物理学报 Acta Physica Sinica

Chinese Physical Society

Institute of Physics, CAS

不可压Navier-Stokes方程的投影方法

张庆海　李阳

Projection methods for the incompressible Navier–Stokes equations

Zhang Qing-Hai　Li Yang

引用信息 Citation: Acta Physica Sinica, 70, 130201 (2021)　DOI: 10.7498/aps.70.20210259

在线阅读 View online: https://doi.org/10.7498/aps.70.20210259

当期内容 View table of contents: http://wulixb.iphy.ac.cn

您可能感兴趣的其他文章

Articles you may be interested in

基于混合雷诺平均/高精度隐式大涡模拟方法的高升力体气动噪声模拟

Aeroacoustic simulation of the high–lift airfoil using hybrid reynolds averaged Navier–Stokes/high–order implicit large eddy simulation method

物理学报. 2019, 68(20): 204702　https://doi.org/10.7498/aps.68.20190777

三维不可压缩流的12速多松弛格子Boltzmann模型

Three–dimensional 12–velocity multiple–relaxation–time lattice Boltzmann model of incompressible flows

物理学报. 2019, 68(23): 234701　https://doi.org/10.7498/aps.68.20190984

一种径向基函数虚拟网格法数值模拟复杂边界流动

Numerical simulation of complex immersed boundary flow by a radial basis function ghost cell method

物理学报. 2017, 66(4): 044704　https://doi.org/10.7498/aps.66.044704

四阶色散非线性薛定谔方程的明暗孤立波和怪波的形成机制

Study on the generation mechanism of bright and dark solitary waves and rogue wave for a fourth–order dispersive nonlinear Schrdinger equation

物理学报. 2020, 69(1): 010502　https://doi.org/10.7498/aps.69.20191384

基于浸入边界–多松弛时间格子玻尔兹曼通量求解法的流固耦合算法研究

A method combined immersed boundary with multi–relaxation–time lattice Boltzmann flux solver for fluid–structure interaction

物理学报. 2017, 66(22): 224702　https://doi.org/10.7498/aps.66.224702

可变周期谐波平衡法求解周期性非定常涡脱落问题

Application of variable–time–period harmonic balance method to periodic unsteady vortex shedding

物理学报. 2019, 68(12): 124701　https://doi.org/10.7498/aps.68.20190126

青年科学评述

不可压 Navier-Stokes 方程的投影方法*

张庆海† 李阳

(浙江大学数学科学学院, 杭州 310027)

(2021 年 2 月 3 日收到; 2021 年 3 月 9 日收到修改稿)

不可压 Navier-Stokes 方程是流体力学的基本控制方程, 其高精度数值模拟具有重要的科学意义. 本综述性文章回顾了求解 Navier-Stokes 方程的投影方法, 重点介绍了时空一致四阶精度的 GePUP 方法. 该方法用一个广义投影算子对不可压 Navier-Stokes 方程进行了重新表述, 使得投影流速的散度由一个热方程控制, 保持了 UPPE 方法的优点. 与 UPPE 方法不同的是, GePUP 方法的推导不依赖于 Leray-Helmholtz 投影算子的各种性质, 并且 GePUP 表述中的演化变量无需满足散度为零的条件, 因此数值近似 Leray-Helmholtz 投影算子的误差对精度和稳定性的影响非常透明. 在 GePUP 方法中, 时间积分和空间离散是完全解耦的, 因此对这两个模块都能以“黑匣子”的方式自由替换. 时间积分模块的灵活性实现了时间上的高阶精度, 并使得 GePUP 算法能同时适用于低雷诺数流体和高雷诺数流体. 空间离散模块的灵活性使得 GePUP 算法能很好地适应不规则边界. 理论分析和数值测试结果都显示, 相对于二阶投影方法, GePUP 方法无论在精度上还是效率上都具有巨大优势.

关键词: 不可压Navier-Stokes方程, 投影方法, 时空一致四阶精度, 无滑移边界条件, 广义投影算子, 压力 Poisson 方程

PACS: 02.60.Cb, 47.10.ad, 47.11.Df **DOI:** 10.7498/aps.70.20210259

1 概 述

1900 年, 著名数学家希尔伯特在国际数学联盟 IMU (International Mathematical Union) 巴黎会议上提出了 23 个数学问题, 这个列表极大地影响了 20 世纪的数学发展. 1998 年, 菲尔兹奖得主 Smale[1] 应 IMU 的邀请, 给出了 18 个他认为在 21 世纪非常重要的数学问题. 2000 年 5 月, 克雷数学研究所发布了七个千禧年数学问题[2], 由于这些问题的重要性, 解决其中任何一个都能获得 100 万美元的奖金. 在千禧年数学问题和 Smale 的 18 个问题中, 有一个问题同时存在, 它就是三维空间中 Navier-Stokes 方程全局解的存在性和正则性.

