

DOI: 10.13718/j.cnki.xdzk.2019.03.010

Navier-Stokes 方程的亚格子模型后处理混合有限元方法^①

张琪慧, 尚月强

西南大学 数学与统计学院, 重庆 400715

摘要: 提出并考察了 3 种基于亚格子模型的后处理混合有限元方法, 其主要思想是: 第一步在粗网格上求解带有亚格子模型稳定项的 Navier-Stokes 方程, 得到最后时刻 T 的有限元解 \mathbf{u}_H ; 第二步在最后时刻 T , 对第一步所得解 \mathbf{u}_H 进行后处理, 主要通过细网格上(或用高阶元)分别求解带有亚格子模型稳定项的 Stokes 问题、Newton 问题或者 Ossen 问题. 实验结果表明: 在选取适当的稳定化参数和网格尺寸条件下, 3 种稳定化的后处理有限元方法提高了稳定化的混合有限元解的精确度, 并且收敛阶较标准的有限元方法明显提高了一阶. 从计算时间看, 除 $\nu = 1$ 以外, 在其它情况下稳定化的 Newton 型后处理花费的时间相对较多, 而稳定化的 Ossen 型后处理花费的时间相对较少. 从精确度来看, Newton 型后处理和 Ossen 型后处理方法所得速度的 H^1 -范误差和压力的 L^2 -范误差比 Stokes 型后处理方法更有效.

关键词: Navier-Stokes 方程; 后处理; 亚格子稳定化; 有限元方法

中图分类号: O241.82

文献标志码: A

文章编号: 1673-9868(2019)03-0067-08

Navier-Stokes 方程是描述流体运动规律的一类典型的非线性方程, 其研究对人们认识和控制湍流格外重要, 并被广泛应用于天气、海流等方面. 近几十年来, 用有限元方法数值求解 Navier-Stokes 方程吸引了许多研究者^[1-3]. 而后处理的思想最初是在文献[4]中基于 Fourier 谱方法提出的, 文献[5]将该方法应用于混合有限元方法提出了后处理的混合有限元方法. 该方法主要分为两步: 第一步, 在粗网格上求解一个非线性系统, 得到最后时刻 T 的有限元解 \mathbf{u}_H ; 第二步, 在最后时刻 T , 对第一步所得解进行后处理, 主要通过细网格上(或用高阶元)求解一个线性的 Stokes 问题^[6], 或一个线性 Ossen 类型的问题^[7], 或者一个 Newton 类型的问题^[8]. 理论分析和数值实验表明: 在选择合适的网格尺寸条件下, 后处理的混合有限元方法与标准的有限元方法相比收敛精度提高了一阶. 由于完全非线性问题要在粗网格上求解, 而对流占优的流体具有不稳定性, 所以用后处理的方法模拟大雷诺数流问题具有一定的挑战性. 本文中, 我们把前面提到的 3 种后处理的有限元方法和亚格子稳定化方法^[9]相结合, 提出了 3 种稳定化的后处理混合有限元方法.

1 亚格子模型后处理混合有限元方法

定义 1 设 Ω 是 \mathbb{R}^d ($d = 2, 3$) 中具有齐次 Dirichlet 边界条件的有界区域, 我们考虑下面的

① 收稿日期: 2018-03-01

基金项目: 重庆市基础科学与前沿技术研究专项项目(cstc2016jcyjA0348).

作者简介: 张琪慧(1994-), 女, 硕士研究生, 主要从事偏微分方程数值解的研究.

通信作者: 尚月强, 博士, 教授.

Navier-Stokes 方程:

$$\mathbf{u}_t - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{f}, \text{ in } \Omega \times (0, T] \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \text{ in } \Omega \times (0, T] \quad (2)$$

$$\mathbf{u} = 0, \text{ on } \partial\Omega \times (0, T] \quad (3)$$

$$\mathbf{u} |_{t=0} = \mathbf{u}_0, \text{ in } \Omega \quad (4)$$

其中: $\mathbf{u}: \Omega \rightarrow \mathbb{R}^2$ 表示速度矢量, $p: \Omega \rightarrow \mathbb{R}$ 表示压力, $\mathbf{f}: \Omega \rightarrow \mathbb{R}^2$ 表示流体驱动的体积力, $\nu > 0$ 为流体粘性系数, \mathbf{u}_0 是使得 $\nabla \cdot \mathbf{u}_0 = 0$ 的初始速度, 并且 $\mathbf{u}_t = \partial \mathbf{u} / \partial t$, 定常数 T 表示最后时刻.

下面针对定义 1 提出的数学问题, 我们引进两个 Hilbert 空间:

$$H = \{\mathbf{u} \in (L^2(\Omega))^d, | \operatorname{div}(\mathbf{u}) = 0, \mathbf{u} \cdot \mathbf{n}|_{\partial\Omega} = 0\}$$

$$V = \{\mathbf{u} \in (H_0^1(\Omega))^d, | \operatorname{div}(\mathbf{u}) = 0\}$$

