在速度全部自由度的全局编号。同时 $\psi_{k_1}, \psi_{k_2}, \psi_{k_3}$ 是定义在压力单元 Δ_{p_k} 上的线性元基函数， k_1, k_2, k_3 代表单元 Δ_{p_k} 上的自由度在压力全部自由度中的全局编号。因此单元 Δ_{p_k} 和 Δ_{v_i} 对 B_x^T 的贡献为

$$\begin{bmatrix} \int_{\Delta_{v_i}} \psi_{k_1} \frac{\partial \phi_{i_1}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_2} \frac{\partial \phi_{i_1}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_3} \frac{\partial \phi_{i_1}}{\partial x} \\ \int_{\Delta_{v_i}} \psi_{k_1} \frac{\partial \phi_{i_2}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_2} \frac{\partial \phi_{i_2}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_3} \frac{\partial \phi_{i_2}}{\partial x} \\ \int_{\Delta_{v_i}} \psi_{k_1} \frac{\partial \phi_{i_3}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_2} \frac{\partial \phi_{i_3}}{\partial x} & \int_{\Delta_{v_i}} \psi_{k_3} \frac{\partial \phi_{i_3}}{\partial x} \end{bmatrix}. \quad (3.19)$$


 图 3.1: 左: p 单元; 右: 与压力单元对应的 4 个速度 v 单元

将贡献矩阵 (3.19) 加到 B_x^T 中的相应位置 (i_1, k_1) 、 (i_1, k_2) 、 (i_1, k_3) 、 (i_2, k_1) 、 (i_2, k_2) 、 (i_2, k_3) 、 (i_3, k_1) 、 (i_3, k_2) 、 (i_3, k_3) 上。通过遍历所有的速度单元，重复上面的操作， B_x^T 拼装完成。 B_y^T 也可以同样拼装。

B_x 和 B_y 的拼装是遍历所有的压力单元，通过单元素索引，找到对应的四个速度单元，然后分别用压力单元和四个速度单元进行拼装。过程相似，便不再赘述。

3.3 预处理策略

在我们的求解 Navier-Stokes 方程的算法中，我们用稳态的 Stokes 方程的解作为初始值，因此在这一部分中，我们考虑针对 Stokes 方程的预处理方法。我们写出 (3.1) 的 $4P_1 - P_1$ 近似的离散弱形式：寻找 $(\mathbf{u}_h, p_h) \in X_E^h \times P^h$ 使得

$$\begin{aligned} \nu \int_{\Omega} \nabla \mathbf{u}_h : \nabla \mathbf{v}_h - \int_{\Omega} p_h (\nabla \cdot \mathbf{v}_h) &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v}_h, \quad \forall \mathbf{v}_h \in \mathbf{X}_0^h, \\ \int_{\Omega} q_h (\nabla \cdot \mathbf{u}_h) &= 0, \quad \forall q_h \in P^h. \end{aligned} \quad (3.20)$$

同样的，令 $\{\phi_j\}_{j=1}^n$ 和 $\{\psi_k\}_{k=1}^m$ 分别为 X_E^h 和 $\times P^h$ 的基函数。则数值解 $\mathbf{u}_h^{(n+1)} = (u_{xh}^{(n+1)}, u_{yh}^{(n+1)}), p_h$ 可以写成如下形式：

$$u_{xh}^{(n+1)} = \sum_{j=1}^n u_j \phi_j, \quad u_{yh}^{(n+1)} = \sum_{j=1}^n v_j \phi_j, \quad p_h^{(n+1)} = \sum_{k=1}^m p_k \psi_k.$$

将 $u_{xh}^{(n+1)}, u_{yh}^{(n+1)}, p_h^{(n+1)}$ 带入离散弱形式 (3.17) 中，可以得到线性方程组

$$\begin{bmatrix} \nu A & 0 & B_x^T \\ 0 & \nu A & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ p \end{bmatrix} = \begin{bmatrix} \hat{f}_x \\ \hat{f}_y \\ \hat{g} \end{bmatrix}. \quad (3.21)$$

为简便起见，我们令 (3.21) 中的右端项分别取为 $\hat{f}_x = 0, \hat{f}_y = 0, \hat{g} = 0$ 。我们选用 MINRES(极小化残差方法)来求解 (3.21)，通常情况下，我们要为 MINRES 选择一个有效的预处理器。在前面一章中，压力空间的质量矩阵：

$$Q = [Q_{ij}], \quad q_{ij} = \int_{\Omega} \nabla \psi_i \cdot \nabla \psi_j,$$

可以作为 Schur 补 $S = B_x A^{-1} B_x^T + B_y A^{-1} B_y^T$ 的近似矩阵。在实际的计算中，压力质量矩阵的对角阵 $\hat{Q} = \text{diag}(Q)$ ，就能有比较好的效果。因此，我们选取对角块矩阵 K

$$K = \begin{bmatrix} \nu A & 0 & 0 \\ 0 & \nu A & 0 \\ 0 & 0 & \hat{Q} \end{bmatrix}, \quad (3.22)$$

作为 (3.21) 的系数矩阵 M 的近似。

$$M = \begin{bmatrix} \nu A & 0 & B_x^T \\ 0 & \nu A & B_y^T \\ B_x & B_y & 0 \end{bmatrix}. \quad (3.23)$$

在预处理过程中，我们需要对矩阵 K 求逆，即完成 $V_{dst} = K^{-1}V_{src}$ 的操作，其中 V_{dst}, V_{src} 分别为

$$V_{dst} = (V_{dst}^x, V_{dst}^y, V_{dst}^p)^T, \quad V_{src} = (V_{src}^x, V_{src}^y, V_{src}^p)^T. \quad (3.24)$$

我们不是直接求解 $V_{dst} = K^{-1}V_{src}$ ，而是把 K 分成三个对角块 $(0, 0)$, $(1, 1)$ 和 $(2, 2)$ 块，然后每个块进行单独求解。我们用 posision 问题的代数多重网格求解器(AFEPack [54] 中的 AMG 求解器)来求解

$$\begin{aligned} \nu A V_{des}^x &= V_{src}^x, \\ \nu A V_{des}^y &= V_{src}^y. \end{aligned}$$

对于 $(2, 2)$ 块，

$$V_{dst}^p = \frac{V_{src}^p}{\text{diag}(\hat{Q})}, \quad (3.25)$$

那么我们就得到了 V_{dst} 。

3.4 移动网格策略

我们主要的想法是有两套网格，速度网格和压力网格，速度网格可以由压力网格加密一次得到。因此我们对压力网格进行移动，然后同步速度网格。

在 $t = t_n$ 时刻，获得有限元解 $(\mathbf{u}_h^{(n)}, p_h^{(n)})$ 。下面的问题是如何用 t_n 时刻数值解和旧的网格 $\mathcal{T}_h^{(n)}$ 来获得新的网格 $\mathcal{T}_h^{(n+1)}$ ，并在新网格上求解 N-S 方程获得 $(\mathbf{u}_h^{(n+1)}, p_h^{(n+1)})$ 。我们采取文献 [28] 中的方法，注意到我们区域的边界均是 Dirichlet 或者 Neumann 边界，分为以下四步：

3.4.1 获取控制函数

选择一个合适的控制函数，对于移动网格的结果是非常重要的。应用在不可压 Navier-Stokes 方程上的控制函数主要有如下几种。令 $m = \frac{1}{G}$ 其中 m 是 (3.31) 中的一个纯量函数。关于 G 有几种不同的选择。一种是基于涡量

$$G_0 = \sqrt{1 + \alpha|\omega|^\beta}, \quad (3.26)$$

其中 $\omega = \nabla \times \mathbf{u}$, α 和 β 是两个正的常数。另外一个选择， G 是基于数值解的梯度

$$G_1 = \sqrt{1 + \alpha|\nabla \mathbf{u}|^\beta}. \quad (3.27)$$

对于线性元 v_h 逼近真实解 v ，下面的后验误差估计公式可以用来近似计算误差

$$|v - v_h|_{1,\Omega} \sim \eta(v_h) := \sqrt{\sum_{l: \text{内部边界}} \int_l [\nabla v_h \cdot \mathbf{n}_l]^2 dl}, \quad (3.28)$$

其中 $[.]_l$ 意味着边 l 上的跳跃，即 $[v]_l = v|_{l^+} - v|_{l^-}$ 。很自然的在每个单元上等分布数值误差 $\eta(v_h)$ ，控制函数为一下形式

$$G_2 = \sqrt{1 + \alpha\eta^2(v_h)}. \quad (3.29)$$

[28] 中对 (3.29) 进行了改进

$$G_3 = \sqrt{1 + \alpha [\eta(v_h)/ \max \eta(v_h)]^\beta}. \quad (3.30)$$

其中 $\beta > 2$ 时有更好的效果。

3.4.2 获取新的逻辑网格

求解椭圆形方程

$$\begin{aligned} \nabla_{\boldsymbol{x}} (m \nabla_{\boldsymbol{x}} \boldsymbol{\xi}) &= 0, \\ \boldsymbol{\xi} &= \boldsymbol{\xi}_b. \end{aligned} \quad (3.31)$$

其中 m 是上一节中的纯量函数。我们定义初始的逻辑网格 \mathcal{T}_c^0 (\mathcal{A}^0 为它的节点)。一旦初始的逻辑网格给定，在整个的求解过程中，将一直保持不变。通过求解(3.31)我们可以得到新的逻辑网格 \mathcal{T}_c^* (\mathcal{A}^* 为它的节点)。

3.4.3 物理网格的移动方向

我们先引入一些定义。 \mathcal{T}_h 为物理区域上的三角剖分。第*i*个点定义为 X_i ，以 X_i 为顶点的单元的集合称之为 T_i 。相应的计算区域上的标记为 \mathcal{T}_c , \mathcal{A}_i 和 $T_{i,c}$ 。 \mathcal{A}_i 点在计算区域上的坐标定义为 $(\mathcal{A}_i^1, \mathcal{A}_i^2)^T$ 。从而我们得到新旧逻辑网格的差:

$$\delta \mathcal{A}_i = \mathcal{A}^{(0)} - \mathcal{A}^*. \quad (3.32)$$

对于一个给定的单元 $E \in \mathcal{T}_h$, $X_{E_k}, 0 \leq k \leq 2$, 作为它的三个顶点。从 $V_{\mathcal{T}_c^*}(\Omega)$ 到 $V_{\mathcal{T}}(\Omega)$ 的分片线性映射在单元 E 上的梯度是常数，并且满足下面的方程组:

$$\begin{aligned} &\begin{pmatrix} \mathcal{A}_{E_1}^{*,1} - \mathcal{A}_{E_0}^{*,1} & \mathcal{A}_{E_2}^{*,1} - \mathcal{A}_{E_0}^{*,1} \\ \mathcal{A}_{E_1}^{*,2} - \mathcal{A}_{E_0}^{*,2} & \mathcal{A}_{E_2}^{*,2} - \mathcal{A}_{E_0}^{*,2} \end{pmatrix} \begin{pmatrix} \frac{\partial x^1}{\partial \xi^1} & \frac{\partial x^1}{\partial \xi^2} \\ \frac{\partial x^2}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^2} \end{pmatrix} \\ &= \begin{pmatrix} X_{E_1}^1 - X_{E_0}^1 & X_{E_2}^1 - X_{E_0}^1 \\ X_{E_1}^2 - X_{E_0}^2 & X_{E_2}^2 - X_{E_0}^2 \end{pmatrix}. \end{aligned}$$

求解上面的方程组，可以获得单元 E 上的 $\partial \boldsymbol{x} / \partial \xi$ 。如果以单元的面积作为权重，则第*i*个点的加权平均的位移定义如下:

$$\delta X_i = \frac{\sum_{E \in T_i} |E| \frac{\partial \boldsymbol{x}}{\partial \xi}|_{in E} \delta \mathcal{A}_i}{\sum_{E \in T_i} |E|}, \quad (3.33)$$

其中 $|E|$ 代表单元 E 的面积。为了避免网格发生缠结，在网格移动向量前乘上一个常量 μ ，即物理区域上新网格 \mathcal{T}^* 的节点表示为:

$$X_i^* = X_i + \mu \delta X_i. \quad (3.34)$$

文献 [28] 中提出 μ 按以下方式给出:

$$\begin{vmatrix} 1 & 1 & 1 \\ x_0^1 + \mu\delta x_0^1 & x_1^1 + \mu\delta x_1^1 & x_2^1 + \mu\delta x_2^1 \\ x_0^2 + \mu\delta x_0^2 & x_1^2 + \mu\delta x_1^2 & x_2^2 + \mu\delta x_2^2 \end{vmatrix} = 0, \quad (3.35)$$

其中 $\mathbf{x}_i = (x_i^1, x_i^2), 0 \leq i \leq 2$ 表示第 i 个点的坐标。令 μ_i^* 为方程 (3.35) 的最小正根，则令

$$\mu = \min(1, \frac{\mu_i^*}{2}). \quad (3.36)$$

3.4.4 散度为 0 的插值

用移动网格方法求解不可压流体时，要保证插值的过程散度是为 0 的。通过求解一个对流方程，对流的速度是网格的移动速度，从而实现旧的物理网格上的数值解到新的物理网格上数值解的插值。令 $u_h = \sum u_i \phi_i, u_h \in \mathbf{X}_E^h$, ϕ_i 是有限元空间 \mathbf{X}^h 的基函数。引入一个虚拟的时间 τ ，假设基函数 ϕ_i 和 u_i 均是关于 τ 的函数，即 $\phi_i = \phi_i(\tau), u_i = u_i(\tau)$ 。我们引入一个从旧网格 $x^{\text{旧}}$ 到新网格 $x^{\text{新}}$ 网格点的连续变换：

$$x_i(\tau) = X_i + \tau(X_i^* - X_i), \quad \tau \in [0, 1], \quad (3.37)$$

其中 $X_i^* = x_i^{\text{新}}, X_i = x_i^{\text{旧}}$ 基于 (3.37) 的连续形式 $x(\tau) = x^{\text{旧}} + \tau(x^{\text{新}} - x^{\text{旧}})$ ，基函数可以定义为 $\phi_i(\tau) = \phi_i(x(\tau))$ 并且 $u_i = u_i(x(\tau))$.