数学家们对 Navier-Stokes 方程的重视源于该方程极其广泛的应用, 尤其是该方程的解对于理解湍流很可能具有根本性的重要意义. 如果一个牛顿流体是性质均匀且各向同性的, 其不可压 Navier-Stokes 方程 (INSE) 的基本形式为

$$\frac{\partial \boldsymbol{u}}{\partial t} = -\boldsymbol{u}\cdot\nabla\boldsymbol{u}+\boldsymbol{g}-\nabla p+\nu\Delta\boldsymbol{u}, \ \text{in}\ \Omega\times(0,+\infty), \tag{1a}$$

$$\nabla\cdot\boldsymbol{u}=0, \quad \text{in}\ \Omega\times(0,+\infty), \tag{1b}$$

其中 t 是时间变量, $\boldsymbol{x}\in\mathbb{R}^{\mathrm{D}}$ 是空间变量 $(\mathrm{D}=2,3)$, $\boldsymbol{g}$ 是外力项, ν 是动力黏性系数, $p(\boldsymbol{x},t)$ 是压强, 而 $\boldsymbol{u}(\boldsymbol{x},t)$ 是流速. 假定流速场的初始条件 $\boldsymbol{u}_0\in\mathcal{C}^\infty$ 光滑无源, 并且其空间导数满足一定的有界条件, 对于外力项的空间时间偏导数也做类似假定. 在千禧年问题中, 空间域 Ω 是 $\mathbb{R}^3$ 或 3-环面 $(\mathbb{R}^3/\mathbb{Z}^3)$, 因此 Ω 没有边界. 但是在实际物理问题中, 往往需要指定 (1) 式中的边界条件. 除周期边界条件外, 本文主要讨论最普遍也是最重要的无滑移边界条件:

$$\boldsymbol{u}=\boldsymbol{0}, \quad \text{on}\ \partial\Omega. \tag{2}$$

* 国家自然科学基金创新研究群体项目 (批准号: 11621101) 和国家自然科学基金 (批准号: 11871429) 资助的课题.

† 通信作者. E-mail: qinghai@zju.edu.cn

© 2021 中国物理学会 Chinese Physical Society

张庆海, 清华大学学士及硕士, 美国康奈尔大学 (Cornell University) 博士. 曾在美国劳伦斯伯克利国家实验室 (Lawrence Berkeley National Lab) 从事博士后研究并曾在美国犹他大学 (University of Utah) 数学系担任科研助理教授. 2015 年获得国家级青年人才项目支持, 2016 年初任浙江大学数学学院教授. 主要研究方向为动边界不可压流体的理论建模和数值计算, 工作聚焦点为界面追踪和高阶有限体积方法. 研究的核心理念是在界面问题中充分运用几何和拓扑的手段来解决几何和拓扑的问题. 代表性工作有: 建立了一个界面追踪及其数值方法的理论分析框架, 提出了四阶及以上精度的界面追踪和曲率估计算法, 并发展了一系列时空一致四阶精度的并行自适应有限体积法. 在 *SIAM Review*, *Mathematics of Computation*, *SIAM Journal on Numerical Analysis*, *SIAM Journal on Scientific Computing*, *PNAS*, *Journal of Computational Physics*, *Computer Methods in Applied Mechanics and Engineering* 等学术刊物上发表论文 30 多篇.

在无滑移边界条件下, 中高雷诺数不可压流体往往含有多个时间尺度和多个空间尺度的流动结构. 因此在进行数值模拟时, 数值方法也必须拥有足够的精度来捕捉所有相关尺度的特征, 这对计算效率提出了极大的挑战. 一阶和二阶方法的计算结果虽然可以收敛, 但是为了达到给定的计算精度, 有限的计算资源可能会很快地消耗殆尽. 另外, 至关重要的湍流特征往往取决于流速场的梯度或高阶导数. 例如, 涡量是流速的一阶导数, 而转捩点是切向流速的二阶法向导数为零的位置. 低阶方法的流速场计算结果虽然可以收敛, 但是对于流速导数决定的重要物理量往往存在较大的误差. 综上所述, 我们希望发展一个时空一致高阶方法, 达到以下五个目标:

(Goal-1) 在时间上和空间上同时达到四次收敛阶;

(Goal-2) 在每个时间步内只需求解常数个关于压强或关于流速的线性方程组, 并且方程组求解的复杂度为 $O(N)$, 这里 N 是控制体个数;

(Goal-3) 时间积分方法的具体细节不出现在整体算法中;

(Goal-4) 时间积分与空间离散完全解耦, 也就是说用户可以在一定范围内自由选择空间离散和时间积分模块组合成一个整体;

(Goal-5) 同时适用于高雷诺数和低雷诺数流体.