对 H, V 分别赋予 $L^2(\Omega)^d$ 空间和 $H_0^1(\Omega)^d$ 空间的内积. 用 $\|\cdot\|_l$ 表示 Sobolev 空间 $W^{l,q}(\Omega)^d$ 的范数, $\|\cdot\|_{-l}$ 表示其对偶空间的范数. 后面还会涉及到商空间 $H^l(\Omega)/\mathbb{R}$, 其范数定义为:

$$\|p\|_{H^l/\mathbb{R}} = \inf\{\|p+c\|_l | c \in \mathbb{R}\}$$

设 $\mathcal{T}_h = (\tau_i^h, \phi_i^h)_{i \in I_h}$ ($h > 0$) 是区域 Ω_h 的准均匀网格剖分, 其中 h 是单元 $\tau_i^h \in \mathcal{T}_h$ 的最大直径, $\phi_i^h: \tau_0 \rightarrow \tau_i^h$ 为单射. 当 $r \geq 2$ 时, 基于网格剖分 \mathcal{T}_h , 我们定义下列有限元空间:

$$\hat{S}_{h,r} = \{\chi_h \in C^0(\overline{\Omega_h}) | \chi_h|_{\tau_i^h} \circ \phi_i^h \in P_{r-1}(\tau_0)\} \subset H^1(\Omega_h)$$

$$\bar{S}_{h,r} = \{\chi_h \in C^0(\overline{\Omega_h}) | \chi_h|_{\tau_i^h} \circ \phi_i^h \in P_{r-1}(\tau_0), \chi_h(x) = 0 \forall x \in \partial\Omega_h\} \subset H_0^1(\Omega_h)$$

其中 $P_{r-1}(\tau_0)$ 表示定义在 τ_0 上的至多 $r-1$ 次多项式空间.

接下来, 我们考虑 Hood-Taylor 元 $(X_{h,r}, Q_{h,r-1})$, 其中

$$X_{h,r} = (\bar{S}_{h,r})^d, Q_{h,r-1} = \hat{S}_{h,r-1} \cap L^2(\Omega_h)/\mathbb{R} \quad r \geq 3$$

并且这个有限元对 $(X_{h,r}, Q_{h,r-1})$ 满足 inf-sup 条件^[6].

根据非线性项 $b(\cdot, \cdot, \cdot)$ 的定义^[9], 我们具体给出离散的非线性项 $b_h(\cdot, \cdot, \cdot)$ 为:

$$b_h(\mathbf{u}_h, \mathbf{v}_h, \boldsymbol{\phi}_h) = ((\mathbf{u}_h \cdot \nabla) \mathbf{v}_h, \boldsymbol{\phi}_h) + 1/2(\operatorname{div}(\mathbf{u}_h) \mathbf{v}_h, \boldsymbol{\phi}_h) \quad \forall \mathbf{u}_h, \mathbf{v}_h, \boldsymbol{\phi}_h \in X_{h,r} \subset H_0^1(\Omega)^d$$

亚格子模型是基于一个椭圆型投影算子 $\Pi_h: H_0^1(\Omega)^d \rightarrow R_1 = \{\mathbf{v} \in H_0^1(\Omega)^d: \mathbf{v}|_{\tau_i^h} \in (p_1)^d, \forall \tau_i^h \in T_h\}$ 所定义, 其中算子 Π_h 的定义为:

$$(\nabla \Pi_h \mathbf{u}, \nabla \mathbf{v}) = (\nabla \mathbf{u}, \nabla \mathbf{v}) \quad \forall \mathbf{u} \in R_1$$

下面根据投影算子 Π_h , 定义亚格子稳定项为:

$$G(\mathbf{u}, \mathbf{v}) = \alpha(\nabla(I - \Pi_h) \mathbf{u}, \nabla(I - \Pi_h) \mathbf{v}) \quad \forall \mathbf{u}, \mathbf{v} \in H_0^1(\Omega)^d$$

其中 $\alpha > 0$ 是稳定化参数, 通常 $\alpha = o(h^s)$ (s 是与所使用有限元空间的阶数有关的参数).

下面在 T 时刻逼近问题(1)–(4). 基于 3 种后处理的有限元方法^[6–8] 和亚格子稳定化方法^[9], 我们的亚格子稳定化的后处理有限元方法具体如下:

算法 1 基于 Stokes 问题的稳定化后处理有限元方法.

1) 给定 \mathbf{u}_0^h 逼近 $\mathbf{u}(0)$, 寻找 $\mathbf{u}_h: [0, T] \rightarrow X_{h,r}$ 和 $Q_{h,r-1}$ 使得:

$$(\dot{\mathbf{u}}_h, \boldsymbol{\phi}_h) + \nu(\nabla \mathbf{u}_h, \nabla \boldsymbol{\phi}_h) + b_h(\mathbf{u}_h, \mathbf{u}_h, \boldsymbol{\phi}_h) + (\nabla p_h, \boldsymbol{\phi}_h) + G(\mathbf{u}_h, \boldsymbol{\phi}_h) = (\mathbf{f}, \boldsymbol{\phi}_h) \quad \forall \boldsymbol{\phi}_h \in X_{h,r} \quad (5)$$

$$(\nabla \cdot \mathbf{u}_h, \boldsymbol{\psi}_h) = 0 \quad \forall \boldsymbol{\psi}_h \in Q_{h,r-1} \quad (6)$$

其中 $\dot{\mathbf{u}}_h = \partial \mathbf{u}_h / \partial t$, 并且通常取 $\mathbf{u}_0^h = S_h(\mathbf{u}_0)$, S_h 是离散 Stokes 投影^[6].