在插值的过程中，我们要保持解曲线 $u_h = \sum u_i \phi_i$ 关于 τ 在弱形式下是不变的。即对 $\forall \psi \in \mathbf{X}^h$, $(\partial_\tau u_h, \psi) = 0$ 。通过直接计算可得

$$\frac{\partial \phi}{\partial \tau} = -\nabla_{\mathbf{x}} \phi_i \cdot \delta \mathbf{x}, \quad (3.38)$$

其中 $\delta \mathbf{x} = x^{\text{旧}} - x^{\text{新}}$ 。紧接着

$$\begin{aligned} 0 &= (\partial_\tau u_h, \psi) \\ &= (\partial_\tau \sum u_i(x(\tau)) \phi_i, \psi) \\ &= (\sum \phi_i \partial_\tau u_i(x(\tau)) + \sum u_i \partial_\tau \phi_i) \\ &= (\sum \phi_i \partial_\tau u_i(x(\tau)) - \sum u_i \nabla_{\mathbf{x}} \phi_i \cdot \delta \mathbf{x}, \psi) \\ &= (\sum \phi_i \partial_\tau u_i(x(\tau)) - \nabla_{\mathbf{x}} u_h \cdot \delta \mathbf{x}, \psi). \end{aligned} \quad (3.39)$$

我们将 (3.39) 应用到不可压流上, 即速度场要满足散度为 0 的条件。令 \mathbf{X}^h 为散度为 0 的空间:

$$\mathbf{X}_E^h = \mathbf{X}_E^h \cap \{\mathbf{u}_h | \nabla \cdot \mathbf{u}_h = 0\}, \quad (3.40)$$

那么 (3.39) 将变成: 寻找 $w_h \in \mathbf{X}^h$ 使得

$$\left(\sum \phi_i \partial_\tau u_i - \sum u_i \nabla_{\mathbf{x}} \phi_i \cdot \delta \mathbf{x}, z_h \right) = 0 \quad \forall z_h \in \mathbf{X}^h. \quad (3.41)$$

上面的结果意味着

$$\sum \phi_i \partial_\tau u_i - \sum u_i \nabla_{\mathbf{x}} \phi_i \cdot \delta \mathbf{x} \in \mathbf{X}^{h\perp}, \quad (3.42)$$

其中 $\mathbf{X}^{h\perp} + \mathbf{X}^h = L^2$ 。根据文献 [41] 中的定理 2.7, 如果区域 Ω 是单连通的, 那么

$$\mathbf{X}^{h\perp} = \{\nabla q | q \in H^1(\Omega)\}, \quad (3.43)$$

则存在 $\nabla p \in \mathbf{X}^{h\perp}$ 使得

$$\begin{aligned} \sum \phi_i \partial_\tau u_i - \sum u_i \nabla_{\mathbf{x}} \phi_i \cdot \delta \mathbf{x} &= -\nabla \hat{p}, \\ \nabla_{\mathbf{x}} \cdot \mathbf{u}_h &= 0. \end{aligned} \quad (3.44)$$

注 3.1. 这里的 \hat{p} 跟外部 Navier-Stokes 方程的解 p 不一致, 只是一个辅助量。

(3.44) 的弱形式: 寻找 $(\mathbf{u}_h, \hat{p}_h) \in \mathbf{X}_E^h \times P^h$ 使得

$$\begin{aligned} \left(\sum \phi_i \partial_\tau u_i - \sum u_i \nabla_{\mathbf{x}} \phi_i \cdot \delta \mathbf{x}, v_h \right) &= (\hat{p}_h, \nabla v_h), \quad \forall v_h \in \mathbf{X}_E^h, \\ (\nabla_{\mathbf{x}} \cdot u_h, q_h) &= 0, \quad \forall q_h \in P^h. \end{aligned} \quad (3.45)$$

(3.44) 和 (3.45) 的初值为在 $t = t_n$ 时刻的网格上外部 N-S 方程的解。时间方向的离散我们用全隐格式:

$$\begin{aligned} \left(\frac{\sum \phi_i u_{i,*}^{(n)} - \sum \phi_i u_{i,*}^{(n)}}{\Delta \tau}, v_h \right) - \left(\sum u_i^{(n)} \nabla \phi_i \cdot \delta \mathbf{x}, v_h \right) &= \left(\hat{p}_{h,*}^{(n)}, \nabla v_h \right), \\ \left(\nabla \cdot \mathbf{u}_{h,*}^{(n)}, q_h \right) &= 0. \end{aligned} \quad (3.46)$$

注意到在做数值解插值的过程中, 物理网格还没有发生移动, 这时候基函数 ϕ 仍然是 $t = t_n$ 时刻网格上的基函数。所以 $u_{h,*}^{(n)} = \sum u_{i,*}^{(n)} \phi_i^{(n)}$, $u_h^{(n)} = \sum u_i^{(n)} \phi_i^{(n)}$

简单整理一下：

$$\begin{aligned} \left(\frac{\mathbf{u}_{h,*}^{(n)} - \mathbf{u}_h^{(n)}}{\Delta\tau}, \mathbf{v}_h \right) - \left(\nabla \mathbf{u}_{h,*}^{(n)} \cdot \delta\mathbf{x}, \mathbf{v}_h \right) &= \left(\hat{p}_{h,*}^{(n)}, \nabla \mathbf{v}_h \right), \\ \left(\nabla \cdot \mathbf{u}_{h,*}^{(n)}, q_h \right) &= 0, \end{aligned} \quad (3.47)$$

其中 $\mathbf{u}_h^{(n)}$ 和 p_h 是在 $t = t_n$ 时刻的网格上, N-S 方程的数值解。而 $\mathbf{u}_{h,*}^{(n)}$ 和 $p_{h,*}^{(n)}$ 是在新的网格上的数值解。但这组解不是外部 N-S 方程的解。在新网格上求解 N-S 方程, 从而得到 t_{n+1} 时刻新网格上的解。为了更好的说明我们的数值方法, 我们将算法的流程图如算法 3.2 所示。

Algorithm 3.2 移动网格有限元方法来求解 Navier Stokes 方程

- 1: 在初始速度网格 $\Delta_v^{(0)}$ 和压力网格 $\Delta_p^{(0)}$ 上, 求解 $t = t_n$ 时刻 Stokes 方程 (3.1), 获得数值解 $\mathbf{u}_h^{(0)}, p_h^{(0)}$ 。
 - 2: **while** $t_n < T$ **do**
 - 3: 在压力网格 $\Delta_p^{(n)}$ 上, 用 $\mathbf{u}_h^{(n)}, p_h^{(n)}$ 来计算控制函数, 并通过求解 (3.31), 获得 $\boldsymbol{\xi}^*$ 。
 - 4: 判断 $\|\boldsymbol{\xi}^* - \boldsymbol{\xi}^{(0)}\|_{L^2}$ 是否小于容忍量 ϵ , 如果是迭代结束, 否则继续做 5 - 8.
 - 5: 用 $\boldsymbol{\xi}^* - \boldsymbol{\xi}^{(0)}$ 两者之差来计算网格 $\Delta_p^{(n)}$ 的移动量 $\delta\mathbf{x}$ 。
 - 6: 利用 5 中 $\delta\mathbf{x}$, 在速度网格 $\Delta_v^{(n)}$ 上求解更新数值解的方程 (3.44), 得到新网格上的中间量 $\mathbf{u}_{h,*}^{(n)}, p_{h,*}^{(n)}$ 。
 - 7: 更新 $\Delta_p^{(n)}$, 通过几何遗传树结构, 来同步 $\Delta_v^{(n)}$, 得到新的 $\Delta_p^{(n+1)}$ 和 $\Delta_v^{(n+1)}$
 - 8: 回到 3。
 - 9: 在 $\Delta_v^{(n+1)}$ 和 $\Delta_p^{(n+1)}$ 上求解 Navier-Stokes 方程 (3.18)。从而真正获得 $t = t_{n+1}$ 时刻的数值解 $\mathbf{u}_h^{(n+1)}, p_h^{(n+1)}$ 。
 - 10: $n = n + 1$
 - 11: **end while**
-

3.5 数值算例

3.5.1 碰撞流

这个例子为稳态 Stokes 方程的精确解, 粘性系数 $\nu = 1.0$:

$$u_x = 20xy^3; \quad u_y = 5x^4 - 5y^4; \quad p = 60x^2y - 20y^3 + \text{constant}. \quad (3.48)$$

其中计算区域 $\Omega = [-1, -1] \times [1, 1]$, 边界条件全部是 Dirichlet 条件。这个例子是用来检验移动网格方法的收敛阶, 此时解比较光滑。从文献 [9] 可知, 我们期望移动网格的收敛阶: 速度有二阶收敛, 压力一阶收敛。

压力网格	$\ \mathbf{u} - \mathbf{u}_h\ _{L^2}$	误差阶	$\ \mathbf{u} - \mathbf{u}_h\ _{H^1}$	$\ p - p_h\ _{L^0}$	误差阶	$\ p - p_h\ _{H^1}$
10×10	1.18×10^{-1}		3.37×10^0	8.49×10^{-1}		1.84×10^1
20×20	2.94×10^{-2}	2.01	1.67×10^0	2.330×10^{-1}	1.82	1.06×10^1
40×40	7.37×10^{-3}	1.99	8.35×10^{-1}	6.54×10^{-2}	1.78	5.89×10^0

表 3.1: 用移动网格计算碰撞流的误差, $\nu = 1.0$ 。

压力网格	涡量 L^2 误差	误差阶	散度 L^2 误差	误差阶
10×10	2.35×10^0		2.41×10^0	
20×20	1.16×10^0	1.01	1.20×10^0	1.00
40×40	5.79×10^{-1}	1.00	6.01×10^{-1}	1.00

表 3.2: 用移动网格计算碰撞流的散度和涡量误差, $\nu = 1.0$ 。

我们选取 (3.27) 为控制函数, 其中 $\mathbf{u} = (u_x, u_y)^T$ 。 α 和 β 分别取为 0.002 和 2。从表 3.1 中可以看出速度 L^2 误差有二阶, 压力收敛阶是大于一阶不到二阶。从表 3.2 可以看出速度散度和涡量均有一阶收敛。

首先我们判断网格是不是往正确的方向移动。从图 3.2 中看出控制函数 G_1 最大的地方分布在区域的四个顶角上, 中间区域控制函数的值是比较小的。网格应该从控制函数小的地方移动到控制函数值大的地方。而网格的移动方向也确实是往四个顶角上移动。因此我们可以确定网格移动方向是对的。再者, 判断选取 G_1 为控制函数是否合适。从图 3.3 中看到: 在均匀网格下, 速度 L^2 误差最大的地方分布在四个顶角上, 这与控制函数的分布是一致的。所以选取 G_1 为控制函数是合理的。

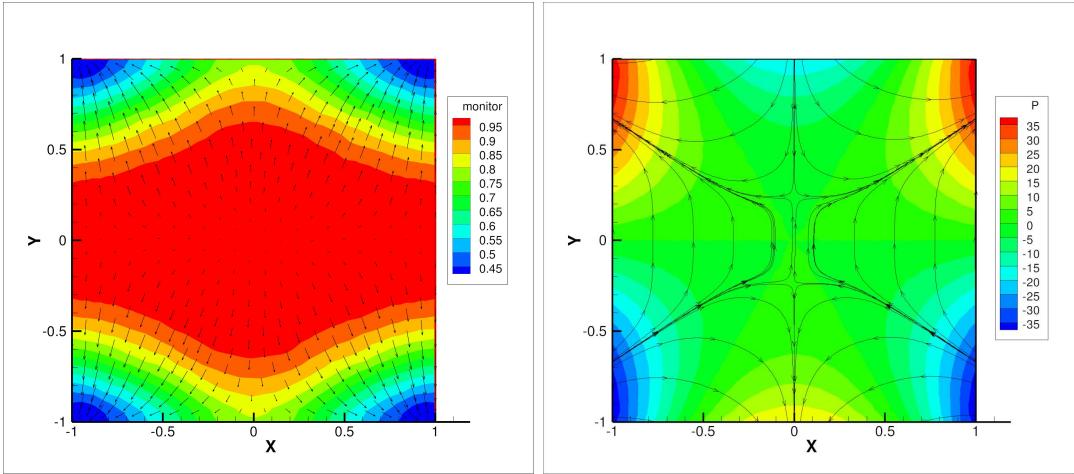


图 3.2: 左: $m = \frac{1}{G_1}$ 的等高线和网格移动方向; 右: 压力等高线和速度的流速线, $\alpha = 0.002, \beta = 2.0$ 。

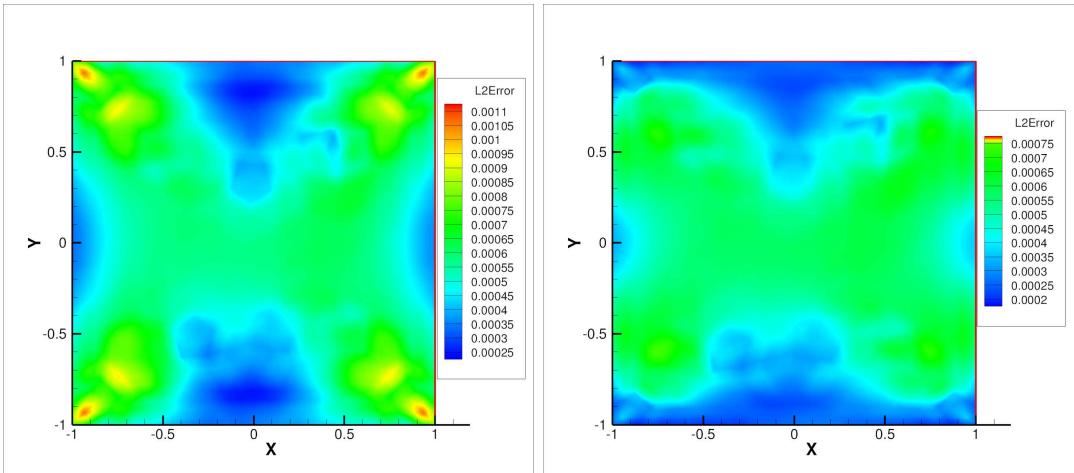


图 3.3: 速度 L^2 误差分布, 左: 均匀网格; 右: 移动网格 $\alpha = 0.002, \beta = 2.0$ 。

3.5.2 喷射流

这个算例模拟的是一个很细的流体喷射入一个静止的流场, 这种现象可以参考 [32], 一本有趣的画册。我们的计算区域是 $\Omega = [0, 12] \times [-3, 3]$ 并且粘性系数是 $\nu = 0.0005$ 。自然条件设置在出流边界 $x = 12$ 上, 同时入流边 $x = 0, y \in [-0.1, 0.1]$ 上设置 Poiseuille 流条件:

$$u_x = 1 - 100y^2, \quad u_y = 0. \quad (3.49)$$

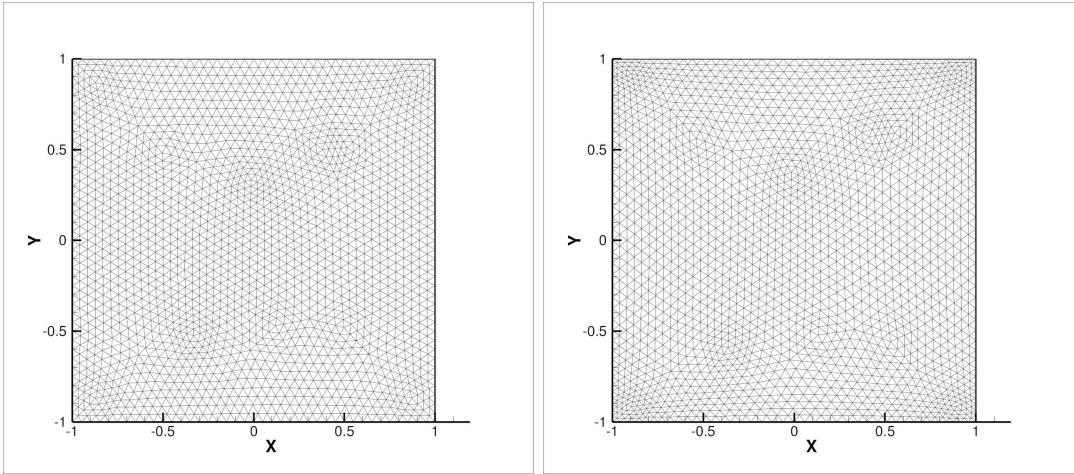


图 3.4: 网格对比, 左: 均匀网格 20×20 ; 右: 移动网格 $\alpha = 0.002, \beta = 2.0$ 。

无滑移边界条件设置在 $\partial\Omega$ 的其余部分。

在移动网格策略中, 我们选取 (3.26) 作为控制函数, 参数 $\alpha = 2.0, \beta = 2.0$ 。我们所知, 当细流喷射到静止的流场中, 会产生两个对称的涡。为了提高计算精度, 在入射流周围需要更多的网格。从图 3.5 中可以看出, 网格集中在涡量值比较大的地方。随着时间的发展, 流体的不稳定现象会出现, 这跟 [32] 中的物理现象是一致的。

3.5.3 流体经过一个凹形台阶

我们考虑 [78] 中的算例, N-S 流体以 $Re = 1000$ 经过一个凹形的台阶, 计算区域是 $\Omega = (0, 4) \times (0, 1) / (1.2, 1.6) \times (0, 0.4)$, 在上下边界上, 设置 $\mathbf{u} = (0, 0)^T$ 的边界条件。入流边上的边界条件为 $\mathbf{u} = (4y(1-y), 0)$, 出流边界是自然条件。