众所周知, 当动力黏性系数很大时, INSE 的半离散系统具有刚性, 需要对 (1a) 式中的扩散项采用隐式时间积分, 否则时间步长的选取将受到数值稳定性非常苛刻的限制. 而当动力黏性系数很小时, 对流占优使得 INSE 的半离散系统不具有刚性, 此时用显式时间积分方法也可以使时间步长免受稳定性的限制. 如果不能满足 (Goal-3), 那么针对不同情况更换时间积分方法将是一件非常繁复的任务. 反之, 如果能以“黑匣子”的方式调用高阶时间积分模块, 将会实现 (Goal-4), 同时能为整体算法提供最大的灵活性, 从而轻松实现 (Goal-5). 因此, (Goal-5) 可以看成是 (Goal-3) 和 (Goal-4) 的副产品.

在第 2 节预备知识和第 3 节经典投影方法的基础上, 本文重点介绍了一个满足上述所有目标 ((Goal-1) —(Goal-5)) 的四阶投影方法 GePUP[3–5], 该方法达到核心目标 ((Goal-3) 和 (Goal-4)) 的关键是策略性地利用投影算子处理不可压约束条件 (1b) 式. 第 4 节推导 INSE 的另一种表述, 作为 GePUP 方法的控制方程, 并简要总结了在有限体积框架下的数值算法. 由于 (Goal-4) 带来的灵活性, 用针对不规则区域的离散格式替换经典有限体积离散格式就得到了一个时空一致四阶精度, 适用于不规则区域的投影方法. 第 5 节中的数值实验结果验证了 GePUP 算法的时空一致四阶精度及其捕捉真实物理过程的能力, 其中 5.4 节系统地论述了 GePUP 算法相对二阶方法 [6,7] 在精度和效率上的优越性.

2 基础知识

2.1 Helmholtz 分解定理

与流速场不同, 动量方程 (1a) 中的压强 p 既不需要初始条件也不需要边界条件. 这是因为 Helmholtz 分解定理指出, 有界区域 $\Omega \subseteq \mathbb{R}^D$ 上任何充分光滑的向量场 $\boldsymbol{v}^*$ 都可以被唯一地分解为一个无源场 $\boldsymbol{v}$ 和一个无旋场 $\nabla\phi$ 之和:

$$\boldsymbol{v}^* = \boldsymbol{v} + \nabla\phi, \tag{3a}$$

$$\nabla\cdot\boldsymbol{v} = 0, \qquad \nabla\times\nabla\phi = \boldsymbol{0}. \tag{3b}$$

该分解可以通过求解 Neumann 边界条件下的 Poisson 方程实现:

$$\Delta\phi = \nabla\cdot\boldsymbol{v}^* \qquad \text{in } \Omega, \tag{4a}$$

$$\boldsymbol{n}\cdot\nabla\phi=\boldsymbol{n}\cdot(\boldsymbol{v}^*-\boldsymbol{v})\qquad \text{on } \partial\Omega, \tag{4b}$$

其中$\boldsymbol{n}$表示区域边界$\partial\Omega$的单位外法向量.

如果定义欧拉加速度

$$\boldsymbol{a}:=\frac{\partial\boldsymbol{u}}{\partial t},\qquad \boldsymbol{a}^*:=-\boldsymbol{u}\cdot\nabla\boldsymbol{u}+\boldsymbol{g}+\nu\Delta\boldsymbol{u}, \tag{5}$$

则可将动量方程 (1a) 写成 (3a) 式的形式:

$$\boldsymbol{a}^*=\boldsymbol{a}+\nabla p. \tag{6}$$

因此压强可由欧拉加速度$\boldsymbol{a}^*$在相差一个常数的意义下由 Helmholtz 定理唯一确定, 并且在这个确定过程中无需任何 p 的边界条件, 或者说 p 的边界条件也是由欧拉加速度$\boldsymbol{a}^*$和$\boldsymbol{a}$在边界上的法向分量确定的. 另外一个从流速场$\boldsymbol{u}$得到压强的方法是对动量方程 (1a) 求散度, 应用 (1b) 式后可以得到关于压强的 Poisson 方程, 从而求解出压强及其梯度.

2.2 Leray-Helmholtz 投影算子

下面的定理在很多偏微分方程的教科书上都可以找到, 如文献 [8]:

定理 1 如果 f 和 g 足够光滑, 具有 Neumann 边界条件的 Poisson 方程

$$\Delta\phi=f\ \text{ in }\ \Omega,\qquad \boldsymbol{n}\cdot\nabla\phi=g\ \text{ on }\ \partial\Omega \tag{7}$$

解存在的充要条件是

$$\int_\Omega f=\oint_{\partial\Omega} g. \tag{8}$$

并且ϕ在相差一个常数的意义下是唯一确定的.