2) 对离散的速度和压力 $(\mathbf{u}_h(T), p_h(T))$ 后处理, 寻找 $(\tilde{\mathbf{u}}_h, \tilde{p}_h) \in (\tilde{X}, \tilde{Q})$ 使得:

$$\nu(\nabla \tilde{\mathbf{u}}_h, \nabla \tilde{\boldsymbol{\phi}}) + (\nabla \tilde{p}_h, \tilde{\boldsymbol{\phi}}) + G(\tilde{\mathbf{u}}_h, \tilde{\boldsymbol{\phi}}) = (\mathbf{f}, \tilde{\boldsymbol{\phi}}) - b_h(\mathbf{u}_h(T), \mathbf{u}_h(T), \tilde{\boldsymbol{\phi}}) - (\dot{\mathbf{u}}_h(T), \tilde{\boldsymbol{\phi}}) \quad \forall \tilde{\boldsymbol{\phi}} \in \tilde{X}$$

$$(\nabla \cdot \tilde{\mathbf{u}}_h, \tilde{\boldsymbol{\psi}}) = 0 \quad \forall \tilde{\boldsymbol{\psi}} \in \tilde{\mathcal{Q}}$$

其中后处理的有限元 $(\tilde{\mathcal{X}}, \tilde{\mathcal{Q}})$ 有以下两种取法:

方法 1 更细网格上的同阶 Hood-Taylor 元 $(\tilde{\mathcal{X}}, \tilde{\mathcal{Q}}) = (X_{h,r}, Q_{h,r-1})$, $r \geq 3$, $\tilde{h} < h$.

方法 2 相同网格上的高阶 Hood-Taylor 元 $(\tilde{\mathcal{X}}, \tilde{\mathcal{Q}}) = (X_{h,r+1}, Q_{h,r})$, $r \geq 3$, $\tilde{h} = h$.

算法 2 基于 Newton 型问题的稳定化后处理有限元方法.

1) 给定 $\mathbf{u}^h(0)$ 逼近 $\mathbf{u}(0)$, 寻找 $\mathbf{u}_h: [0, T] \rightarrow X_{h,r}$ 和 $\mathbf{p}_h: [0, T] \rightarrow Q_{h,r-1}$, 使得 $(\mathbf{u}_h, \mathbf{p}_h)$ 满足 (5) – (6) 式;

2) 对离散的速度和压力 $(\mathbf{u}_h(T), \mathbf{p}_h(T))$ 后处理, 寻找 $(\tilde{\mathbf{u}}_h, \tilde{\mathbf{p}}_h) \in (\tilde{\mathcal{X}}, \tilde{\mathcal{Q}})$ 使得:

$$\begin{aligned} \nu(\nabla \tilde{\mathbf{u}}_h, \nabla \tilde{\boldsymbol{\phi}}) + b_h(\tilde{\mathbf{u}}_h, \mathbf{u}_h(T), \tilde{\boldsymbol{\phi}}) + b_h(\mathbf{u}_h(T), \tilde{\mathbf{u}}_h, \tilde{\boldsymbol{\phi}}) + (\nabla \tilde{\mathbf{p}}_h, \tilde{\boldsymbol{\phi}}) + G(\tilde{\mathbf{u}}_h, \tilde{\boldsymbol{\phi}}) = \\ (f, \tilde{\boldsymbol{\phi}}) + b_h(\mathbf{u}_h(T), \mathbf{u}_h(T), \tilde{\boldsymbol{\phi}}) - (\dot{\mathbf{u}}_h(T), \tilde{\boldsymbol{\phi}}) \quad \forall \tilde{\boldsymbol{\phi}} \in \tilde{\mathcal{X}} \\ (\nabla \cdot \tilde{\mathbf{u}}_h, \tilde{\boldsymbol{\psi}}) = 0 \quad \forall \tilde{\boldsymbol{\psi}} \in \tilde{\mathcal{Q}} \end{aligned}$$

算法 3 基于 Ossen 型问题的稳定化后处理有限元方法.

1) 给定 $\mathbf{u}^h(0)$ 逼近 $\mathbf{u}(0)$, 寻找 $\mathbf{u}_h: [0, T] \rightarrow X_{h,r}$ 和 $\mathbf{p}_h: [0, T] \rightarrow Q_{h,r-1}$, 使得 $(\mathbf{u}_h, \mathbf{p}_h)$ 满足 (5) – (6) 式;

2) 对离散的速度和压力 $(\mathbf{u}_h(T), \mathbf{p}_h(T))$ 后处理, 寻找 $(\tilde{\mathbf{u}}_h, \tilde{\mathbf{p}}_h) \in (\tilde{\mathcal{X}}, \tilde{\mathcal{Q}})$ 使得:

$$\begin{aligned} \nu(\nabla \tilde{\mathbf{u}}_h, \nabla \tilde{\boldsymbol{\phi}}) + b_h(\mathbf{u}_h(T), \tilde{\mathbf{u}}_h, \tilde{\boldsymbol{\phi}}) + (\nabla \tilde{\mathbf{p}}_h, \tilde{\boldsymbol{\phi}}) + G(\tilde{\mathbf{u}}_h, \tilde{\boldsymbol{\phi}}) = (f, \tilde{\boldsymbol{\phi}}) - (\dot{\mathbf{u}}_h(T), \tilde{\boldsymbol{\phi}}) \quad \forall \tilde{\boldsymbol{\phi}} \in \tilde{\mathcal{X}} \\ (\nabla \cdot \tilde{\mathbf{u}}_h, \tilde{\boldsymbol{\psi}}) = 0 \quad \forall \tilde{\boldsymbol{\psi}} \in \tilde{\mathcal{Q}} \end{aligned}$$