我们选取涡量作为控制函数, 控制函数的参数是 $\alpha = 0.4, \beta = 2.0$ 。众所周知, 在这个问题中流体在凹进去的角上, 会出现奇异性。 $t = 0.5s, t = 1s$, 和 $t = 2s$ 时刻的网格以及涡量的等高线在图 3.7, 图 3.8 和图 3.10 中展示出来。我们发现, 网格在凹进去的边上比较密, 并且跟涡的结构是一致的。

$t = 2s$ 时, 压力的等高线以及速度的流速线在图 3.11 中给出。我们可以看出, 拐角处压力的等高线是平滑的。

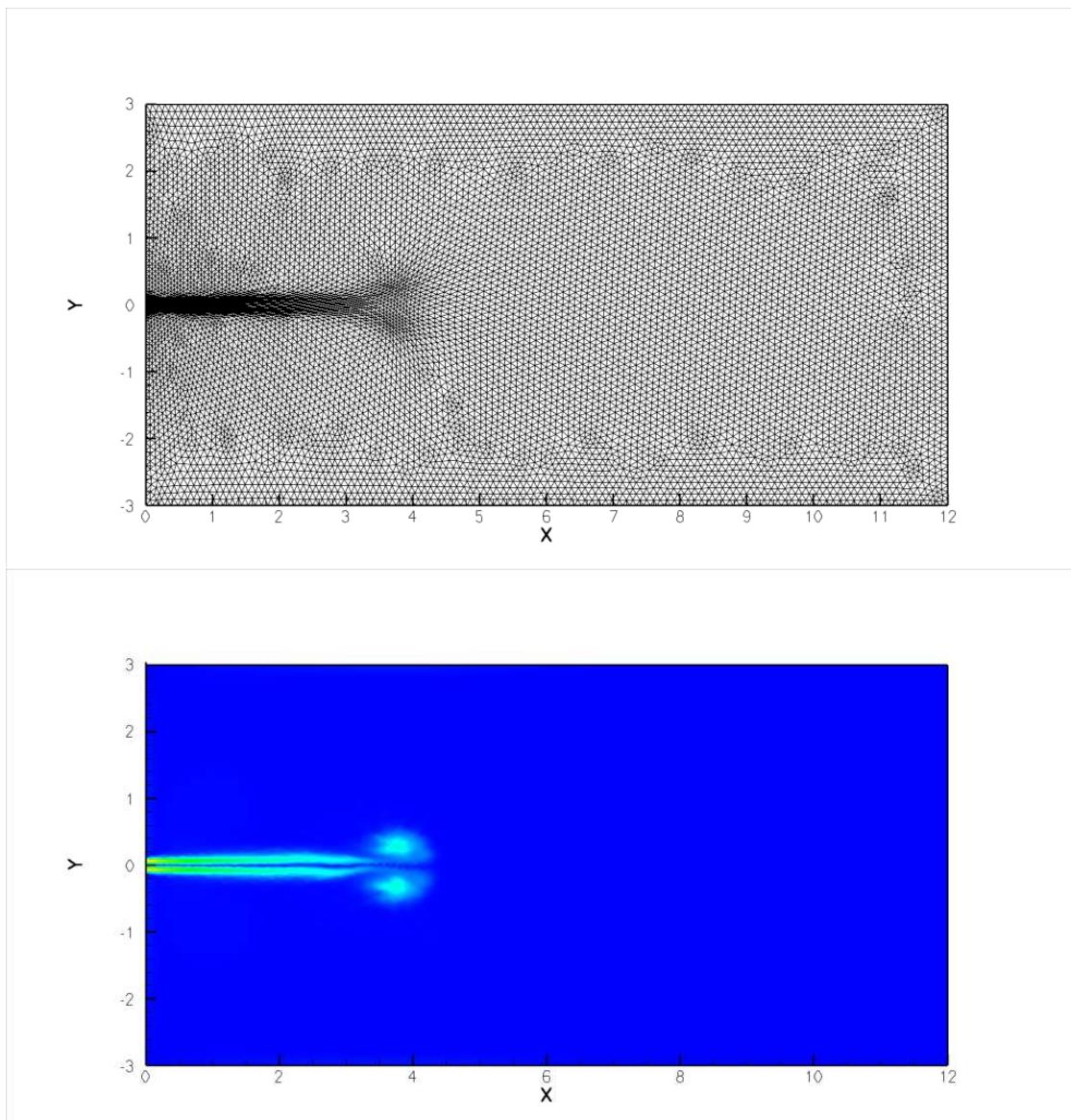


图 3.5: 顶部: 移动网格, 底部: 涡量等高线, $t = 12s$ 。

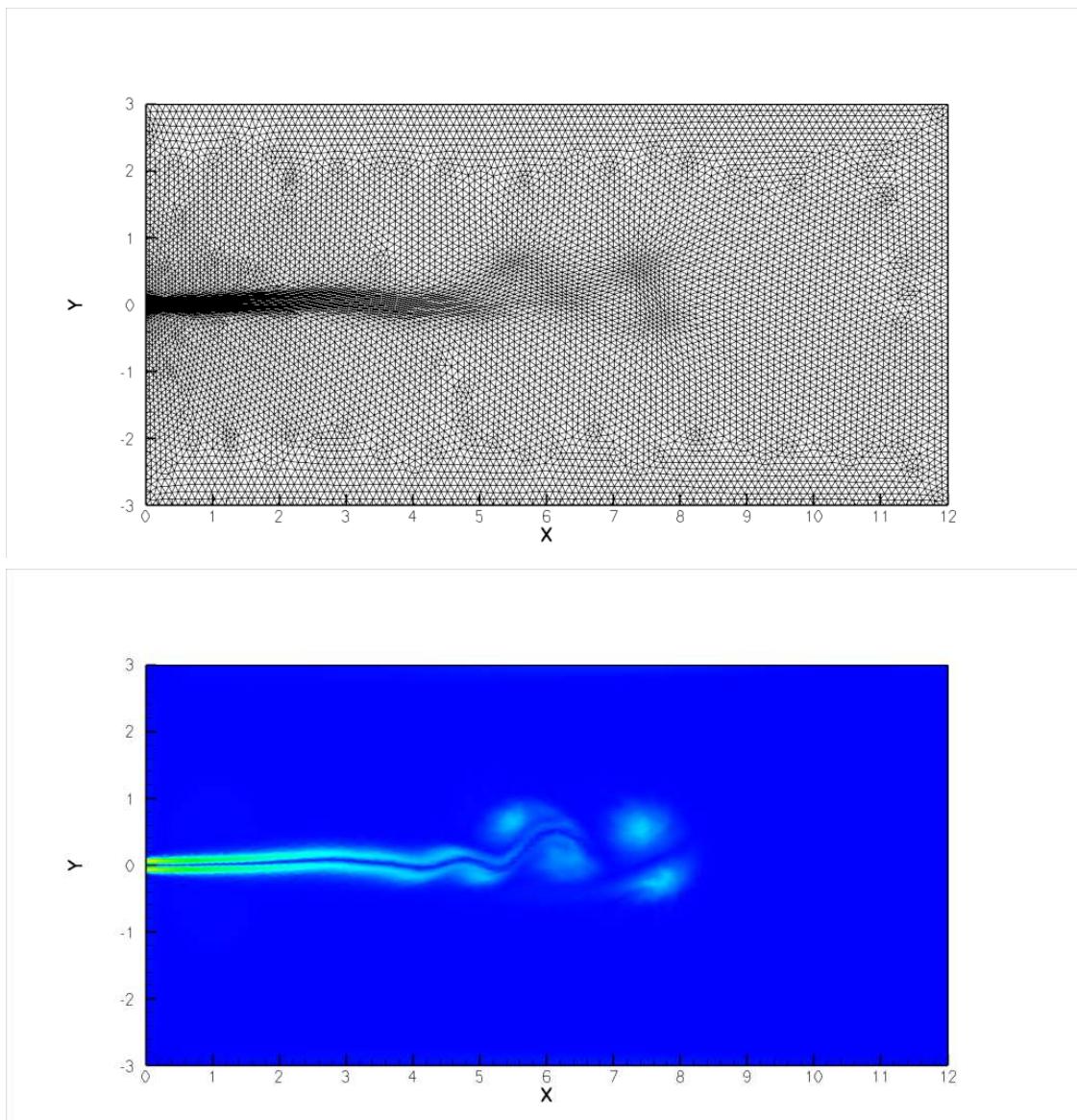


图 3.6: 顶部: 移动网格, 底部: 涡量等高线, $t = 27s$ 。

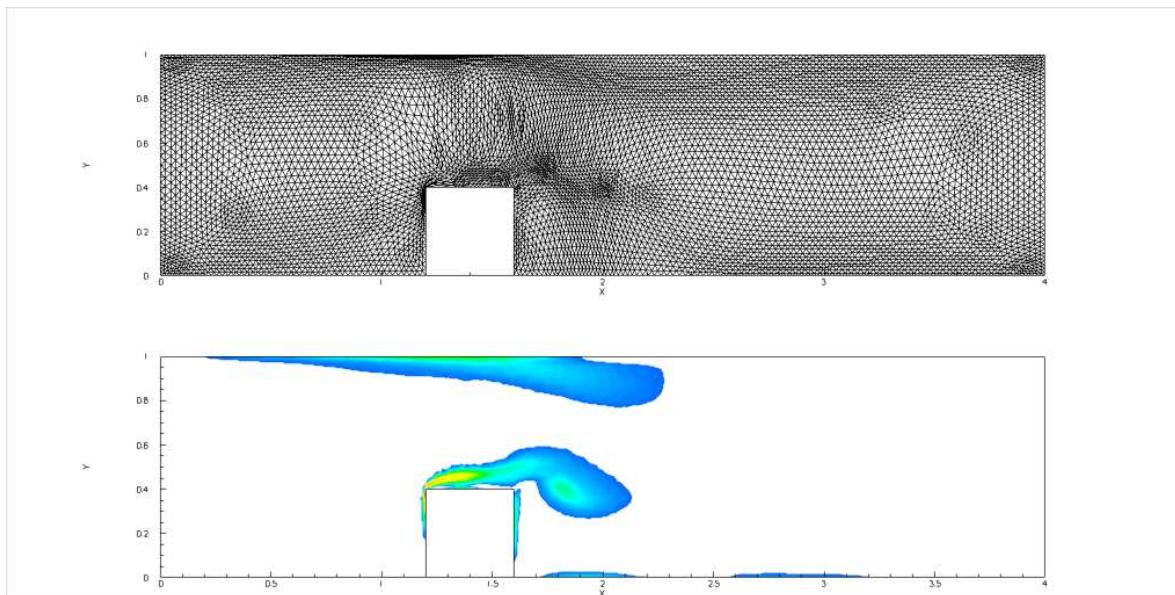


图 3.7: 顶部: 移动网格; 底部: 涡量等高线, $t = 0.5s$ 。

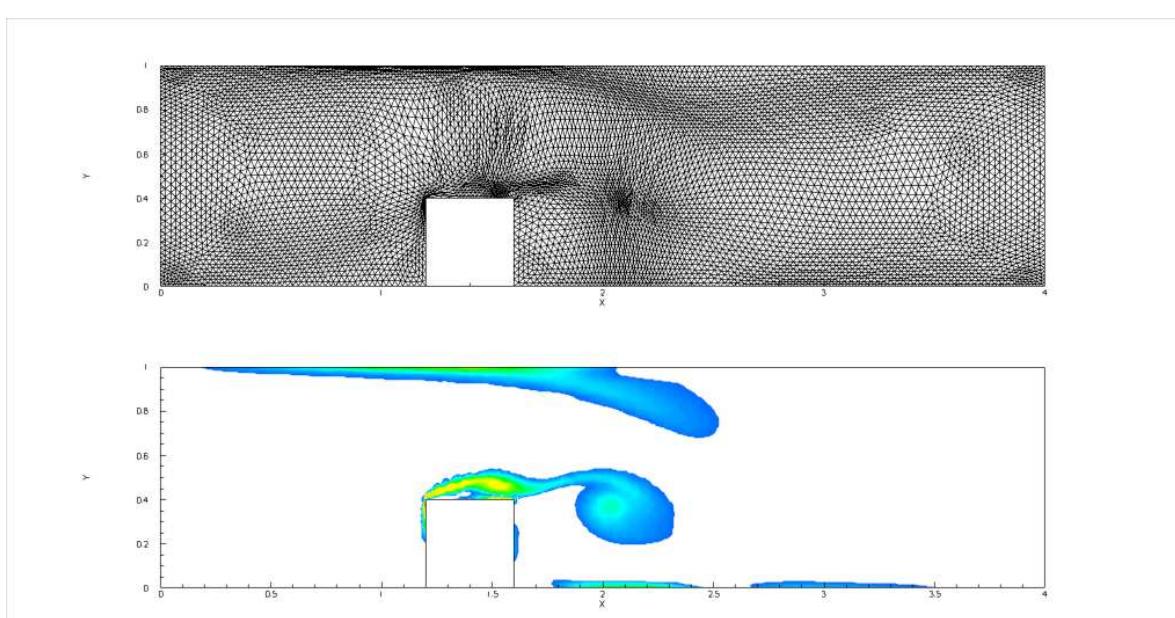


图 3.8: 顶部: 移动网格; 底部: 涡量等高线, $t = 1s$ 。

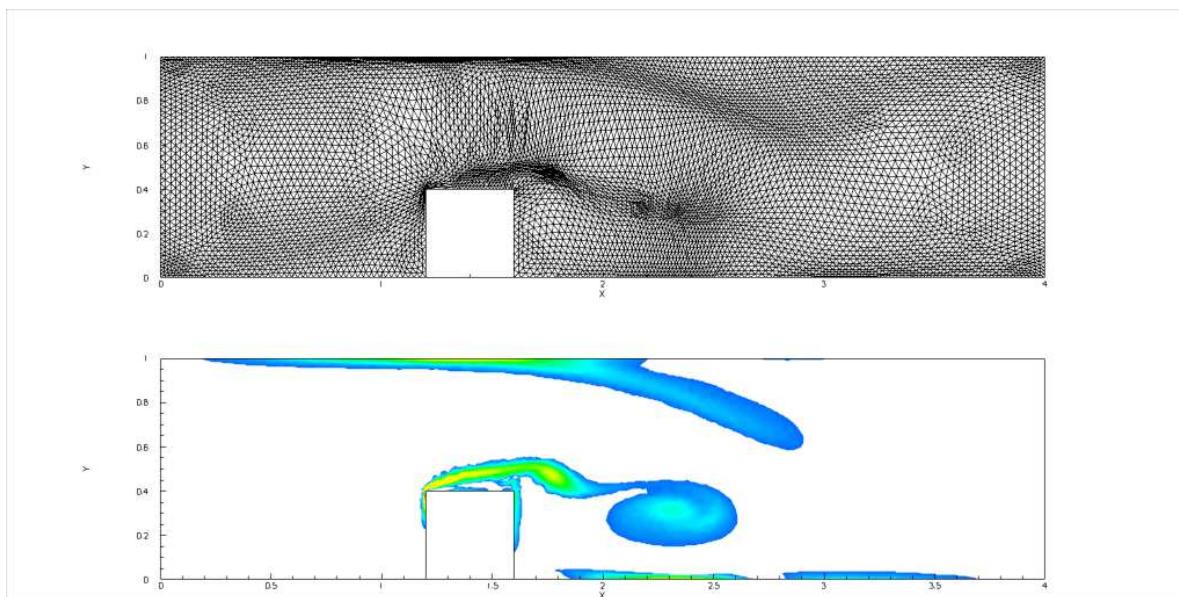


图 3.9: 顶部: 移动网格; 底部: 涡量等高线, $t = 1.5s$ 。

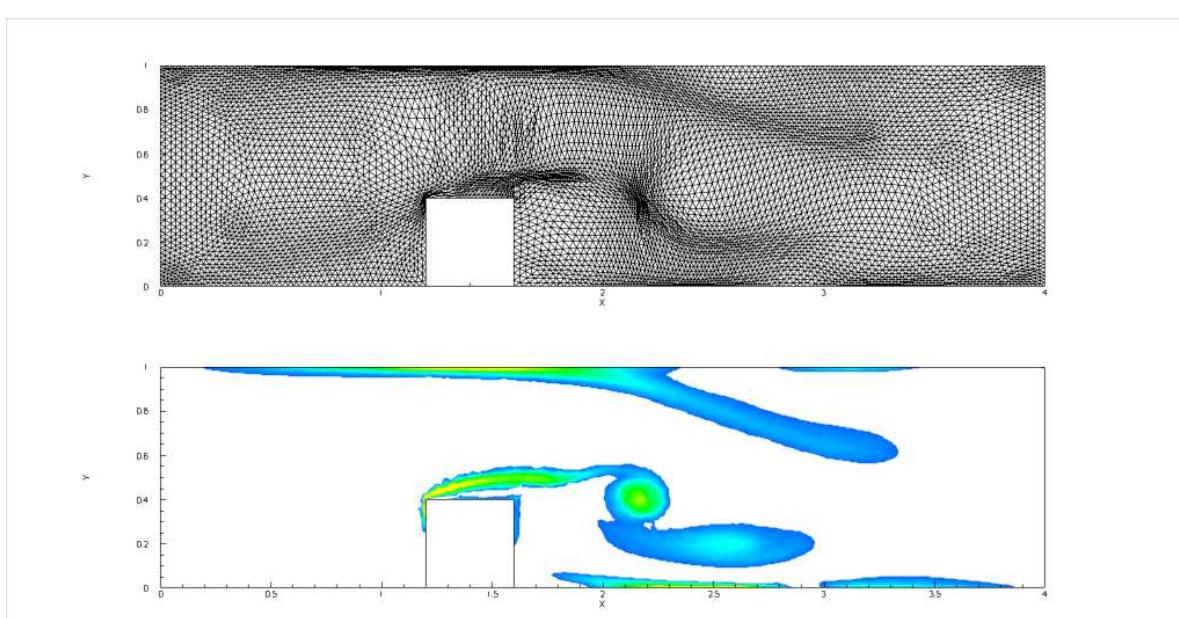


图 3.10: 顶部: 移动网格; 底部: 涡量等高线, $t = 2s$ 。

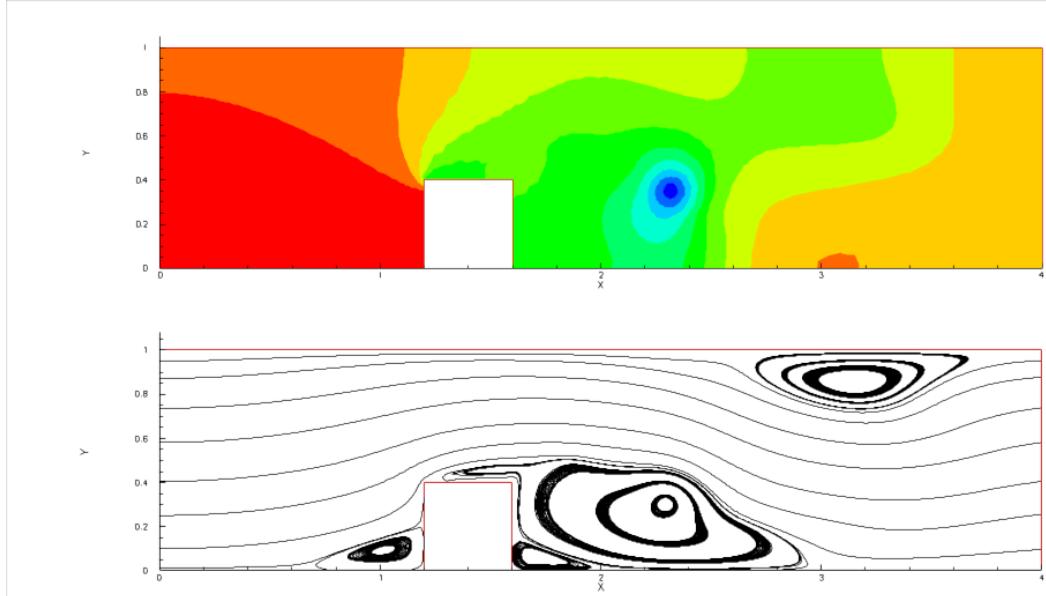


图 3.11: 顶部: 压力等高线; 底部: 速度流速线, $t = 2s$ 。

3.5.4 圆柱绕流

这个是经典算例，在一个长方形的管道中，流体流过一个圆柱。在 [21] 中，作者用移动网格方法求解流函数形式的 N-S 方程。圆柱圆心是 $(0, 0)$ ，半径是 $r = 0.3$ 。粘性系数是 $\nu = 0.003$ ，计算区域是 $\Omega = [-1, 5] \times [-1, 1]$ 。Poiseuille 流 $u = 1 - y^2, v = 0$ 设置在入流边界 $x = -1$ 。边界条件 $u = v = 0$ 施加在管道的上下边界。出流边界 $x = 5$ 是自然边界条件。

如果我们关注流体的细微结构，因此需要在小结构上有高分辨率。为了抓住涡的结构，在我们的移动策略中，(3.26) 作为控制函数是一个很好的选择。具体的参数是 α 和 β 分别为 1.0, 2.0。

在图 3.12 和图 3.13 中，展示了网格和涡的等高线的变化。从中可以看出，我们的网格可以抓住涡的结构，并且我们的网格质量比 [21] 中要好。当我们改变雷诺数为 $Re = 2000$ 时，从图 3.14 中可以看出，网格和涡量的结构并不像 $Re = 2000/3$ 时那么明显。

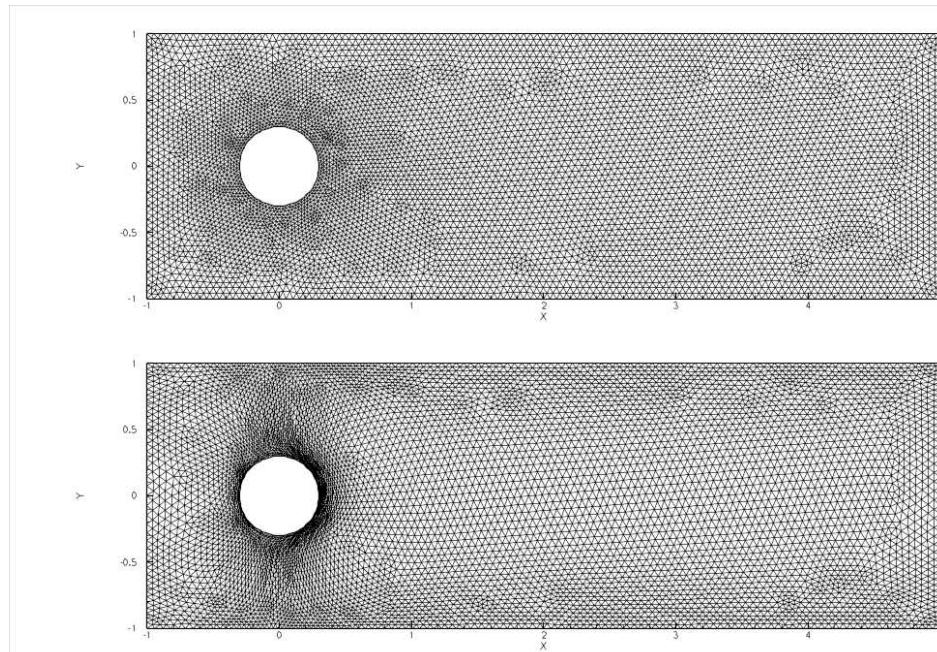


图 3.12: 顶部: 均匀网格; 底部: 初始移动网格, 粘性系数 $\nu = 0.003$ 。

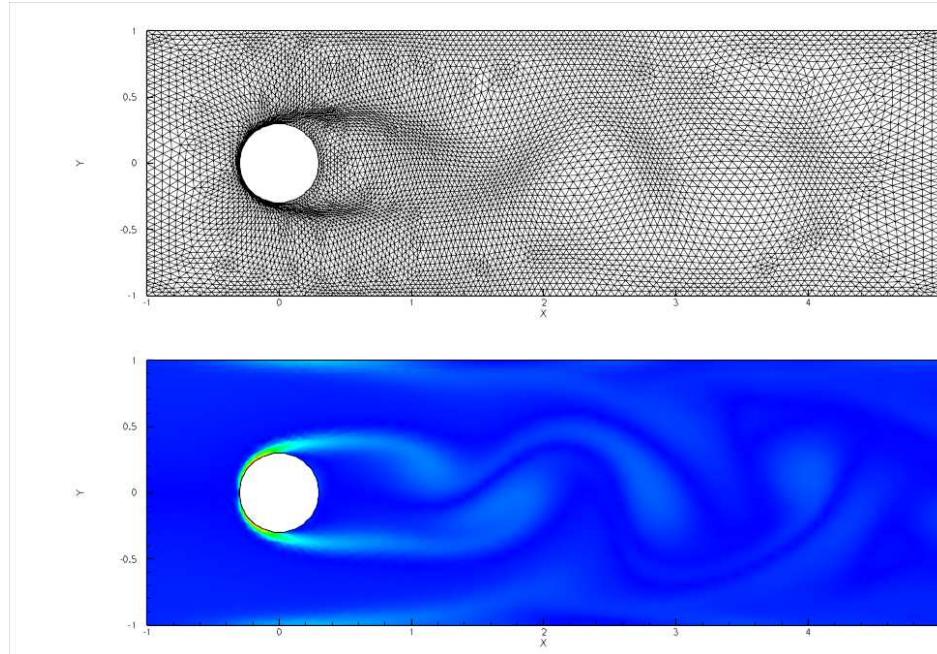


图 3.13: 顶部: 移动网格; 底部: 涡量等高线, $t = 24.5s$, 粘性系数 $\nu = 0.003$ 。

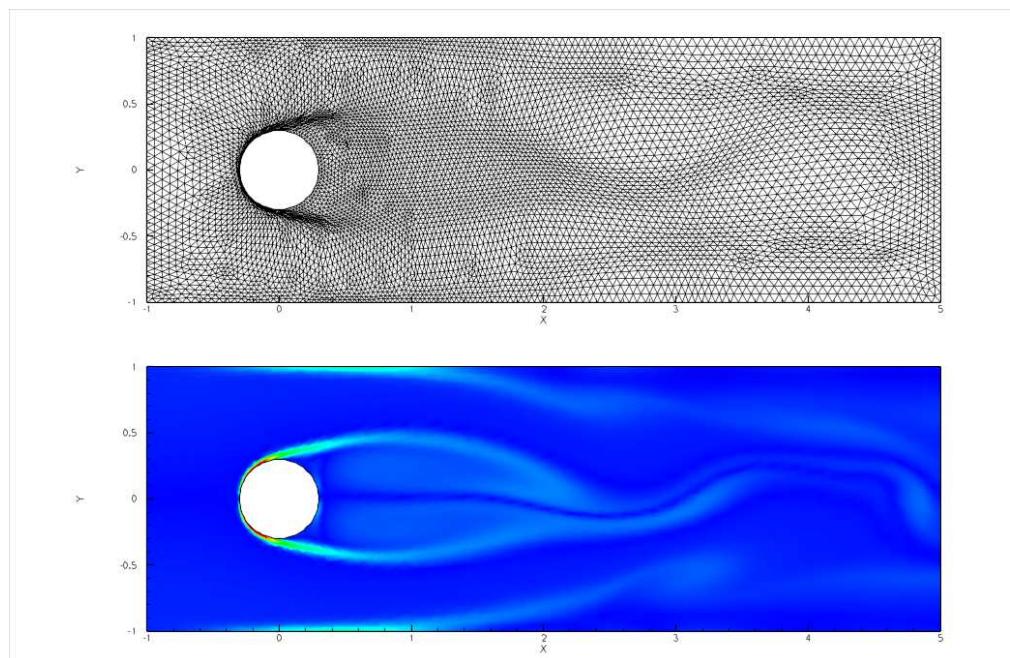


图 3.14: 顶部: 移动网格; 底部: 涡量等高线, $t = 23s$, 粘性系数 $\nu = 0.001$ 。

第 4 章 不可压 Navier-Stokes 方程的代数多重网格预处理的移动网格有限元方法