容易验证 (4) 式满足可解条件 (8) 式且$\nabla\phi$唯一确定. 因此对于任意给定的向量场$\boldsymbol{v}^*$都可以唯一确定 (3a) 式中的无源场$\boldsymbol{v}$. 由这个唯一性可以定义一个线性算子$\mathscr{P}$, 称为 Leray-Helmholtz 投影算子, 从而把 Helmholtz 定理中的分解步骤整合其中,

$$\mathscr{P}\boldsymbol{v}^*:=\boldsymbol{v}=\boldsymbol{v}^*-\nabla\phi. \tag{9}$$

由 (3b) 式可知, 在任意时刻 $\mathscr{P}$ 都满足

$$\mathscr{P}^2=\mathscr{P},\quad \nabla\cdot\mathscr{P}\boldsymbol{v}^*=0,\quad \mathscr{P}\nabla\phi=\boldsymbol{0}. \tag{10}$$

将 Leray-Helmholtz 投影算子作用在欧拉加速度$\boldsymbol{a}^*$上可以得到$\boldsymbol{a}$, 从而得到压强梯度$\nabla p=(\mathcal{I}-\mathscr{P})\boldsymbol{a}^*$, 这个压强梯度的提取方式与时间无关.

在周期边界条件下, Leray-Helmholtz 投影算子$\mathscr{P}$和 Laplace 算子Δ是可交换的, 将$\mathscr{P}$作用于 (1a) 式, 再结合 (10) 式中最后一个性质可得

$$\frac{\partial\boldsymbol{u}}{\partial t}=\mathscr{P}(-\boldsymbol{u}\cdot\nabla\boldsymbol{u}+\boldsymbol{g})+\nu\Delta\boldsymbol{u}. \tag{11}$$

在应用 Method-of-lines (MOL) 方法数值求解 INSE 时, 由于 (1b) 式不是一个动力过程而是一个限制条件, 直接对 (1) 式进行离散得到的常微分方程组很难在求解过程中保证收敛阶[9]. 相比来说, (11) 式是关于流速场的一个动力过程, 可以直接应用 MOL 方法. 这是 (11) 式相对于 (1) 式的一个显著优势.

2.3 Laplace-Leray 交换子

在无滑移条件下, Leray-Helmholtz 投影算子$\mathscr{P}$和 Laplace 算子Δ是不可交换的. 为了度量这两者的不可交换性并推导出不带约束条件的关于流速场的动力过程, 定义 Laplace-Leray 交换子为

$$[\Delta,\mathscr{P}]:=\Delta\mathscr{P}-\mathscr{P}\Delta. \tag{12}$$

为了推导其表达式, 先定义梯度-散度交换子

$$\mathcal{B}=[\nabla,\nabla\cdot]:=\Delta-\nabla\nabla\cdot, \tag{13}$$

满足

$$\nabla\cdot\mathcal{B}=0, \tag{14a}$$

$$\Delta\mathscr{P}=\mathcal{B}, \tag{14b}$$

其中 (14a) 式是由 (13) 式以及Δ和$\nabla\cdot$的交换性得到的, 而 (14b) 式是由 (9) 式, Δ和∇的交换性, (4a) 式以及 (13) 式得到的:

$$\begin{aligned}\Delta\mathscr{P}\boldsymbol{v}^*&=\Delta(\boldsymbol{v}^*-\nabla\phi)=\Delta\boldsymbol{v}^*-\nabla\Delta\phi\\&=\Delta\boldsymbol{v}^*-\nabla\nabla\cdot\boldsymbol{v}^*=\mathcal{B}\boldsymbol{v}^*.\end{aligned} \tag{15}$$

应用$\mathscr{P}$于 (15) 式的第一和第三项, 结合 (10) 式中最后一个性质, 可得

$$\mathscr{P}\Delta\mathscr{P}\boldsymbol{v}^*=\mathscr{P}\Delta\boldsymbol{v}^*. \tag{16}$$

因此 Laplace-Leray 交换子满足

$$[\Delta,\mathscr{P}]:=\Delta\mathscr{P}-\mathscr{P}\Delta=(\mathcal{I}-\mathscr{P})\Delta\mathscr{P}=(\mathcal{I}-\mathscr{P})\mathcal{B}, \tag{17}$$

其中$\mathcal{I}$表示恒等算子. 由 (9) 式知$(\mathcal{I}-\mathscr{P})\boldsymbol{v}^*=\nabla\phi$, 因此$[\Delta,\mathscr{P}]\boldsymbol{v}^*$是某标量场的梯度, 若令$\boldsymbol{v}^*$为 (1) 式中的流速$\boldsymbol{u}$, 得到的标量场称为 Stokes 压强, 其梯度满足

$$\nabla p_{\rm s}:=(\Delta\mathscr{P}-\mathscr{P}\Delta)\boldsymbol{u}. \tag{18}$$

由 (17) 式和 (18) 式可得

$$\mathcal{B}\boldsymbol{u}=\mathscr{P}\mathcal{B}\boldsymbol{u}+\nabla p_{\rm s}. \tag{19}$$

结合 (10) 式中第二个性质以及 (14a) 式可得

$\Delta p_{\mathrm{s}}=0$, 即Stokes 压强既无旋也无源, 是调和的.