2 数值实验

本节将利用 FreeFem++ 软件^[10] 进行数值实验来证明算法的有效性和比较 3 种亚格子稳定化后处理混合有限元方法的不同. 在实验中, Hood-Taylor 元用于空间离散, 算法中第一步的非线性系统由 Crank-Nicolson Admas-Bashforth 方法求解, 第二步在相同的网格上用高阶 Hood-Taylor 元 $P_3 - P_2$ 元后处理.

2.1 解析解

特别地, 对于算法 1, 我们选择 Navier-Stokes 方程的准确解为:

$$\begin{aligned} \mathbf{u}_1 &= 10x^2(x-1)^2y(y-1)(2y-1)\cos(t), (x, y, t) \in \Omega \times [0, T] \\ \mathbf{u}_2 &= -10x(x-1)(2x-1)y^2(y-1)^2\cos(t), (x, y, t) \in \Omega \times [0, T] \\ \mathbf{p} &= 10(2x-1)(2y-1)\cos(t), (x, y, t) \in \Omega \times [0, T] \end{aligned}$$

其中, 求解区域 $\Omega = [0, 1] \times [0, 1] \subset \mathbb{R}^2$, 并且取 $\nu = 1$, $T = 0.1$, $\alpha = 0.1 h^2 |\log(h)|$. 表 1 给出了 T 时刻稳定化的混合有限元逼近解和后处理所得解的第一个分量的 L^2 -范误差, 其中 $\|\mathbf{u}(T) - \mathbf{u}_N^h\|_0$ 表示稳定化的混合有限元速度解的误差, $\|\mathbf{u}(T) - \tilde{\mathbf{u}}_N^h\|_0$ 表示稳定化后处理速度解的误差, 第二个分量类似可得; 表 2 为 H^1 -范误差; 表 3 为压力解的误差比较, 其中 $\|\mathbf{P}(T) - \mathbf{P}_h\|_0$ 为稳定化的混合有限元解的压力误差, $\|\mathbf{P}(T) - \tilde{\mathbf{P}}_h\|_0$ 表示稳定化后处理压力误差. 通过表 1–3 可知, 稳定化的后处理混合有限元方法确实提高了稳定化的混合有限元解的精确度.

表 1 稳定化的混合有限元解和后处理的 L^2 -范误差

| h | Δt | $\ \mathbf{u}(T) - \mathbf{u}_N^h\ _0$ | 收敛率 1 | $\ \mathbf{u}(T) - \tilde{\mathbf{u}}_N^h\ _0$ | 收敛率 2 |
|-----|------------|--|----------|--|----------|
| 1/3 | 1/1 000 | 0.002 623 32 | — | 0.000 546 192 | — |
| 1/4 | 1/2 000 | 0.001 144 46 | 2.883 42 | 0.000 204 008 | 3.423 26 |
| 1/5 | 1/4 000 | 0.000 582 591 | 3.025 86 | 8.988 45e-005 | 3.673 12 |
| 1/6 | 1/8 000 | 0.000 332 497 | 3.076 18 | 4.551 18e-005 | 3.732 71 |

表 2 稳定化的混合有限元解和后处理解的 H^1 -范误差

| h | Δt | $\ \mathbf{u}(T) - \mathbf{u}_N^h \ _1$ | 收敛率 1 | $\ \mathbf{u}(T) - \tilde{\mathbf{u}}_N^h \ _1$ | 收敛率 2 |
|-----|------------|--|----------|--|----------|
| 1/3 | 1/1 000 | 0.056 121 6 | — | 0.011 215 6 | — |
| 1/4 | 1/2 000 | 0.033 580 3 | 1.785 24 | 0.005 077 48 | 2.754 74 |
| 1/5 | 1/4 000 | 0.022 172 3 | 1.860 21 | 0.002 728 83 | 2.782 69 |
| 1/6 | 1/8 000 | 0.015 690 6 | 1.896 55 | 0.001 665 23 | 2.709 01 |

表 3 稳定化的混合有限元压力解和后处理压力解的误差

| h | Δt | $\ \mathbf{P}(T) - \mathbf{P}_h \ _0$ | 收敛率 1 | $\ \mathbf{P}(T) - \tilde{\mathbf{P}}_h \ _0$ | 收敛率 2 |
|-----|------------|--|----------|--|----------|
| 1/3 | 1/1 000 | 0.293 433 | — | 0.012 140 7 | — |
| 1/4 | 1/2 000 | 0.168 947 | 1.918 99 | 0.004 673 56 | 3.318 38 |
| 1/5 | 1/4 000 | 0.113 955 | 1.764 72 | 0.002 215 05 | 3.346 04 |
| 1/6 | 1/8 000 | 0.087 428 8 | 1.453 35 | 0.001 225 75 | 3.245 49 |

为了对比 3 种算法, 取网格尺寸 $h=1/160$, 时间步长 $\Delta t=1/8 000$, 分别取 $\nu=1, 0.1, 0.01$ 和 0.001 , 进行数值求解. 数值结果在表 4—6 中给出, 通过比较发现: 从花费时间的长短来看, 除 $\nu=1$ 以外, 在其它情况下稳定化的 Newton 型后处理花费的时间相对较多, 而稳定化的 Ossen 型后处理花费的时间相对较少. 从精确度来看, 用 3 种算法所得速度的 L^2 -范误差基本一致, 但由算法 2 和算法 3 所得的速度的 H^1 -范误差和压力的 L^2 -范误差比算法 1 更准确.