原始变量形式的不可压 Navier-Stokes 方程

$$\begin{aligned} \partial_t \mathbf{u} - \nu \nabla^2 \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p &= \mathbf{f}, \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned} \quad (4.1)$$

在 $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ 上的边界条件以及初值条件:

$$\begin{aligned} \mathbf{u} &= \mathbf{w}, && \text{on } \partial\Omega_D \times [0, T] \\ \nu \frac{\partial \mathbf{u}}{\partial n} - p &= \mathbf{0}, && \text{on } \partial\Omega_N \times [0, T], \\ \mathbf{u}|_{t=0} &= \mathbf{u}_0, && \text{in } \Omega. \end{aligned} \quad (4.2)$$

其中 $\Omega \in \mathcal{R}^d$, ($d = 2, 3$) 是计算区域, $[0, T]$ 是时间区间, \mathbf{u} 是速度变量并且常量 p 是压力, \mathbf{n} 是边界 $\partial\Omega$ 上的外法向方向, $\nu > 0$ 是粘性动力学系数。我们用 [55] 和 [29] 中提出的移动网格有限元方法求解 (4.1) 和 (4.2)。

据我们所知, 用满足 inf-sup 条件的 $4P_1 - P_1$ 元可以推出一个鞍点问题, 许和何([76]、[64]) 将两格子方法应用到 Navier-Stokes 方程的求解。很多人在为 Krylov 子空间方法提供预处理子方面做了很多工作。在 [5] 文中, 概括了求解鞍点问题常用的数值方法, 例如块预处理和多重网格预处理。文献([3], [2], [34], [39] 等) 中提出了许多块预处理方法, 来找到鞍点问题的 schur 补的更好的近似。[7] 中提出了求解 Ossen 系统的增广的基于 Lagarian 方法。在 [6] 中提出了求解不可压 Navier-Stokes 方程的维数分解的预处理方法。([12]、[13]) 中提出了求解 Navier-Stokes 方程的一种高效的 Krylov 子空间方法, 这种方法是用代数多重网格做预处理子。但是据我们所知, 对于鞍点问题的高效预处理方法基本上都是基于均匀网格上的, 尽管在 [6] 一文中, 考虑了拉伸网格的情形。

在 [37] 的基础上, 我们将为移动网格有限元方法求解 (4.1) 和 (4.2) 提供了一个代数多重网格预处理方法。我们通过数值例子来表明, 我们的预处理子的高效性。

4.1 数值离散

4.1.1 时间方向的离散

在时间层离散上，我们把时间区间 $[0, T]$ 分成 N 份， $\{t_i\}_{i=1}^N$ 。令 \mathbf{u}^j 和 p^j 是连续形式解 $\mathbf{u}(\cdot, t_j)$ 和 $p(\cdot, t_j)$ 的离散近似。在时间格式的离散上，主要分为算子分裂方法和全隐格式。在算子分裂方法，也可以看成是分部方法。最简单的方法是将算子分成两步，在每一步中分别用向前和向后欧拉格式，可以参考 [59]。在 [46] 中，将这种两步分裂方法应用到 Navier-Stokes 方程的时间离散上。通常的两步方法如下：在算法 4.1 中 \mathbf{u}^* 的取值要保证散度为 0