定义标量p_{c}满足

$$\nabla p_{\mathrm{c}}:=(\mathcal{I}-\mathscr{P})\left(\boldsymbol{g}-\boldsymbol{u}\cdot\nabla\boldsymbol{u}\right). \tag{20}$$

应用 Leray-Helmholtz 投影算子于 (1a) 式, 再由 (6) 式, (17) 式, (18) 式及 (20) 式可得

$$\nabla p=\nabla p_{\mathrm{c}}+\nu\nabla p_{\mathrm{s}}. \tag{21}$$

因此 INSE 中的压强梯度由两部分构成, 其中∇p_{c}平衡外力项和非线性对流项的散度, 而∇p_{s}度量了 Laplace 算子和 Leray-Helmholtz 投影算子的非交换性. 在$\nu\to 0$和$\nu\to+\infty$这两种极限情况下, 压强梯度相应地由∇p_{c}和$\nu\nabla p_{\mathrm{s}}$所主导, 因而 INSE 的性质在这两种情况下会有很大的差别.

3 一些经典方法

3.1 鞍点法

鞍点法[10]将 (1) 式中的流速和压强耦合起来同时求解. 例如, 由 Crank-Nicolson 方法可得

$$\begin{aligned}&\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^{n}}{\Delta t}+\boldsymbol{G}p^{n+\frac{1}{2}}\\ =&-\boldsymbol{N}\left(\boldsymbol{u}^{n+\frac{1}{2}}\right)+\boldsymbol{g}^{n+\frac{1}{2}}+\nu\boldsymbol{L}\boldsymbol{u}^{n+\frac{1}{2}},\end{aligned} \tag{22a}$$

$$\boldsymbol{D}\boldsymbol{u}^{n+1}=0, \tag{22b}$$

其中, $\boldsymbol{u}^{n+\frac{1}{2}}=\frac{1}{2}\left(\boldsymbol{u}^{n+1}+\boldsymbol{u}^{n}\right)$, $\boldsymbol{N}\left(\boldsymbol{u}^{n+\frac{1}{2}}\right)\approx\left[\left(\boldsymbol{u}^{n+\frac{1}{2}}\cdot\nabla\right)\boldsymbol{u}^{n+\frac{1}{2}}\right]$是对非线性对流项的二阶近似. 将 (22) 式写成一个线性方程组:

$$\boldsymbol{A}\begin{bmatrix}\boldsymbol{u}^{n+1}\\ p^{n+\frac{1}{2}}\end{bmatrix}=\boldsymbol{F}. \tag{23}$$

由于矩阵 $\boldsymbol{A}$ 具有鞍点结构, 这个方法被称为“鞍点法”. 同时求解流速和压强使得线性方程组中的未知量个数最大化, 因此这种方法对方程组的求解效率提出了很大的挑战[11]. 另外, (23) 式中矩阵 $\boldsymbol{A}$ 的推导依赖于空间离散和具体时间积分方法的细节. 对于较为复杂的时间积分方法, 如高阶 Additive Runge-Kutta (ARK) 方法, 矩阵 $\boldsymbol{A}$ 的复杂性往往是难以接受的. 因此, 这个方法难以实现 (Goal-3) 和 (Goal-4), 从而也难以做到 (Goal-1) 和 (Goal-2).

3.2 Chorin 的一阶投影方法

1968 年, Chorin[12] 提出了以下一阶投影方法:

$$\frac{\boldsymbol{u}^{*}-\boldsymbol{u}^{n}}{\Delta t}=-\boldsymbol{C}(\boldsymbol{u}^{*},\boldsymbol{u}^{n})+\boldsymbol{g}^{n}+\nu\boldsymbol{L}\boldsymbol{u}^{*}, \tag{24a}$$

$$\boldsymbol{u}^{n+1}=\boldsymbol{P}\boldsymbol{u}^{*}, \tag{24b}$$

其中 $\boldsymbol{C}(\boldsymbol{u}^{*},\boldsymbol{u}^{n})$是对对流项的近似, $\boldsymbol{P}=\boldsymbol{I}-\boldsymbol{G}\boldsymbol{L}^{-1}\boldsymbol{D}$是对 Leray-Helmholtz 投影算子的一个近似. 该方法首先在忽略压强梯度项的情况下计算一个辅助流速场$\boldsymbol{u}^{*}$, 再将$\boldsymbol{u}^{*}$投影到散度为零的空间上来实现不可压约束 (1b) 式. 由于流速和压强在计算域内解耦, 在每个时间步内, 都只需要求解一系列关于流速或压强的边值问题, 因此线性方程组的规模较鞍点法会小很多, 这种效率优势可能是投影方法得到广泛应用的重要原因之一.