表 4 3 种算法所得逼近解的 L^2 -范误差

| ν | 算法 1 | | 算法 2 | | 算法 3 | |
|-------|--|------------|--|------------|--|------------|
| | $\ \mathbf{u}(T) - \tilde{\mathbf{u}}_N^h \ _0$ | 计算时间 /s | $\ \mathbf{u}(T) - \tilde{\mathbf{u}}_N^h \ _0$ | 计算时间 /s | $\ \mathbf{u}(T) - \tilde{\mathbf{u}}_N^h \ _0$ | 计算时间 /s |
| 1 | 4.170 23e-008 | 646.368 | 3.281 82e-008 | 645.385 | 3.281 8e-008 | 642.455 |
| 0.1 | 4.122 22e-007 | 6 446.090 | 3.271 9e-007 | 6 475.770 | 3.271 86e-007 | 6 438.950 |
| 0.01 | 4.042 53e-006 | 12 240.900 | 3.263 82e-006 | 12 300.300 | 3.264 11e-006 | 12 229.000 |
| 0.001 | 3.439 92e-005 | 18 037.300 | 3.160 34e-005 | 18 158.800 | 3.181 14e-005 | 18 019.600 |

表 5 3 种算法所得逼近解的 H^1 -范误差

| ν | 算法 1 | | 算法 2 | | 算法 3 | |
|-------|--|------------|--|------------|--|------------|
| | $\ \mathbf{u}(T) - \tilde{\mathbf{u}}_N^h \ _1$ | 计算时间 /s | $\ \mathbf{u}(T) - \tilde{\mathbf{u}}_N^h \ _1$ | 计算时间 /s | $\ \mathbf{u}(T) - \tilde{\mathbf{u}}_N^h \ _1$ | 计算时间 /s |
| 1 | 3.181 36e-005 | 646.368 | 3.191 44e-006 | 645.385 | 3.191 44e-006 | 642.455 |
| 0.1 | 0.000 317 569 | 6 446.090 | 3.190 04e-005 | 6 475.770 | 3.190 04e-005 | 6 438.950 |
| 0.01 | 0.003 120 13 | 12 240.900 | 0.000 318 435 | 12 300.300 | 0.000 318 434 | 12 229.000 |
| 0.001 | 0.026 555 6 | 18 037.300 | 0.003 128 28 | 18 158.800 | 0.003 128 08 | 18 019.600 |

表 6 3 种算法所得逼近解的压力误差

| ν | 算法 1 | | 算法 2 | | 算法 3 | |
|-------|--|------------|--|------------|--|------------|
| | $\ \mathbf{P}(T) - \tilde{\mathbf{P}}_h \ _0$ | 计算时间 /s | $\ \mathbf{P}(T) - \tilde{\mathbf{P}}_h \ _0$ | 计算时间 /s | $\ \mathbf{P}(T) - \tilde{\mathbf{P}}_h \ _0$ | 计算时间 /s |
| 1 | 2.533 81e-005 | 646.368 | 2.591 14e-006 | 645.385 | 2.591 99e-006 | 642.455 |
| 0.1 | 2.533 85e-005 | 6 446.090 | 2.592 76e-006 | 6 475.770 | 2.588 36e-006 | 6 438.950 |
| 0.01 | 2.533 86e-005 | 12 240.900 | 2.643 26e-006 | 12 300.300 | 2.609 82e-006 | 12 229.000 |
| 0.001 | 2.533 92e-005 | 18 037.300 | 3.884 7e-006 | 18 158.800 | 3.060 6e-006 | 18 019.600 |

2.2 圆柱绕流

在这个实验中, 我们考虑一个定义在 $\Omega = [0, 6] \times [0, 1]$ 上的圆柱绕流问题, 其中该圆柱的圆心 $(x, y) = (1, 0.5)$, 半径为 0.15. 取入流速度为

$$u_1(x, y, t) = -6y(1 - y)$$

$$u_2(x, y, t) = 0$$

并且在其它边界上满足无滑动边界条件^[11]. 在这个问题中, Hood-Taylor 元用于空间离散, 粘性系数 $\nu = 0.001$. 设置网格尺寸为 $h = 1/12$, 稳定化参数 $\alpha = 0.1h^2 |\log(h)|$, 时间步长 $\Delta t = 0.001$, 分别取最后时刻 $T = 1, 2, 4, 6, 8, 10$. 图 1 描述了 3 种算法关于动能和时间 T 的关系. 从图中可以看出在 $T = 1, 2, 4, 6$ 时 3 种算法所得解的动能差距不大, 但在 $T = 8, 10$ 时算法 2 和算法 3 所得解的动能比算法 1 的大.