Algorithm 4.1 Peaceman-Rachford 格式

给定 $\mathbf{u}^0, p^0, \theta \in [0, 1], \alpha \in (0, 1), \beta \in (0, 1)$ ，假设 \mathbf{u}^n 已知，通过以下求解 \mathbf{u}^{n+1} ：

$$\frac{\mathbf{u}^{n+\theta} - \mathbf{u}^n}{\theta \delta t} - \alpha \nu \nabla^2 \mathbf{u}^{n+\theta} + \mathbf{u}^* \cdot \nabla \mathbf{u}^{n+\theta} = \beta \nu \nabla^2 \mathbf{u}^n - \nabla p^n \quad \text{在 } \Omega \text{ 内}, \quad (4.3)$$

$$\mathbf{u}^{n+\theta} = \mathbf{g}^{n+\theta} \quad \text{在 } \partial \Omega \text{ 上}:$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{N+\theta}}{(1-\theta)\delta t} - \beta \nu \nabla^2 \mathbf{u}^{n+1} + \nabla p^{n+1} = \alpha \nu \nabla^2 \mathbf{u}^{n+\theta} - \mathbf{u}^* - \nabla \mathbf{u}^{n+\theta} \quad \text{在 } \Omega \text{ 内}, \quad (4.4)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \quad \text{在 } \Omega \text{ 内},$$

$$\mathbf{u}^{n+1} = \mathbf{g}^{n+1} \quad \text{在 } \partial \Omega.$$

的条件，并且 $\alpha + \beta = 1$ 。它实际上是先求解一个对流方程 (4.3)，然后再求解一个广义的 Stokes 方程 (4.4)。这个算法第一步计算的时候需要 p^0 的值，它可以在很小的时间步长情况下，全隐格式向前发展一步得到。一般自然的， $\mathbf{u}^* = \mathbf{u}^n$ ，这样在时间格式上只有一阶精度的。这种方法不是无条件稳定的，并且需要比较小的时间步长。当 $\theta = \frac{1}{2}, \mathbf{u}^* = \mathbf{u}^{n+\theta}$ 时，可以获得二阶精度，这时候求解 (4.3) 变成了一个求解非线性的对流方程。详细内容可以参考([65])。[26] 将两步 Peaceman-Rachford 推广到三步，保持了当 $t \leftarrow \infty$ 时的数值稳定性：这种算法需要在每个时间步中计算两个广义的 Stokes 方程 (4.6) 和 (4.7)，和一个非线性的对流方程 (4.5)。当我们取 $\alpha = \beta = 1/2$ 或者 $\theta = 1 - 1/\sqrt{2}, \alpha + \beta = 1$ ，当 $t \leftarrow \infty$ 时，该格式时间上有二阶精度。特别的，

Algorithm 4.2 Glowinski Θ 格式

给定 $\mathbf{u}^0, \theta \in [0, 1/2], \alpha \in (0, 1), \beta \in (0, 1)$, 假设 \mathbf{u}^n 已知, 通过以下求解 \mathbf{u}^{n+1} :

$$\begin{aligned} \frac{\mathbf{u}^{n+\theta} - \mathbf{u}^n}{\theta \delta t} - \alpha \nu \nabla^2 \mathbf{u}^{n+\theta} + \nabla p^{n+\theta} &= \beta \nu \nabla^2 \mathbf{u}^n - \mathbf{u}^n \cdot \nabla \mathbf{u}^n \quad \text{在 } \Omega \text{ 内}, \\ \nabla \cdot \mathbf{u}^{n+\theta} &= 0 \quad \text{在 } \Omega \text{ 内}, \\ \mathbf{u}^{n+\theta} &= \mathbf{g}^{n+\theta} \quad \text{在 } \partial \Omega \text{ 上}; \end{aligned} \quad (4.5)$$

$$\begin{aligned} \frac{\mathbf{u}^{n+1-\theta} - \mathbf{u}^{n+\theta}}{(1-2\theta)\delta t} - \beta \nu \nabla^2 \mathbf{u}^{n+1-\theta} + \mathbf{u}^* \cdot \nabla \mathbf{u}^{n+1-\theta} &= \alpha \nu \nabla^2 \mathbf{u}^{n+\theta} - \nabla p^{n+\theta} \cdot \nabla \mathbf{u}^n \quad \text{在 } \Omega \text{ 内}, \\ \mathbf{u}^{n+1-\theta} &= \mathbf{g}^{n+1-\theta} \quad \text{在 } \partial \Omega \text{ 上}; \end{aligned} \quad (4.6)$$

$$\begin{aligned} \frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1-\theta}}{\theta \delta t} - \alpha \nu \nabla^2 \mathbf{u}^{n+1} + \nabla p^{n+1} &= \beta \nu \nabla^2 \mathbf{u}^{n+1-\theta} - \mathbf{u}^* \cdot \nabla \mathbf{u}^{n+1-\theta} \quad \text{在 } \Omega \text{ 内}, \\ \nabla \cdot \mathbf{u}^{n+1} &= 0 \quad \text{在 } \Omega \text{ 内}, \\ \mathbf{u}^{n+\theta} &= \mathbf{g}^{n+\theta} \quad \text{在 } \partial \Omega \text{ 上}. \end{aligned} \quad (4.7)$$

当 $\theta = 1 - 1/\sqrt{2}, \alpha = (1 - 2\theta)/(1 - \theta), \beta = \theta/(1 - \theta)$ 时, 这种方法在时间上是有二阶精度的, 并且是无条件稳定的。[67] 中提出了一种线性化 Θ - 格式, 它保持了二阶精度。它的想法是将对流方程 (4.6) 中的对流项

$$\mathbf{u}^* = \frac{2\theta - 1}{\theta} \mathbf{u}^n + \frac{1 - \theta}{\theta} \mathbf{u}^{n+\theta}. \quad (4.8)$$

从而使得非线性对流扩散方程变成了线性化的对流扩散方程。我们将全隐格式的近似方法表述如下: 最简单的时间格式是一步有限差分离散, 在如何处理非线性对流项上可以归纳为以下算法: 其中

$$\mathbf{u}^{n+\theta} = \theta \mathbf{u}^{n+1} + (1 - \theta) \mathbf{u}^n, \quad (4.12)$$

$$p^{n+\theta} = \theta p^{n+1} - (1 - \theta) p^n. \quad (4.13)$$

$\theta = 1$ 时, 是将非线性项隐式化处理了。当 $\mathbf{u}^{n+\theta} = \mathbf{u}^{n+1}, \mathbf{u}^* = \mathbf{u}^{n+1}$ 格式变成了时间方向一阶精度的向后欧拉格式, 当 $\mathbf{u}^{n+\theta} = \mathbf{u}^{n+1/2}, \mathbf{u}^* = \mathbf{u}^{n+1/2}$ 时, 就是时间方向的二阶精度的 Crank-Nicolson 格式。但是这两种格式在每个时间步

Algorithm 4.3 非线性隐式 Θ 格式

给定 $\mathbf{u}^0, \theta \in [0, 1]$, 假设 \mathbf{u}^n 已知, 通过以下求解 \mathbf{u}^{n+1} :

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\delta t} - \nu \nabla^2 \mathbf{u}^{n+\theta} + \mathbf{u}^* \cdot \nabla \mathbf{u}^{n+\theta} + \nabla p^{n+\theta} = 0 \quad \text{在 } \Omega \text{ 内}, \quad (4.9)$$

$$\nabla \cdot \mathbf{u}^{n+\theta} = 0 \quad \text{在 } \Omega \text{ 内}, \quad (4.10)$$

$$\mathbf{u}^{n+\theta} = \mathbf{g}^{n+\theta} \quad \text{在 } \partial\Omega \text{ 上}. \quad (4.11)$$

中都要求解一个非线性问题, 因此在计算效率上都不是高效的。一种折中的办法是将 $\mathbf{u}^{n+\theta} = \mathbf{u}^{n+1}, \mathbf{u}^* = \mathbf{u}^n, p^{n+\theta} = p^{n+1}$, 这样虽然在时间上损失了一阶精度, 但是时间离散格式还是无条件稳定的, 因此我们本文中的方法就是采用的这种线性化策略。我们来验证线性化向后欧拉方法的稳定性。不失一般性, 我们取 $\mathbf{g}^{n+1} = \mathbf{0}$, 对(4.9)方程两边同时乘上 \mathbf{u}^{n+1} , 并在 Ω 上做 L_2 内积。我们可以通过选取基函数使得对流项:

$$\int_{\Omega} (\mathbf{u}^n \cdot \mathbf{u}^{n+1}) \cdot \mathbf{u}^{n+1} = 0. \quad (4.14)$$

并且对质量守恒方程 (4.10) 关于 p^{n+1} 做 L_2 内积, 再利用分部积分:

$$\int_{\Omega} \nabla p^{n+1} \cdot \mathbf{u}^{n+1} = - \int_{\Omega} p^{n+1} (\nabla \cdot \mathbf{u}^{n+1}) = 0.$$

因此,

$$\frac{1}{\delta t} \int_{\Omega} (\mathbf{u}^{n+1} - \mathbf{u}^n) \mathbf{u}^{n+1} + \nu \int_{\Omega} \nabla \mathbf{u}^{n+1} \cdot \nabla \mathbf{u}^{n+1} = 0. \quad (4.15)$$

对 (4.15) 在 $[t_n, t_{n+1}]$ 上进行积分, 可得

$$\begin{aligned} \|\mathbf{u}^{n+1}\|^2 + \nu \int_{t_n}^{t_{n+1}} \|\nabla \mathbf{u}\|^2 &= \int_{\Omega} \mathbf{u}^n \mathbf{u}^{n+1} \\ &\leq \frac{1}{2} \int_{\Omega} ((\mathbf{u}^n)^2 + (\mathbf{u}^{n+1})^2) \\ &= \frac{1}{2} \|\mathbf{u}^n\|^2 + \frac{1}{2} \|\mathbf{u}^{n+1}\|^2. \end{aligned} \quad (4.16)$$

从而我们可以得到

$$\frac{1}{2} \|\mathbf{u}^{n+1}\|^2 + \nu \int_{t_n}^{t_{n+1}} \|\mathbf{u}^{n+1}\|^2 \leq \frac{1}{2} \|\mathbf{u}^n\|^2. \quad (4.17)$$

注意到 (4.17) 中的第二项是严格正的，因此

$$\frac{1}{2} \|\mathbf{u}^{n+1}\|^2 < \frac{1}{2} \|\mathbf{u}^n\|^2. \quad (4.18)$$

的成立跟时间步长 δt 无关。即线性化向后欧拉格式是无条件稳定的。在这种线性化策略的基础上，一种保持时间上二阶精度的方法是 $\theta = 1/2$, $\mathbf{u}^* = 3/2\mathbf{u}^n - 1/2\mathbf{u}^{n-1}$, 这是 [66] 中提出的。我们将这种两步方法表述如下：其中

Algorithm 4.4 Simo-Armero 格式

给定 $\mathbf{u}^0, \mathbf{u}^1$, 通过以下求解 $\mathbf{u}^k, k = 1, \dots, n+1$, $\mathbf{u}^{n+1/2} = \frac{1}{2}(\mathbf{u}^{n+1} + \mathbf{u}^n)$

$$\begin{aligned} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\delta t} - \nu \nabla^2 \mathbf{u}^{n+1/2} + \left(\frac{3}{2} \mathbf{u}^n - \frac{1}{2} \mathbf{u}^{n-1} \right) \cdot \nabla \mathbf{u}^{n+1/2} + \nabla p^{n+1/2} &= 0 \quad \text{在 } \Omega \text{ 内}, \\ \nabla \cdot \mathbf{u}^{n+1/2} &= 0 \quad \text{在 } \Omega \text{ 内}, \\ \mathbf{u}^{n+1/2} &= \mathbf{g}^{n+1/2} \quad \text{在 } \partial\Omega \text{ 上}. \end{aligned} \quad (4.19)$$

u^1 可以由向后欧拉格式得到，算法 4.4 也是无条件稳定的，证明过程可以根据线性化向后欧拉格式的证明。

4.1.2 空间离散

根据第 2 章中，我们提到的 $4P_1 - P_1$ 元，这种有限元是基于两套三角形网格和两个有限元空间。根据 [53] 中的几何遗传树中的四叉树结构，速度网格可以通过压力网格全局加密一次得到，如图 2.4 所示。速度单元和压力单元间的 $1 - 1$ 对应关系可以根据几何遗传树结构，相对容易的获得。我们先给出一些符号的定义： \mathcal{T}_h 是速度网格的一个三角剖分，最大的网格尺度 $h = \max_{T \in \mathcal{T}_h} \text{diam}(T)$ ，同样的， $\mathcal{T}_H(H = 2h)$ 是压力网格的三角剖分。速度和压力的有限维空间分别为 $\mathbf{X}^h \subset (H_0^1(\Omega))^2$ 和 $P^H \subset \mathcal{L}^2(\Omega)$ 。那么 Navier-Stokes 问题的全离散形式为：

给定 t_n 时刻的 (\mathbf{u}_h^n, p_H^n) , 来计算 $(\mathbf{u}_h^{n+1}, p_H^{n+1})$ 通过

$$\begin{aligned} \frac{1}{dt} (\mathbf{u}_h^{n+1}, \mathbf{v}_h) + \nu (\nabla \mathbf{u}_h^{n+1}, \nabla \mathbf{v}_h) + (\mathbf{u}_h^n \cdot \nabla \mathbf{u}_h^{n+1}, \mathbf{v}_h) - (p_H^{n+1}, \nabla \mathbf{v}_h) &= \frac{1}{dt} (\mathbf{u}_h^n, \mathbf{v}) \\ (\nabla \cdot \mathbf{u}_h^{n+1}, q_H) &= 0. \end{aligned} \quad (4.20)$$

对所有的 $(\mathbf{v}_h, q_H) \in \mathbf{X}^h \times P^H$.