3.3 分步二阶投影方法

在 Chorin 的基础上, 学者们提出了许多二阶投影方法[13,6,14–17], 出发点是先对 INSE 进行时间上的二阶隐式离散:

$$\begin{aligned}&\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^{n}}{\Delta t}+\nabla p^{n+1/2}=\\ &-\left[(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right]^{n+\frac{1}{2}}+\boldsymbol{g}^{n+\frac{1}{2}}+\frac{\nu}{2}\Delta(\boldsymbol{u}^{n+1}+\boldsymbol{u}^{n}),\end{aligned} \tag{25a}$$

$$\nabla\cdot\boldsymbol{u}^{n+1}=0, \tag{25b}$$

这里$[(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}]^{n+1/2}$表示对流项在时刻$t^{n+1/2}$的显式二阶近似. 如果再进行空间离散就得到了鞍点法的 (22) 式. 与 Chorin 的投影方法类似, 可以引入辅助变量将压强和流速解耦. 不同的是, 达到二阶精度需要引入两个辅助变量.

定义 2 分步二阶投影方法由以下三个步骤构成[17]:

(Proj-1) 选择一个边界条件$\boldsymbol{B}(\boldsymbol{u}^{*})=\boldsymbol{0}$以求解辅助变量$\boldsymbol{u}^{*}$:

$$\begin{aligned}\frac{\boldsymbol{u}^{*}-\boldsymbol{u}^{n}}{\Delta t}+\nabla q=&-\left[(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right]^{n+1/2}+\boldsymbol{g}^{n+1/2}\\ &+\frac{\nu}{2}\Delta(\boldsymbol{u}^{*}+\boldsymbol{u}^{n}),\end{aligned} \tag{26}$$

(Proj-2) 利用投影算子实现不可压约束 (25b) 式:

$$\boldsymbol{u}^{*}=\boldsymbol{u}^{n+1}+\Delta t\nabla\phi^{n+1}, \tag{27a}$$

$$\nabla\cdot\boldsymbol{u}^{n+1}=0. \tag{27b}$$

注意ϕ^{n+1}的边界条件要与$\boldsymbol{u}^{n+1}$的物理边界条件以及$\boldsymbol{u}^{*}$的边界条件$\boldsymbol{B}(\boldsymbol{u}^{*})=\boldsymbol{0}$相容.

(Proj-3) 更新压强$p^{n+1/2}$:

$$p^{n+1/2}=q+\boldsymbol{U}(\phi^{n+1}). \tag{28}$$

一个具体的分步二阶投影方法由以上步骤中的三个要素确定, 即压强近似函数 q、辅助变量 $\boldsymbol{u}^*$ 的边界条件 $\boldsymbol{B}(\boldsymbol{u}^*)$ 以及压强更新函数 $\boldsymbol{U}$. 其中投影步骤 (Proj-2) 通过求解 Neumann 边界条件下的 Poisson 方程实现:

$$\Delta t\Delta\phi^{n+1} = \nabla\cdot\boldsymbol{u}^*, \tag{29a}$$

$$\Delta t\frac{\partial\phi^{n+1}}{\partial\boldsymbol{n}} = \boldsymbol{n}\cdot\boldsymbol{u}^*|_{\partial\Omega}. \tag{29b}$$

将 (27a) 式代入 (26) 式, 结合 (25a) 式知 (Proj-3) 步骤中的压强更新公式 (28) 式可写为

$$p^{n+1/2} = q + \phi^{n+1} - \frac{\nu\Delta t}{2}\Delta\phi^{n+1}. \tag{30}$$

以下简要回顾在定义 2 的框架下的两个经典的二阶投影方法.

3.3.1 Bell, Colella & Glaz (BCG)

Bell, Colella 和 Glaz[6] 指定

$$q = p^{n-1/2},\quad \boldsymbol{B}(\boldsymbol{u}^*) = \boldsymbol{u}^*|_{\partial\Omega} = \boldsymbol{0},$$
$$\nabla p^{n+1/2} = \nabla p^{n-1/2} + \nabla\phi^{n+1}. \tag{31}$$

在文献 [17] 中, 作者运用正则模式分析和数值实验验证了在 L_∞ 范数度量下, 该方法对于流速的求解是二阶收敛的, 而压强的计算结果则仅有一阶收敛. 造成压强降阶的原因是 (31) 式中对压强梯度的更新公式与 (30) 式是不匹配的, 忽略 (30) 式中的高阶项会导致压强 p 的精度损失 (通常会出现边界层) 以及压强梯度在边界上的不准确. 的确, (29b) 式和 (31) 式意味着

$$\boldsymbol{n}\cdot\nabla\phi^{n+1}|_{\partial\Omega} = 0,\quad \boldsymbol{n}\cdot\nabla p^{n+1/2} = \boldsymbol{n}\cdot\nabla p^{n-1/2}.$$