图 2 描绘了在 $T = 1, 2, 4, 6, 8, 10$ 的情况下, 算法 2 在圆柱周围的速度分量 \tilde{u}_{h_1} 的等值线; 图 3 是速度分量 \tilde{u}_{h_2} 的等值线; 而图 4 是压力等值线. 对于算法 1 和算法 3, 可得到类似的结果.

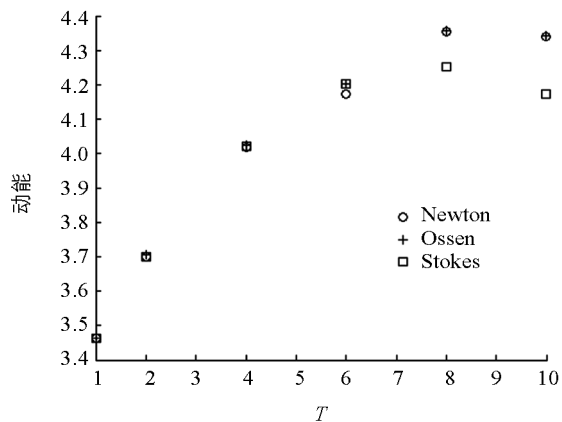


图 1 动能与时间关系图

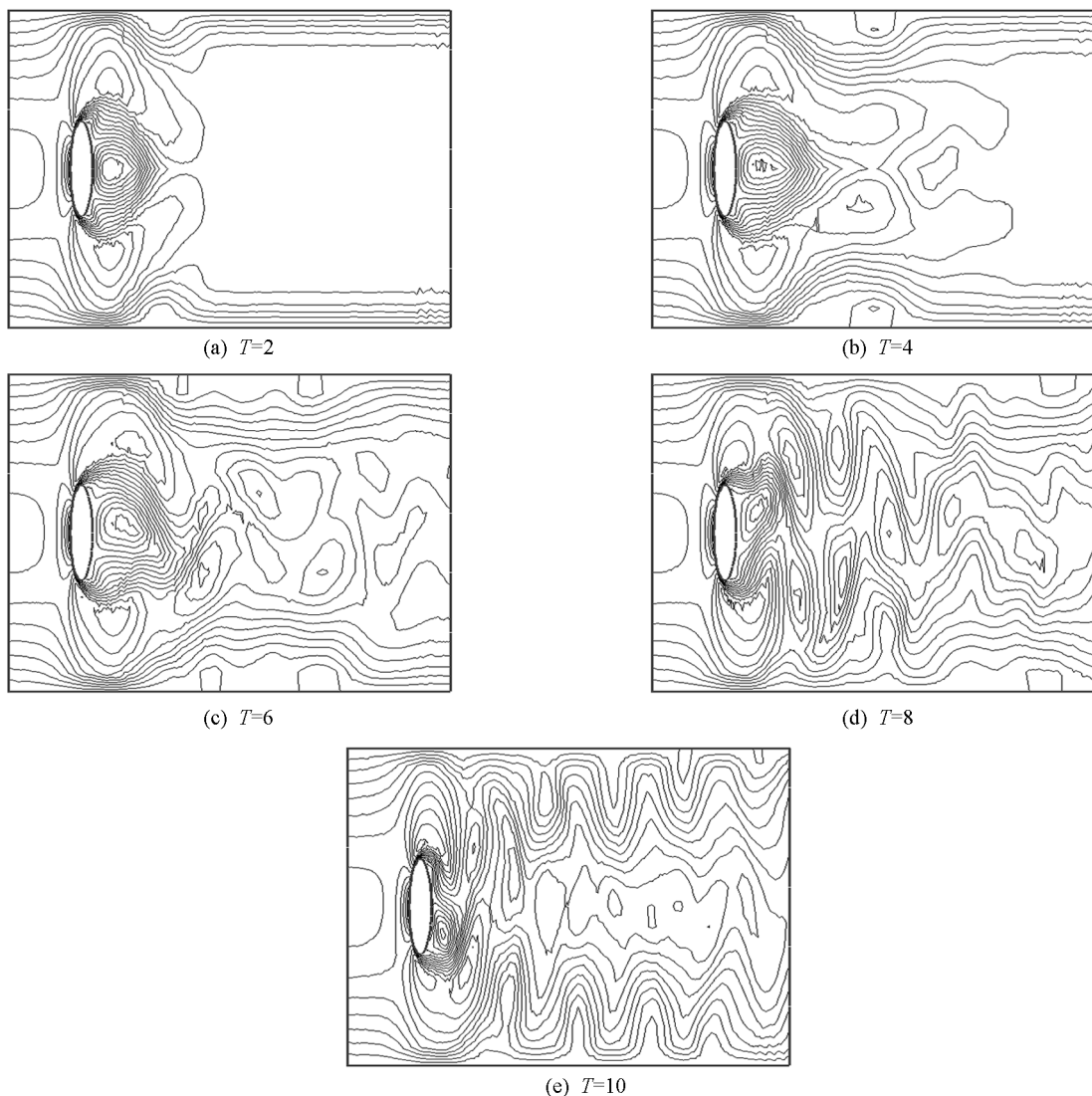


图 2 算法 2 在不同时刻的速度分量 \tilde{u}_{h_1} 等值线

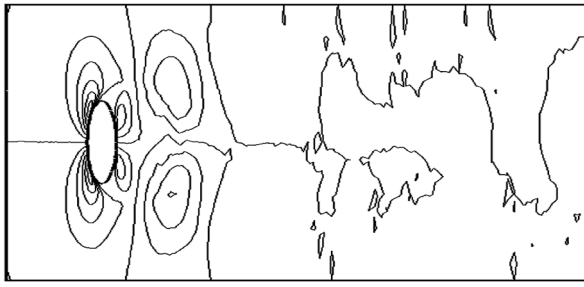
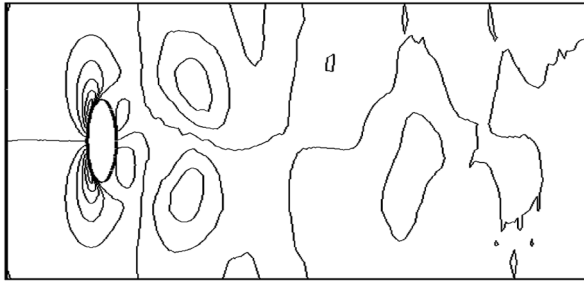
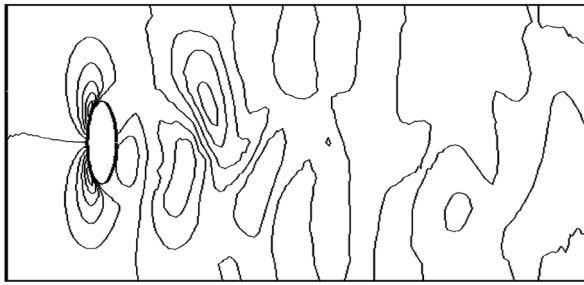