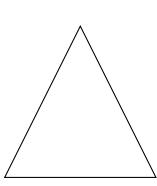


图 4.1: 全局加密一次的网格

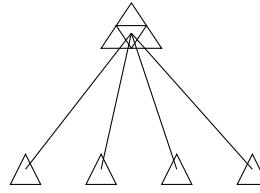
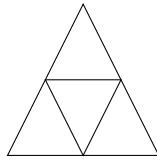


图 4.2: 它的几何遗传树

图 4.3: 几何遗传树结构

4.2 快速 Krylov 求解

令 $(\{\phi_j\}_{j=1}^n, 0)^T$ 和 $(0, \{\phi_j\}_{j=1}^n)^T$ 是速度空间 \mathbf{X}^h 的线性元基函数。同时 $\{\psi_k\}_{k=1}^m$ 为压力空间 P^H 的线性元基函数。 t_{n+1} 时刻速度数值解的分量形式 $\mathbf{u}_h^{n+1} = (u_h^{x,n+1}, u_h^{y,n+1})^T$, $t = t_{n+1}$ 的压力数值解为 \mathbf{p}_H^{n+1} , 可以写为:

$$u_h^{x,n+1} = \sum_{j=1}^{n_u} \alpha_j^{x,n+1} \phi_j, \quad u_h^{y,n+1} = \sum_{j=1}^{n_u} \alpha_j^{y,n+1} \phi_j, \quad p_H^{n+1} = \sum_{k=1}^{n_p} \alpha_k^{p,n+1} \psi_k. \quad (4.21)$$

将 (4.21) 带入弱形式 (4.20) 中, 可以得到一个鞍点问题:

$$\begin{bmatrix} \frac{1}{dt} M + \nu A + N & 0 & B_x^T \\ 0 & \frac{1}{dt} M + \nu A + N & B_y^T \\ B_x & B_y & 0 \end{bmatrix} \begin{bmatrix} \alpha^{x,n+1} \\ \alpha^{y,n+1} \\ \alpha^{p,n+1} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ 0 \end{bmatrix}, \quad (4.22)$$

注意到散度矩阵 $B = [B_x, B_y]$ 为

$$B_x := [B_x]_{kj} = -\left(\psi_k, \frac{\partial \phi_j}{\partial x}\right), k = 1, \dots, n_p, j = 1, \dots, n_u, \quad (4.23)$$

$$B_y := [B_y]_{kj} = -\left(\psi_k, \frac{\partial \phi_j}{\partial y}\right), k = 1, \dots, n_p, j = 1, \dots, n_u. \quad (4.24)$$

因为速度单元上的基函数和压力单元上的基函数并不在同一张网格上, 所以装配矩阵 B 并不是一个显然的过程。这需要根据上面提到的速度单元和压力单元间的 1 – 1 索引, 我们可以仅仅只用速度和压力单元上的线性元函数来装配 B , 拼装 B^T 的过程类似。我们定义 $F_\nu^{n+1} = \frac{1}{dt} M + \nu A + N$, 其中

$$M := [M]_{ij} = (\phi_i, \phi_j), \quad i, j = 1, \dots, n_u, \quad (4.25)$$

$$A := [A]_{ij} = (\nabla \phi_i, \nabla \phi_j), \quad i, j = 1, \dots, n_u, \quad (4.26)$$

$$N := [N]_{ij} = (\mathbf{u}_h^n, \nabla \phi_i, \phi_j), i, j = 1, \dots, n_u. \quad (4.27)$$

为了高效的求解线性方程组 (4.22)，我们用预处理的 GMRES(generalized minimal residual algorithm)作为求解器。在 [37] 中考虑的块三角预处理 \mathcal{P} 定义如下：

$$\mathcal{P} = \begin{pmatrix} F & 0 & B_x^T \\ 0 & F & B_y^T \\ 0 & 0 & S \end{pmatrix}, \quad (4.28)$$

其中 $S = B_x F^{-1} B_x^T + B_y F^{-1} B_y^T$ 是 schur 补矩阵。预处理就是要完成近似矩阵的求逆，实施 \mathcal{P}^{-1} 的过程分为两步，第一步是求解 schur 补的系统，第二步是求解两个纯量的跟 F 有关的系统。直接求解 schur 补的方程组是非常耗时的，因为它系数矩阵里包含一个逆矩阵。所以，在实际的计算中，找到 schur 补的近似矩阵，然后用近似矩阵代替 schur 补矩阵来进行求解。第 2 章中提到 PCD 中近似 schur 补的矩阵为：

$$S_* = A_p F_p^{-1} Q_p, \quad (4.29)$$

其中 A_p, F_p 和 Q_p 全部定义在压力空间上。 Q_p 是压力质量矩阵， A_p 是压力刚度矩阵， F_p 是对流扩散矩阵，它们分别定义为：

$$F_p := [F_p]_{ij} = \nu(\nabla \psi_i, \nabla \psi_j) + (\mathbf{u}_h^n \cdot \nabla \psi_i, \psi_j), \quad i, j = 1, \dots, n_p, \quad (4.30)$$

$$A_p := [A_p]_{ij} = (\nabla \psi_i, \nabla \psi_j) \quad i, j = 1, \dots, n_p. \quad (4.31)$$

令

$$W_p^n := [W_p^n]_{ij} = (\mathbf{u}_h^n \cdot \nabla \psi_i, \psi_j), \quad i, j = 1, \dots, n_p, \quad (4.32)$$

那么 F_p 可以被重新写成：

$$F_p = \nu A_p + W_p^n. \quad (4.33)$$

我们实施 PCD 预处理通过

$$S_*^{-1} = Q_p^{-1} F_p A_p^{-1}. \quad (4.34)$$

精确的 PCD 预处理算子定义为

$$\mathcal{M}^{-1} = \begin{pmatrix} F^{-1} & 0 & B_x^T \\ 0 & F^{-1} & B_y^T \\ 0 & 0 & S_*^{-1} \end{pmatrix} \quad (4.35)$$

我们分两步解释预处理的求解过程：令 $V^d = \mathcal{M}^{-1}V^s$, 其中

$$V^d = (V_x^d, V_y^d, V_p^d)^T, V^s = (V_x^s, V_y^s, V_p^s)^T. \quad (4.36)$$

第一步我们求解：

$$V_p^d = S_*^{-1}V_p^s = Q_p^{-1}F_pA_p^{-1}V_p^s. \quad (4.37)$$

它其中包含两个泊松问题的求解 Q_p^{-1} 和 A_p^{-1} , 所以我们可以用为求解泊松问题设计的代数多重网格求解器来实现求解。第二步我们再用这种多重网格求解器去求解

$$\begin{aligned} V_x^d &= F^{-1}(V_x^s - B_x^T V_p^d), \\ V_y^d &= F^{-1}(V_y^s - B_y^T V_p^d). \end{aligned} \quad (4.38)$$

将 (4.37) 和 (4.38) 中的解合并在一起，就得到了要求解的 V^d .

在实际的计算中，我们对矩阵 F 、 F_p 、 Q_p 和 A_p 的求解，是用固定的几步代数多重网格迭代(algebraic multi-grid)来代替精确求解。这就是迭代的 PCD 方法。我们用的多重网格求解器是在 AFEPack [54] (一个自适应有限元包)中。在 ([37], 第10章)中，数值算例说明了 PCD 预处理的高效。[35] 一文将 PCD 预处理方法应用到 bouyancy 驱动流问题上。在我们的数值实验中，发现如果将 (4.31) 中的

$$F_p = \nu A_p + W_p^n, \quad (4.39)$$

中 A_p 前面的系数 ν 去掉，会发现 GMRES 的迭代步数会减少。我们采用 [39] 中在 Neumann 边界上的处理矩阵 F_p 和 A_p 的方法，来提高求解效率。 F_p 需要在边界 $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ 上满足边界条件

$$\nu \frac{\partial p_h}{\partial n} + (\mathbf{w}_h \cdot \mathbf{n})p_h = 0. \quad (4.40)$$

通过第 2 章中的预处理方法分析知，在出流边界 $\partial\Omega_N$ 上，离散的压力 p_h 需要满足一个齐次的 Dirichlet 边界条件。而在 Ω_D 上，压力 p_h 要满足自然条件 $\frac{\partial p_h}{\partial n} = 0$ ，那就意味着我们在Dirichlet边界上，对 A_p 不采取任何操作。我们知道对于方腔流，在所有的边界上 $\mathbf{w}_h \cdot \mathbf{n} = 0$ ，所以 (4.40) 将会退化成

$$\frac{\partial p_h}{\partial n} = 0. \quad (4.41)$$

这就意味着我们在所有的边界 $\partial\Omega$ 上，对 F_p, A_p 不采取任何操作。在本文中，我们将 PCD 预处理方法应用到移动网格有限元方法，来高效地求解 Navier-Stokes 问题 (4.22)。

4.3 移动网格策略

4.3.1 网格移动的流程

再第3章中我们详细的介绍过移动网格的策略，这一章中简要描述一下。在 $t = t_n$ 时刻，我们获得了数值解 $\mathbf{u}_h^{(n)}, p_H^{(n)}$ 在 t_n 时刻的网格 \mathcal{T}_h^n 上。我们根据 [28] 中的方法，用保持散度为0的差值方法将 \mathcal{T}_h^n 上的数值解差值到 t_{n+1} 时刻的网格 $\mathcal{T}_h^{(n+1)}$ 上。简单的来说，总共分为三步：

step 1 获取控制函数。令 $m = 1/G$ ，其中 G 是控制函数。基于涡量的控制函数

$$G = \sqrt{1 + \alpha|\omega|^\beta}, \quad (4.42)$$

其中 $\omega = \nabla \times \mathbf{u}$, α, β 是两个正的常数。在本文中， $\beta = 2$ 有比较的效果， α 根据不同的问题，选取不同的值。

step 2 获取物理网格上的移动方向。求解

$$\begin{aligned} \nabla_x(m\nabla_x \boldsymbol{\xi}) &= 0, \\ \boldsymbol{\xi}|_{\partial\Omega} &= \boldsymbol{\xi}_b, \end{aligned} \quad (4.43)$$

来获得一个新的逻辑网格 \mathcal{T}_c^* , \mathcal{A}^* 作为它的节点。我们可以得到新的逻辑网格和初始逻辑网格 \mathcal{T}_c^0 (节点为 \mathcal{A}^0) 之间的误差

$$\delta\mathcal{A} = \mathcal{A}^0 - \mathcal{A}^*. \quad (4.44)$$

我们可以根据 $\delta\mathcal{A}$ 来获得物理区域的位移 δX_i , 通常再乘上一个正的常数 μ 避免网格缠接:

$$X_i^{(n+1)} = X_i^{(n)} + \mu\delta X_i. \quad (4.45)$$

step 3 保持散度为 0 的差值。在用移动网格有限元方法来求解不可压流体方程的时候，需要在数值解差值的过程中保持散度为 0。在 [28] 中，数值解在新网格 $\mathcal{T}^{(n+1)}$ 上的重新分布，是通过求解一个类似无粘的 Navier-Stokes 方程。

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial \tau} - \nabla_x \mathbf{u} \cdot \delta \mathbf{x} &= -\nabla \hat{p}, \\ \nabla_x \cdot \mathbf{u} &= 0, \end{aligned} \quad (4.46)$$

其中 $\delta\mathbf{x} := \mathbf{x}^{\text{old}} - \mathbf{x}^{\text{new}}$, $\mathbf{x}^{\text{old}}, \mathbf{x}^{\text{new}}$ 是物理区域上的两组坐标。 τ 是一个虚拟时间，通常取成 1.0。因为对流速度 $\delta\mathbf{x}$ 相对较小。这里 \hat{p} 是一个临时变量，为了跟 (4.1) 中的压力变量区别开来。

(4.46) 的弱形式是：寻找 $(\mathbf{u}_h, \hat{p}_H) \in X_E^h \times P^H$ 使得

$$\begin{aligned} (\partial_\tau \mathbf{u}_h - \nabla_{\mathbf{x}} \mathbf{u}_h \cdot \delta\mathbf{x}, \mathbf{v}_h) &= (\hat{p}_H, \nabla \mathbf{v}_h), \quad \forall \mathbf{v}_h \in X_E^h, \\ (\nabla_{\mathbf{x}} \cdot \mathbf{u}, q_H) &= 0, \quad \forall q_H \in P^H. \end{aligned} \quad (4.47)$$

在本文中，我们用时间层上用显示格式离散 (4.47)：

$$\begin{aligned} \left(\frac{\mathbf{u}_{h,*}^{(n)} - \mathbf{u}_h^{(n)}}{\delta t}, \mathbf{v}_h \right) + \left(\delta\mathbf{x} \cdot \nabla \mathbf{u}_h^{(n)}, \mathbf{v}_h \right) &= \left(\hat{p}_{H,*}^{(n)}, \nabla \mathbf{v}_h \right), \quad \forall \mathbf{v}_h \in X_E^h. \\ (\nabla \cdot \mathbf{u}_{h,*}^n, q_H) &= 0, \quad \forall q_H \in P^H. \end{aligned} \quad (4.48)$$

其中 $\mathbf{u}_h^{(n)}$ 和 $p_H^{(n)}$ 是方程 (4.1) 在 $t = t_n$ 时刻的网格上计算的数值解。 $\mathbf{u}_{h,*}^{(n)}$ 和 $p_{h,*}^{(n)}$ 是在新网格 $\mathcal{T}^{(n+1)}$ 上， t_n 时刻的数值解，是一个中间变量。

4.3.2 用AMG 预处理求解 (4.48)

(4.48) 将会得到一个线性系统，它的系数矩阵为 \mathcal{M}^p ，定义如下：

$$\mathcal{M}^p = \begin{pmatrix} \frac{1}{\delta t} Q_p & 0 & B_x^T \\ 0 & \frac{1}{\delta t} Q_p & B_y^T \\ B_x & B_y & 0 \end{pmatrix} \quad (4.49)$$

据我们所知，矩阵 \mathcal{M}^p 的 schur 补矩阵为：

$$M_S = B_x Q_p^{-1} B_x^T + B_y Q_p^{-1} B_y^T. \quad (4.50)$$

根据第 2 章，对于满足 LBB 条件的混合元近似，当边界条件全部是封闭流体边界时， M_S 是跟压力空间的刚度矩阵 A_p 谱等价的。因此我们可以用 A_p 来作为 Schur 补矩阵的近似矩阵。那么我们可以选取

$$\mathcal{K} = \begin{pmatrix} Q_p & 0 & B_x^T \\ 0 & Q_p & B_y^T \\ 0 & 0 & M_S^* \end{pmatrix}, \quad (4.51)$$

来作为 (4.49) 的块预处理矩阵, 其中 $M_S^* = A_p$ 或者 $M_S^* = \frac{1}{\nu}A_p$ 。我们选择不同的 M_S^* 来作为 schur 补矩阵的近似矩阵, 来对比求解效率的差别。在我们的实际计算中, $\frac{1}{\nu}A_p$ 要比 A_p 效率要高。对于入流/出流问题, A_p 需要在 Neumann 边界上做出一些修改来提高计算效率。在出流边界 $\partial\Omega_N$ 上, 需要满足压力 $p_h = 0$ 。而在 Ω_D 上, 压力 p_h 要满足 $\frac{\partial p_h}{\partial n} = 0$ 。所以我们在 Dirichlet 边界上, 对 A_p 不采取任何操作。细节可以参考 [39]。注意到, 一旦网格发生移动, 所有的矩阵 $M, B_x^T, B_y^T, B_x, B_y, A_p$ 都需要重新构建。

在我们的算法中, PCD 预处理的 GMRES 作为求解线性系统 (4.22) 的求解器。我们定义 GMRES 收敛的停止准则为

$$\|r^{(k)}\| \leq 10^{-6}\|r^{(0)}\| \quad (4.52)$$

其中 $r^{(k)}$ 是线性系统 (4.22) 的残差, r^0 是 (4.22) 的右端项的残差。最后, 为了清楚地表述我们的算法, 我们给出流程图如算法 4.5:

Algorithm 4.5 移动网格有限元方法求解 Navier-Stokes

- 1: 用 AMG 预处理来求解稳态的 Stokes 方程来获得初值值 $\mathbf{u}_h^{(0)}, p_H^{(0)}$ 。
 - 2: **while** $t_n < T$ **do**
 - 3: 在 $\Delta_p^{(n)}$ 上, 用 $\mathbf{u}_h^{(n)}, p_H^{(n)}$ 计算控制函数, 并且通过求解调和映射 (4.43) 来获得新的逻辑网格 ξ^* 。
 - 4: 判断如果 $\xi^* - \xi^{(0)}$ 的 L_2 范数是不是比容忍量小。若是, 则迭代结束, 否则继续5 - 8。
 - 5: 用 $\xi^* - \xi^{(0)}$ 之间的误差来计算压力网格 $\Delta_p^{(n)}$ 在物理区域上的移动量 δx 。
 - 6: 在速度网格 $\Delta_v^{(n)}$ 上, 用AMG预处理方法求解方程 (4.48) 来获取中间变量 $\mathbf{u}_{h,*}^{(n)}, p_{H,*}^{(n)}$ 。
 - 7: 更新压力网格 $\Delta_p^{(n)}$ 到 $\Delta_p^{(n+1)}$ 并且利用几何遗传树结构来同步 $\Delta_v^{(n)}$ 到 $\Delta_v^{(n+1)}$ 。
 - 8: 回到3。
 - 9: 用 AMG 预处理方法来求解 Navier-Stokes 问题 (4.22) , 从而获得新网格 $\Delta_v^{(n+1)}$ 和 $\Delta_p^{(n+1)}$ 上的数值解 $\mathbf{u}_h^{(n+1)}, p_H^{(n+1)}$ 。
 - 10: $n = n + 1$
 - 11: **end while**
-

4.4 数值算例

我们用三个数值算例来展示我们的方法。在实际的计算过程中，我们采用稳态的 Stokes 方程的解作为 Navier-Stokes 方程的初始值，边界条件设置跟 Navier-Stokes 方程一致。在我们的算法中，初始的物理区域和逻辑区域是一致的。网格的移动效果以及数值解在下面展示出来。我们的代码是基于有限元包 AFEPack。