而这一般是不对的. 但若修改 (31) 式中的压强梯度更新公式使得其与 (30) 式一致, 即

$$q = p^{n-1/2},\quad \boldsymbol{B}(\boldsymbol{u}^*) = \boldsymbol{u}^*|_{\partial\Omega} = \boldsymbol{0},$$
$$\nabla p^{n+1/2} = \nabla p^{n-1/2} + \nabla\phi^{n+1} - \frac{\nu\Delta t}{2}\nabla\Delta\phi^{n+1}. \tag{32}$$

则对流速和压强的求解都能达到二阶精度[17].

Martin, Colella 和 Graves 提出了 BCG 方法的改进版, 称为 MCG 方法, 见 5.4 节.

3.3.2 无压投影

在 Kim 和 Moin[13] 提出的无压投影方法中, 压强近似变量 q 不出现在辅助变量 $\boldsymbol{u}^*$ 的求解过程 (Proj-1) 中:

$$q = 0, \tag{33a}$$

$$\boldsymbol{n}\cdot\boldsymbol{u}^*|_{\partial\Omega} = 0,\quad \boldsymbol{\tau}\cdot\boldsymbol{u}^*|_{\partial\Omega} = \Delta t\boldsymbol{\tau}\cdot\nabla\phi^n|_{\partial\Omega}, \tag{33b}$$

$$\nabla p^{n+1/2} = \nabla\phi^{n+1} - \frac{\nu\Delta t}{2}\nabla\Delta\phi^{n+1}, \tag{33c}$$

其中 $\boldsymbol{\tau}$ 为 $\partial\Omega$ 的单位切向量, 而 $q=0$ 这一选择导致 $\boldsymbol{u}^*$ 与 $\boldsymbol{u}^{n+1}$ 之间差别较大; 如果达到二阶精度, 其边界条件不能简单地提为 $\boldsymbol{u}^*|_{\partial\Omega} = \boldsymbol{0}$. 由于 $\boldsymbol{n}\cdot\boldsymbol{u}^{n+1}$ 的正确性总能由投影步骤 (Proj-2) 保证, 因此从理论上说 $\boldsymbol{u}^*$ 的法向分量 $\boldsymbol{n}\cdot\boldsymbol{u}^*$ 的选择具有很大的灵活性. 但从数值计算角度看, $\boldsymbol{u}^*$ 的边界条件选取方式会影响 $\boldsymbol{u}^*$ 在靠近计算域边界处的性质, 也会对压强 p 在边界处的性质产生影响. 的确, (29a) 式和 (33c) 式意味着

$$p^{n+1/2} = \phi^{n+1} - \frac{\nu}{2}\nabla\cdot\boldsymbol{u}^*.$$

在文献 [17] 中, Brown 等用数值实验展示了除非选择 $\boldsymbol{u}^*$ 的边界条件使得其在靠近边界处光滑, 否则该方法求解压强时不能达到二阶精度.

分步二阶投影方法虽然取得了巨大的成功, 但要直接将其推广到更高阶精度是非常困难的, 主要表现在定义 2 中三个不确定因素的耦合: 若改变时间积分方法, 则三个因素的选取方式全都需要重新推导, 这无疑是极其繁琐的. 其次, 辅助变量 $\boldsymbol{u}^*$ 不具有物理意义, 其边界条件必须通过具体时间积分方法的细节来进行推导. 另外, 压强和流速尽管在计算域内部是解耦的, 但它们在边界上仍然通过辅助变量 $\boldsymbol{u}^*$ 耦合在一起. 综上所述, 用分步二阶投影方法很难达到第 1 节中的五个目标. 也正是因为这些原因, 之前很多学者把投影方法称为时间分步法, 并认为它是不可能达到四阶精度的.

3.4 Gauge 方法

在二阶投影方法中, 对辅助变量 $\boldsymbol{u}^*$ 提恰当的边界条件通常比较困难. Weinan 和 Liu[15] 提出了 Gauge方法, 该格式的出发点是用规范变量 χ 来替代压强 p, 他们还引入辅助变量

$$\boldsymbol{m} := \boldsymbol{u} + \nabla\chi, \tag{34}$$

并策略性地选择 χ 和 $\boldsymbol{m}$ 使得由此导出的流速 $\boldsymbol{u}$ 和压强 p 满足 INSE. 考虑

$$\boldsymbol{m}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = \boldsymbol{g} + \nu\Delta\boldsymbol{m} \quad \text{in } \Omega, \tag{35a}$$