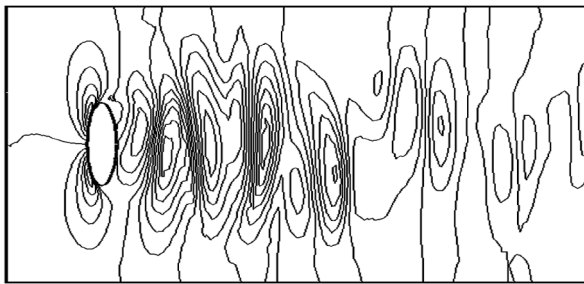
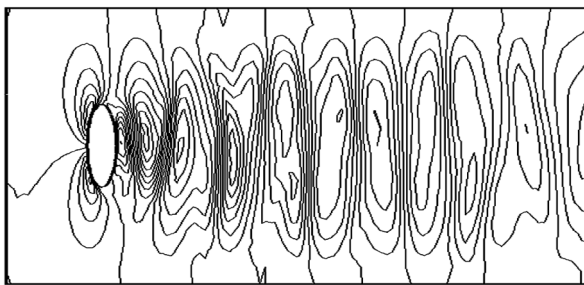
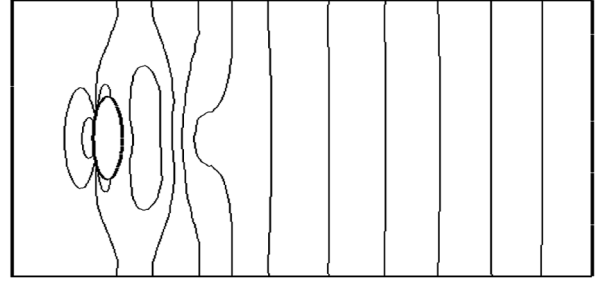
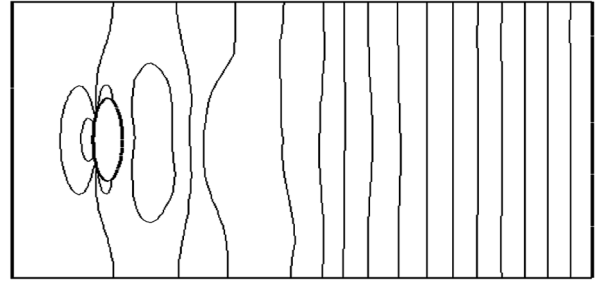
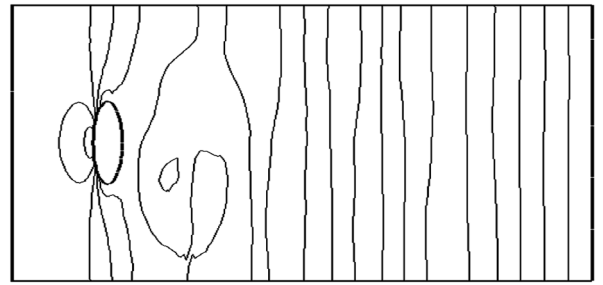
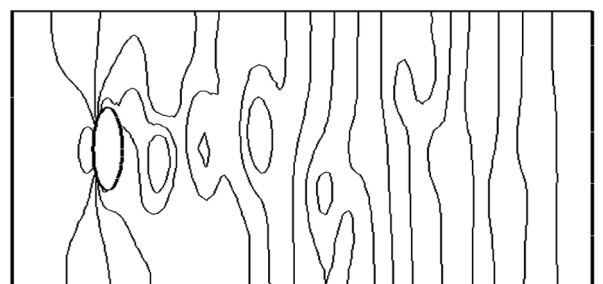
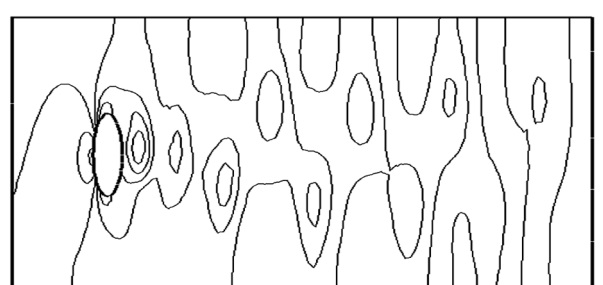
(a) $T=2$ (b) $T=4$ (c) $T=6$ (d) $T=8$ (e) $T=10$ 图 3 算法 2 在不同时刻的速度分量 \tilde{u}_{h2} 等值线(a) $T=2$ (b) $T=4$ (c) $T=6$ (d) $T=8$ (e) $T=10$