4.4.1 方腔驱动流

我们考虑经典算例：正则化的方腔流。我们的计算区域是 $\Omega = [-1, 1] \times [-1, 1]$ ，粘性系数是 $Re = 2000$ 。Dirichlet 边界条件设置在所有的边界 $\partial\Omega$ 上。在顶部边上，速度 $\mathbf{u} = (1 - x^4, 0)^T$ ，无滑移条件设置在 $\partial\Omega$ 的其它部分。

在我们的移动策略中，我们选取涡量 (4.42) 为控制函数。参数取成 $\alpha = 0.5, \beta = 2.0$ 时网格移动效果比较好。在图 4.4 中展示的是发展到稳态的移动网格以及涡量的等高线。我们可以看出网格集中在顶部边和右边边界，这些地方都是涡量比较大的地方。速度的流速线如图 4.6。在图 4.5 中，我们对比了均匀网格和移动网格下，速度散度大小。结果发现，速度散度值在移动网格下比均匀网格要小一半，这说明了，移动网格的效果比均匀网格要好。

1. 对比求解 (4.22) 中 GMRES 的迭代步数。

从图 4.7 中我们可以看出：在 PCD 预处理中选取 $F_p = A_p + W_p^n$ 需要的 GMRES 的迭代次数比 $F_p = \nu A_p + W_p^n$ 少。我们在表 4.1 中对比了，用 AMG 和 ILU 两种预处理方法求解时，GMRES 的平均迭代步数。我们发现 AMG 预处理比 ILU 效率高很多。

2. 对比求解 (4.48) 中 GMRES 的迭代步数。

表 4.2，我们对比不同的预处理矩阵 M_S^* 对应的平均的 GMRES 迭代步数。结果发现：在求解 4.48 时选取 $M_S^* = \frac{1}{\nu} A_p$ ，GMRES 迭代步数会减少非常明显，相对比 $M_S^* = A_p$ 。

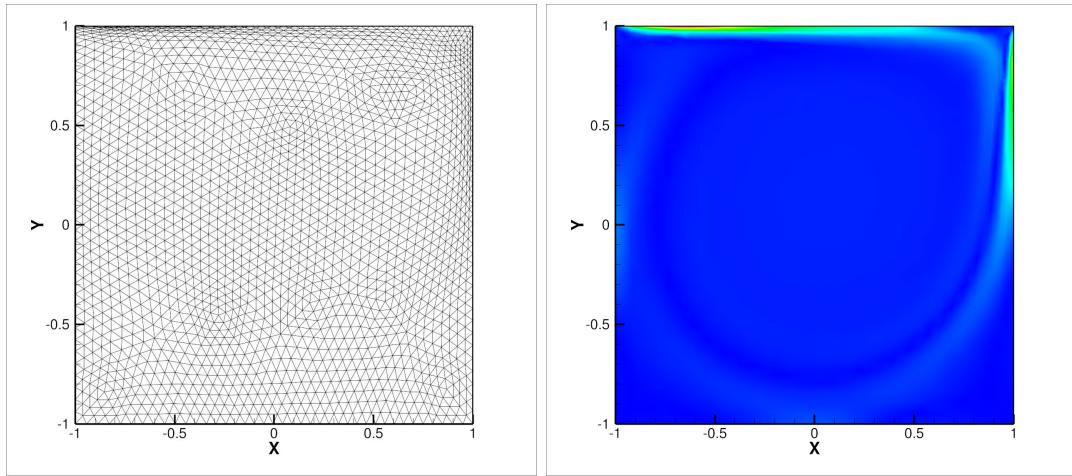


图 4.4: 方腔流: 左: 移动网格, 右: 涡量的等高线, 压力网格 20×20 , $Re = 2000$ 。

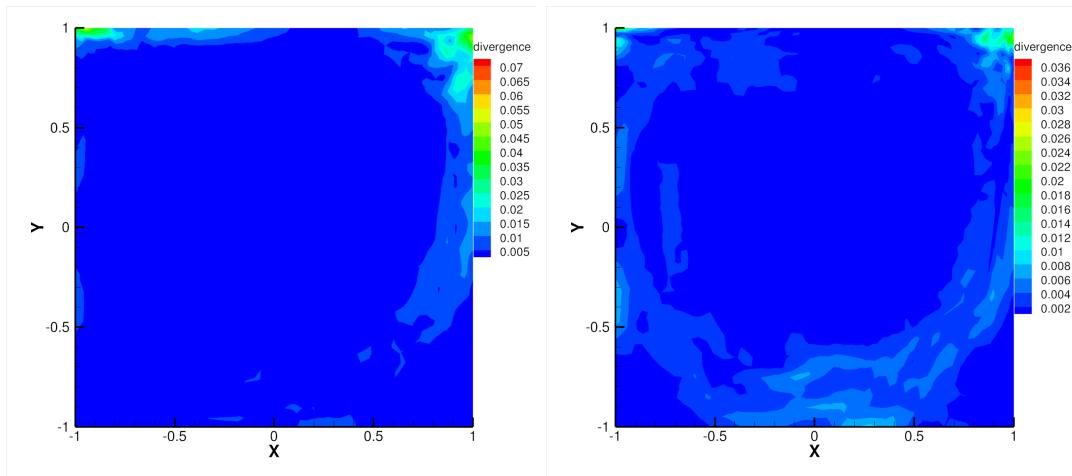
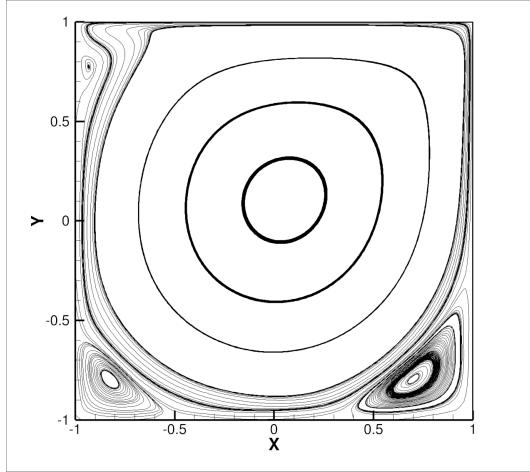
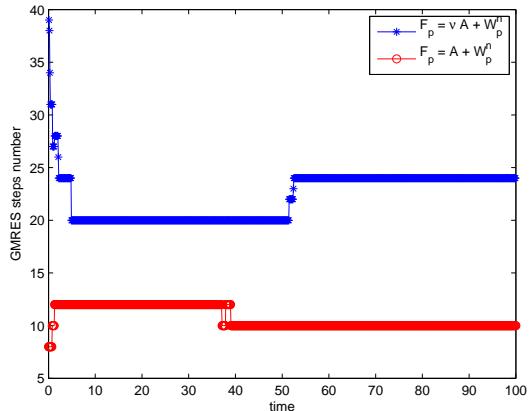


图 4.5: 方腔流, 速度的散度等高线, 左: 均匀网格, 右: 移动网格, 压力网格 20×20 , $Re = 2000$ 。

4.4.2 向后台阶流

这个算例模拟的是流体经过一个向后的台阶。管道的长度是 $l = 5$, 在入流边界 $x = -1, y \in (0, -1)$ 设置的是 Poiseuille 流条件 $\mathbf{u} = (1 - y^2, 0)^T$ 。在管道的顶端和底端设置的无滑移边界条件 $\mathbf{u} = (0, 0)^T$, 自然条件设置在出流边界 $x = 5, y \in (-1, 1)$ 上。我们选取粘性系数为 $\nu = 0.02$, 那么当 $t \rightarrow \infty$ 时, 流体

图 4.6: 方腔流: 流速线, 压力网格 20×20 , $Re = 2000$ 。图 4.7: 方腔流: 对比在 PCD 预处理中用不同的 F_p 来求解 (4.22) 时, GMRES 的迭代步数, 压力网格 20×20 , $Re = 2000$ 。

趋向于稳态。

我们选取 (4.42) 为控制函数, 控制函数中的参数是 $\alpha = 2.0, \beta = 2.0$ 。我们知道奇异性将会出现在流体扩展的拐角处, 因此这里需要更多的网格。在图 4.8 中, 网格集中在凹进去的拐角处, 这跟我们的设想一致。

要满足 CFL 条件, 我们的计算时间步长在 0.008 左右。对比用不同的 M_S^* 求解 (4.48) 的 GMRES 迭代步数在图 4.9 中给出。从中可以发现选取 $M_S^* = \frac{1}{\nu} A_p$

压力网格	GMRES 的平均迭代次数	
	AMG	ILU
20×20	10.8	107.5
40×40	16.01	307.33
80×80	27.18	> 500

表 4.1: 方腔流: 对比用 AMG 和 ILU 预处理来求解 (4.22) 时, GMRES 的迭代步数, $Re = 2000$.

压力网格	GMRES 迭代次数	
	$M_S^* = \frac{1}{\nu} A_p$	$M_S^* = A_p$
20×20	4.96	13.19
40×40	10.99	42.36

表 4.2: 方腔流: 对比用不同的预处理矩阵 M_S^* , 求解 (4.48) 时, GMRES 平均迭代次数, $Re = 2000$.

比 $M_S^* = A_p$ 更高效。

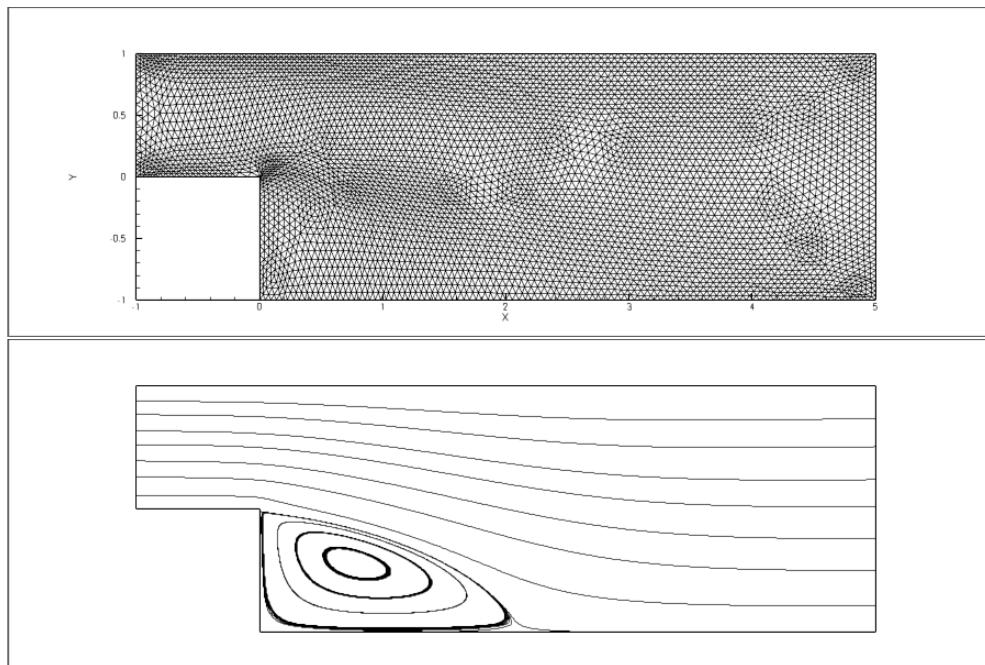


图 4.8: 向后台阶流, 上: 移动网格, 下: 流速线, $t = 100s$, 粘性系数 $\nu = 0.02$.

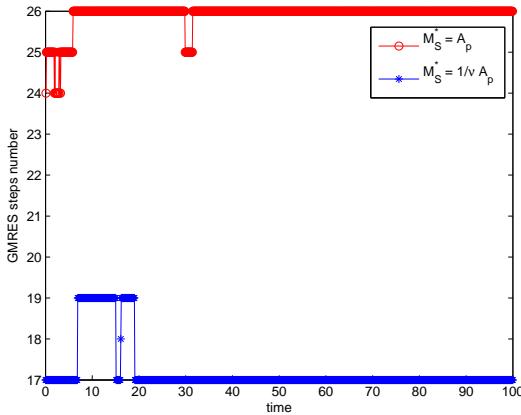


图 4.9: 向后台阶流: 对比不同的 M_S^* 求解 (4.48) 时, GMRES 随时间的迭代次数, $\nu = 0.02$

4.4.3 圆柱绕流

这个算例是第 3 章的一个算例。圆形障碍的圆心是 $(0, 0)$, 半径是 $r = 0.3$, 令粘性系数 $\nu = 1/300$, 通过简单计算雷诺数 $Re = \frac{2rU_m}{\nu} = 240$, 其中 U_m 是入流平均速度。计算区域是 $\Omega = [-1, 5] \times [-1, 1]$ 。在入流边界 $x = -1$, 设置具有 Poiseuille 性质的边界条件 $\mathbf{u} = (1 - y^2, 0)^T$ 。在管道的上边界和下边界, 设定条件 $\mathbf{u} = (0, 0)^T$ 。自然条件设置在出流边界 $x = 5$ 上。

在我们的移动策略中, 用户可以设置控制函数 (4.42) 的参数 α 和 β 。 α 的值越大, 网格聚集的程度就越大。从图 4.10 中, 我们可以看出 $\alpha = 5$ 时 GMRES 的迭代步数要比 $\alpha = 1$ 时的迭代次数多。在表 4.3 中, 我们对比了在求解 (4.22) 和 (4.48) 时, 预处理方法中选取不同的 F_p 和 M_S^* , GMRES 的平均迭代次数。结果发现, 选取 $F_p = \nu A + W_p^n$, GMRES 的平均迭代次数会相对减少 10 次。同时, 选取 $M_S^* = \frac{1}{\nu} A_p$ 比 $M_S^* = A_p$ 迭代次数会减少一半, 因此选取 $M_S^* = \frac{1}{\nu} A_p$ 更高效。

在图 4.11 中, 展示了 $t = 2s$ 时的网格移动效果。可以发现, 网格明显的聚集在圆形障碍物的周围。据我们所知, 当选取适当的雷诺数时, 会出现涡街现象, 参考 [32] 一文, 正如图 4.12 所展示的。

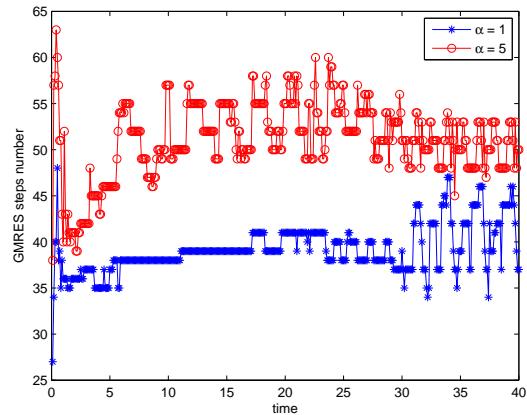


图 4.10: 圆柱绕流: 对比不同 α , 用 PCD 预处理求解 (4.22), GMRES 的迭代次数, $Re = 240$.

线性系统	GMRES 平均迭代次数			
	$F_p = A + W_p^n$	$F_p = \nu A + W_p^n$	$M_S^* = \frac{1}{\nu} A_p$	$M_S^* = A_p$
(4.22)	51.06	45.13	61.06	61.08
(4.48)	22.25	40.35	23.77	40.78

表 4.3: 圆柱绕流: 对比用不同的预处理矩阵 F_p 和 M_S^* 分别求解 (4.22) 和 (4.48) 时, GMRES 的平均迭代次数, $\alpha = 5, Re = 240$ 。

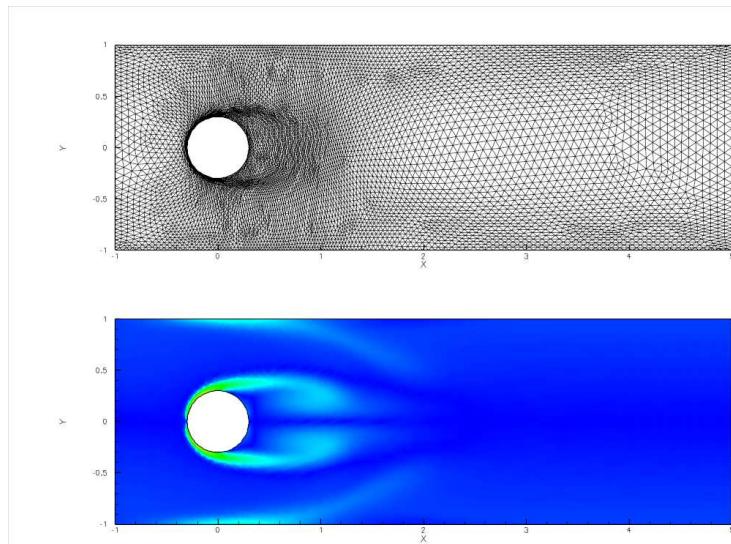


图 4.11: 圆柱绕流, 上: 移动网格, 下: 涡量等高线, $t = 2s$, $Re = 240$ 。

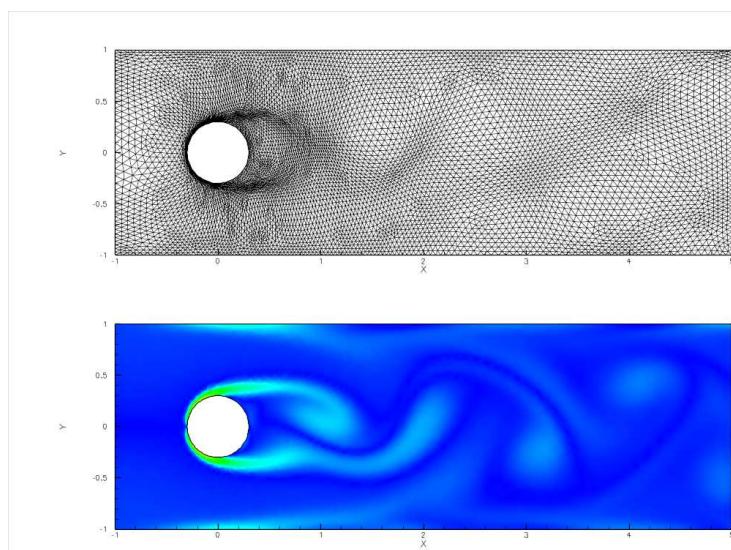


图 4.12: 圆柱绕流, 上: 移动网格, 下: 涡量等高线, $t = 40s$, $Re = 240$ 。

参考文献

- [1] 王烈衡and 许学军, 有限元方法的数学基础, 科学出版社, 2004.
- [2] Bai Z.Z., Structured preconditioners for nonsingular matrices of block two-by-two structures, *Mathematics of Computation*, 2006, 75(254): 791–815.
- [3] Bai Z.Z. and Ng M.K., On inexact preconditioners for nonsymmetric matrices, *SIAM Journal on Scientific Computing*, 2005, 26(5): 1710–1724.
- [4] Baines M.J., *Moving finite elements*, Oxford University Press, Inc., 1994.
- [5] Benzi M., Golub G.H. and Liesen J., Numerical solution of saddle point problems, *Acta numerica*, 2005, 14: 1–137.
- [6] Benzi M., Ng M., Niu Q. and Wang Z., A relaxed dimensional factorization preconditioner for the incompressible navier–stokes equations, *Journal of Computational Physics*, 2011, 230(16): 6185–6202.
- [7] Benzi M. and Olshanskii M.A., An augmented lagrangian-based approach to the oseen problem, *SIAM Journal on Scientific Computing*, 2006, 28(6): 2095–2113.
- [8] Benzi M. and Tüma M., A robust incomplete factorization preconditioner for positive definite matrices, *Numerical Linear Algebra with Applications*, 2003, 10(5-6): 385–400.
- [9] Bercovier M. and Pironneau O., Error estimates for finite element method solution of the stokes problem in the primitive variables, *Numerische Mathematik*, 1979, 33(2): 211–224.
- [10] Berrone S. and Marro M., Space–time adaptive simulations for unsteady navier–stokes problems, *Computers & Fluids*, 2009, 38(6): 1132–1144.

- [11] Bochev P.B., Dohrmann C.R. and Gunzburger M.D., Stabilization of low-order mixed finite elements for the stokes equations, SIAM Journal on Numerical Analysis, 2006, 44(1): 82–101.
- [12] Boyle J., Mihajlovic M. and Scott J., Hsl_mi20: an efficient amg preconditioner, Tech. rep., Citeseer, 2007.
- [13] Boyle J., Mihajlović M. and Scott J., Hsl_mi20: an efficient amg preconditioner for finite element problems in 3d, International journal for numerical methods in engineering, 2010, 82(1): 64–98.
- [14] Brackbill J.U., An adaptive grid with directional control, Journal of Computational Physics, 1993, 108(1): 38–50.
- [15] Braess D. and Hackbusch W., A new convergence proof for the multigrid method including the v-cycle, SIAM Journal on Numerical Analysis, 1983, 20(5): 967–975.
- [16] Brenner S. and Scott R., The mathematical theory of finite element methods, vol. 15, Springer Science & Business Media, 2007.
- [17] Brezina M., Falgout R., MacLachlan S., Manteuffel T., McCormick S. and Ruge J., Adaptive smoothed aggregation (α sa) multigrid, SIAM review, 2005, 47(2): 317–346.
- [18] Brezzi F. and Fortin M., Mixed and hybrid finite element methods, vol. 15, Springer Science & Business Media, 2012.
- [19] Budd C.J., Huang W. and Russell R.D., Adaptivity with moving grids, Acta Numerica, 2009, 18: 111–241.
- [20] Cai M., Mu M. and Xu J., Numerical solution to a mixed navier-stokes/darcy model by the two-grid approach, SIAM Journal on Numerical Analysis, 2009, 47(5): 3325–3338.

- [21] Cao W., Huang W. and Russell R.D., Anr-adaptive finite element method based upon moving mesh pdes, *Journal of Computational Physics*, 1999, 149(2): 221–244.
- [22] Cheng X.l., Han W. and Huang H.c., Analysis of some mixed elements for the stokes problem, *Journal of computational and applied mathematics*, 1997, 85(1): 19–35.
- [23] Cuvelier C., Segal A. and Van Steenhoven A.A., *Finite element methods and Navier-Stokes equations*, vol. 22, Springer Science & Business Media, 1986.
- [24] Danaila I., Moglan R., Hecht F. and Le Masson S., A newton method with adaptive finite elements for solving phase-change problems with natural convection, *Journal of Computational Physics*, 2014, 274: 826–840.
- [25] de Boor C., Good approximation by splines with variable knots. II, in: Conference on the Numerical Solution of Differential Equations, Lecture Notes in Mathematics, vol. 363 of Lecture Notes in Mathematics, Berlin: Springer-Verlag, 1974 pp. 12–20.
- [26] Dean E.J. and Glowinski R., On some finite element methods for the numerical simulation of incompressible viscous flow, *Incompressible computational fluid dynamics- Trends and advances*, Cambridge, United Kingdom and New York, Cambridge University Press, 1993,, 1993, pp. 17–65.
- [27] Di Y., Li R. and Tang T., A general moving mesh framework in 3d and its application for simulating the mixture of multi-phase flows, *Commun. Comput. Phys*, 2008, 3(3): 582–602.
- [28] Di Y., Li R., Tang T. and Zhang P., Moving mesh finite element methods for the incompressible navier–stokes equations, *SIAM Journal on Scientific Computing*, 2005, 26(3): 1036–1056.
- [29] Di Y., Li R., Tang T. and Zhang P., Moving mesh finite element methods for the incompressible Navier-Stokes equations, *SIAM Journal on Scientific Computing*, 2005, 26(3): 1036–1056.

- [30] Di Y., Li R., Tang T. and Zhang P., Level set calculations for incompressible two-phase flows on a dynamically adaptive grid, *Journal of Scientific Computing*, 2007, 31(1): 75–98.
- [31] Dvinsky A.S., Adaptive grid generation from harmonic maps on Riemannian manifolds, *Journal of Computational Physics*, 1991, 95(2): 450–476.
- [32] Dyke M.V., *An album of fluid motion*, 1982.
- [33] Ebeida M.S., Davis R.L. and Freund R.W., Unsteady incompressible flow simulation using galerkin finite elements with spatial/temporal adaptation, in: 47th AIAA Aerospace Sciences Meeting, Orlando, FL, 2009 .
- [34] Elman H., Howle V.E., Shadid J., Silvester D. and Tuminaro R., Least squares preconditioners for stabilized discretizations of the navier-stokes equations, *SIAM Journal on Scientific Computing*, 2007, 30(1): 290–311.
- [35] Elman H., Mihajlović M. and Silvester D., Fast iterative solvers for buoyancy driven flow problems, *Journal of Computational Physics*, 2011, 230(10): 3900–3914.
- [36] Elman H. and Silvester D., Fast nonsymmetric iterations and preconditioning for navier-stokes equations, *SIAM Journal on Scientific Computing*, 1996, 17(1): 33–46.
- [37] Elman H.C., Silvester D.J. and Wathen A.J., *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, 2005.
- [38] Elman H.C., Silvester D.J. and Wathen A.J., *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Oxford University Press (UK), 2014.
- [39] Elman H.C. and Tuminaro R.S., Boundary conditions in approximate commutator preconditioners for the navier-stokes equations, *Electronic Transactions on Numerical Analysis*, 2009, 35: 257–280.

- [40] Fischer B., Ramage A., Silvester D.J. and Wathen A.J., Minimum residual methods for augmented systems, *BIT Numerical Mathematics*, 1998, 38(3): 527–543.
- [41] Girault V. and Raviart P.A., *Finite element methods for Navier-Stokes equations: theory and algorithms*, vol. 5, Springer Science & Business Media, 2012.
- [42] Gresho P. and Sani R., *Incompressible flow and the finite element method. volume 2: Incompressible flow and finite element*, 1998.
- [43] Gunzburger M.D., *Finite element methods for viscous incompressible flows: a guide to theory, practice, and algorithms*, Elsevier, 2012.
- [44] Hamilton R.S., *Harmonic maps of manifolds with boundary*, vol. 471, Springer, 2006.
- [45] Harlow F.H., Welch J.E. et al., Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Physics of fluids*, 1965, 8(12): 2182.
- [46] He Y., Two-level method based on finite element and crank-nicolson extrapolation for the time-dependent navier-stokes equations, *SIAM Journal on Numerical Analysis*, 2003, 41(4): 1263–1285.
- [47] Hirt C., Amsden A.A. and Cook J., An arbitrary lagrangian-eulerian computing method for all flow speeds, *Journal of Computational Physics*, 1974, 14(3): 227–253.
- [48] Huang W., Ren Y. and Russell R.D., Moving mesh methods based on moving mesh partial differential equations, *Journal of Computational Physics*, 1994, 113(2): 279–290.
- [49] Huang W., Ren Y. and Russell R.D., Moving mesh partial differential equations (MMPDES) based on the equidistribution principle, *SIAM Journal on Numerical Analysis*, 1994, 31(3): 709–730.

- [50] Kay D., Loghin D. and Wathen A., A preconditioner for the steady-state navier–stokes equations, SIAM Journal on Scientific Computing, 2002, 24(1): 237–256.
- [51] Klawonn A. and Starke G., Block triangular preconditioners for nonsymmetric saddle point problems: field-of-values analysis, Numerische Mathematik, 1999, 81(4): 577–594.
- [52] Li J., He Y. and Chen Z., Performance of several stabilized finite element methods for the stokes equations based on the lowest equal-order pairs, Computing, 2009, 86(1): 37–51.
- [53] Li R., On multi-mesh h-adaptive methods, Journal of Scientific Computing, 2005, 24(3): 321–341.
- [54] Li R. and Liu W., AFEPack, <http://circus.math.pku.edu.cn/AFEPack>.
- [55] Li R., Liu W., Tang T. and Zhang P., Moving mesh finite element methods based on harmonic maps, 2001.
- [56] Li R., Tang T. and Zhang P., Moving mesh methods in multiple dimensions based on harmonic maps, Journal of Computational Physics, 2001, 170(2): 562–588.
- [57] Li R., Tang T. and Zhang P., A moving mesh finite element algorithm for singular problems in two and three space dimensions, Journal of Computational Physics, 2002, 177(2): 365–393.
- [58] Mu M. and Xu J., A two-grid method of a mixed stokes-darcy model for coupling fluid flow with porous media flow, SIAM journal on numerical analysis, 2007, 45(5): 1801–1813.
- [59] Peaceman D.W. and Rachford Jr H.H., The numerical solution of parabolic and elliptic differential equations, Journal of the Society for industrial and Applied Mathematics, 1955, 3(1): 28–41.

- [60] Ruge J. and Stüben K., Algebraic multigrid, *Multigrid methods*, 1987, 3(13): 73–130.
- [61] Saad Y., *Iterative methods for sparse linear systems*, Siam, 2003.
- [62] Saad Y. and Schultz M.H., Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on scientific and statistical computing*, 1986, 7(3): 856–869.
- [63] Schoen R. and Yau S.T., On univalent harmonic maps between surfaces, *Inventiones Mathematicae*, 1978, 44: 265–278.
- [64] Shen L. and Xu J., On a schur complement operator arisen from navier-stokes equations and its preconditioning, *LECTURE NOTES IN PURE AND APPLIED MATHEMATICS*, 1999, pp. 481–490.
- [65] Silvester D.J. and Wathen A.J., Fast & robust solvers for time-discretised incompressible navier-stokes equations, *PITMAN RESEARCH NOTES IN MATHEMATICS SERIES*, 1996, pp. 154–168.
- [66] Simo J. and Armero F., Unconditional stability and long-term behavior of transient algorithms for the incompressible navier-stokes and euler equations, *Computer Methods in Applied Mechanics and Engineering*, 1994, 111(1): 111–154.
- [67] Smith A. and Silvester D., Implicit algorithms and their linearization for the transient incompressible navier-stokes equations, *IMA journal of numerical analysis*, 1997, 17(4): 527–545.
- [68] Tang T. and Xu J., *Adaptive Computations: Theory and Algorithms*, Science Press, 2007.
- [69] Van der Vorst H.A., *Iterative Krylov methods for large linear systems*, vol. 13, Cambridge University Press, 2003.
- [70] Wabro M., Coupled algebraic multigrid methods for the oseen problem, *Computing and Visualization in Science*, 2004, 7(3-4): 141–151.

- [71] Wang D. and Wang X.P., A three-dimensional adaptive method based on the iterative grid redistribution, *Journal of Computational Physics*, 2004, 199: 423–436.
- [72] Wang H., Li R. and Tang T., Efficient computation of dendritic growth with r-adaptive finite element methods, *Journal of Computational Physics*, 2008, 227(12): 5984–6000.
- [73] Weizhang H. and Russell R.D., *Adaptive moving mesh methods*, vol. 174, Springer Science & Business Media, 2010.
- [74] Winslow A.M., Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh., *Journal of Computational Physics*, 1966, 135(2): 128–138.
- [75] Xiao-liang C. and Shaikh A.W., Analysis of the iterative penalty method for the stokes equations, *Applied mathematics letters*, 2006, 19(10): 1024–1028.
- [76] Xu J., Iterative methods by space decomposition and subspace correction, *SIAM Review*, 1992, 34(4): 581–613.
- [77] Zheng H., Hou Y. and Shi F., Adaptive variational multiscale methods for incompressible flow based on two local gauss integrations, *Journal of Computational Physics*, 2010, 229(19): 7030–7041.
- [78] Zheng H., Hou Y. and Shi F., A posteriori error estimates of stabilization of low-order mixed finite elements for incompressible flow, *SIAM Journal on Scientific Computing*, 2010, 32(3): 1346–1360.