图 4 算法 2 在不同时刻的压力等值线

3 结 论

在这篇文章中, 我们主要通过数值实验研究了 3 种基于亚格子模型的后处理混合有限元方法, 实验结果表明: 在选取适当的稳定化参数和网格尺寸的条件下, 3 种稳定化的后处理有限元方法提高了稳定化的混合有限元解的精确度, 并且收敛阶也明显提高了一阶. 但是从计算时间看, 除 $\nu = 1$ 以外, 在其它情况下稳定化的 Newton 型后处理花费的时间相对较多, 而稳定化的 Ossen 型后处理花费的时间相对较少. 从精确度来看, 用 3 种算法所得的速度的 L^2 -范误差基本一致, 但 Newton 型后处理和 Ossen 型后处理方法所得的速度的 H^1 -范误差和压力的 L^2 -范误差比 Stokes 型后处理方法更有效.

参考文献:

- [1] TEMAN R. Navier-Stokes Equations: Theory and Numerical Analysis [M]. Amsterdam: North-Holland Publishing, 1977.
- [2] GIRAULT V, RAVIART P A. Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms [M]. Berlin: Springer-Verlag, 1986.
- [3] GLOWINSKI R. Finite Element Methods for Incompressible Viscous Flow [M] // Handbook of Numerical Analysis. Amsterdam: Elsevier, 2003.
- [4] ARCHILLA B G, NOVO J, TITI E S. Postprocessing the Galerkin Method: A Novel Approach to Approximate Inertial Manifolds [J]. SIAM Journal on Numerical Analysis, 1998, 35(3): 941-972.
- [5] GARC B. Postprocessing the Galerkin Method: The Finite-Element Case [J]. SIAM Journal on Numerical Analysis, 2000, 37(2): 470-499.
- [6] AYUSO B, ARCHILLA B G, NOVO J. The Postprocessed Mixed Finite-Element Method for the Navier-Stokes Equations [J]. SIAM Journal on Numerical Analysis, 2005, 43(3): 1091-1111.
- [7] FRUTOS J D, ARCHILLA B G, NOVO J. Static Two-Grid Mixed Finite-Element Approximations to the Navier-Stokes Equations [J]. Journal of Scientific Computing, 2012, 52(3): 619-637.
- [8] DURANGO F, NOVO J. Two-Grid Mixed Finite-Element Approximations to the Navier-Stokes Equations Based on a Newton-Type Step [J]. Journal of Scientific Computing, 2018, 74(1): 456-473.
- [9] SHANG Y Q. A Two-Level Subgrid Stabilized Oseen Iterative Method for the Steady Navier-Stokes Equations [J]. Journal of Computational Physics, 2013, 233(1): 210-226.
- [10] HECHT F. New Development in Freefem++ [J]. Journal of Numerical Mathematics, 2012, 20(3-4): 159-344.
- [11] SHANG Y Q. Error Analysis of a Fully Discrete Finite Element Variational Multiscale Method for Time-Dependent Incompressible Navier-Stokes Equations [J]. Numerical Methods for Partial Differential Equations, 2013, 29(6): 2025-2046.

A Subgrid Stabilizing Postprocessed Mixed Finite Element Method for the Navier-Stokes Equations

ZHANG Qi-hui, SHANG Yue-qiang

School of Mathematics and Statistics, Southwest University, Chongqing 400715, China

Abstract: In this paper, we mainly study three postprocessed mixed finite element methods for the incompressible Navier-Stokes equations, which are based on a subgrid model. These methods consist of two steps. The first step is to solve a subgrid stabilized nonlinear Navier-Stokes problem on a coarse grid to obtain an approximate solution \mathbf{u}_H at time T . The second step is to postprocess \mathbf{u}_H on a finer grid (or by high-order finite elements), by solving a stabilized Stokes problem, a stabilized Newton-Type problem, or a stabilized Oseen problem. The numerical results show that under the conditions of selecting appropriate stabilizing parameters and grid sizes, the postprocessed finite element method can improve the precision of the mixed finite-element solution, and the order of convergence is obviously improved by one unit compared with the standard subgrid stabilized method. From the point of the computational time, in addition to $\nu=1$, the stabilized Newton-type postprocessed method takes a relatively more time than the others, while the stabilized Oseen-type postprocessed method takes the least time among the three methods. And from the point of precision of the computed solutions, the Newton and Oseen-type postprocessed methods are better than the Stokes-type postprocessed method.

Key words: Navier-Stokes equations; postprocessing; subgrid stabilization; finite element method

责任编辑 张 枸