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on } \partial\Omega. \tag{35b}$$

由 (34) 式, (35a) 式和 (1a) 式可得

$$p = \chi_t - \nu\Delta\chi. \tag{36}$$

(34) 式和 (35) 式的优势在于可以用规范变量的自由度来对$\boldsymbol{m}$和χ提明确的边界条件, 例如

$$\boldsymbol{n}\cdot\nabla\chi = 0,\ \ \boldsymbol{n}\cdot\boldsymbol{m} = 0,\ \ \boldsymbol{\tau}\cdot\boldsymbol{m} = \boldsymbol{\tau}\cdot\nabla\chi \quad \text{on } \partial\Omega, \tag{37}$$

或

$$\chi = 0,\ \ \boldsymbol{n}\cdot\boldsymbol{m} = \boldsymbol{n}\cdot\nabla\chi,\ \ \boldsymbol{\tau}\cdot\boldsymbol{m} = 0 \quad \text{on } \partial\Omega. \tag{38}$$

对控制方程 (34) 式, (35) 式及边界条件 (37) 式离散后得到以下定义.

定义 3 Gauge 方法包括以下步骤:

(GM-1) 求解规范变量$\boldsymbol{m}^{n+1}$:

$$\frac{\boldsymbol{m}^{n+1}-\boldsymbol{m}^n}{\Delta t} = -[(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}]^{n+1/2} + \boldsymbol{g}^{n+1/2} + \frac{\nu}{2}\Delta(\boldsymbol{m}^{n+1}+\boldsymbol{m}^n), \tag{39}$$

注意, $\boldsymbol{m}^{n+1}$的边界条件要与$(\boldsymbol{m}^{n+1}-\nabla\chi^{n+1})|_{\partial\Omega} = \boldsymbol{0}$相容,

$$\boldsymbol{n}\cdot\boldsymbol{m}^{n+1} = 0,\ \ \boldsymbol{\tau}\cdot\boldsymbol{m}^{n+1} = \boldsymbol{\tau}\cdot\nabla(2\chi^n-\chi^{n-1}) \text{ on } \partial\Omega, \tag{40}$$

其中第二个等式是由外插公式 ($\chi^{n+1}\approx 2\chi^n-\chi^{n-1}$) 近似后得到的.

(GM-2) 求解流速$\boldsymbol{u}^{n+1}$:

$$\boldsymbol{u}^{n+1} = \boldsymbol{m}^{n+1} - \nabla\chi^{n+1}, \tag{41}$$

其中χ^{n+1}满足

$$\Delta\chi^{n+1} = \nabla\cdot\boldsymbol{m}^{n+1} \ \text{ in } \ \Omega, \tag{42a}$$

$$\boldsymbol{n}\cdot\nabla\chi^{n+1} = 0 \ \text{ on } \ \partial\Omega. \tag{42b}$$

(GM-3) 如果需要的话, 压强 p 可由 (36) 式得到:

$$p^{n+1/2} = \frac{\chi^{n+1}-\chi^n}{\Delta t} - \frac{\nu}{2}\Delta(\chi^{n+1}+\chi^n). \tag{43}$$

该方法对流速$\boldsymbol{u}$和压强 p 都能达到二阶收敛, 并且 (40) 式中使用的外插公式是必要的, 若仅使用滞后值χ^n则只有一阶精度[15]. 由边界条件 (40) 式可知, 该方法与前面提到的方法具有同样的问题, 即具体算法依赖于时间积分方法的细节.

3.5 PPE 方法

Pressure Poisson Equation (PPE) 方法[18–24]的核心理念是用一个压力 Poisson 方程替换流速的不可压条件:

$$\frac{\partial\boldsymbol{u}}{\partial t} = -\boldsymbol{u}\cdot\nabla\boldsymbol{u} + \boldsymbol{g} - \nabla p + \nu\Delta\boldsymbol{u} \quad \text{in } \ \Omega, \tag{44a}$$

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{on } \ \partial\Omega, \tag{44b}$$

$$\Delta p = \nabla\cdot(\boldsymbol{g} - \boldsymbol{u}\cdot\nabla\boldsymbol{u} + \nu\Delta\boldsymbol{u}) \quad \text{in } \ \Omega, \tag{44c}$$

$$\boldsymbol{n}\cdot\nabla p = \boldsymbol{n}\cdot(\boldsymbol{g} + \nu\Delta\boldsymbol{u}) \quad \text{on } \ \partial\Omega, \tag{44d}$$

其中 (44c) 式是对动量方程 (1a) 取散度后结合不可压约束 (1b) 式得到的, 而 (44d) 式是由 (44a) 式的法向分量及无滑移边界条件得到的. 若将$\nabla\cdot\boldsymbol{u} = 0$作为 (44) 式的边界条件, 则 (44) 式等价于 INSE[19].

相对于前面四小节中的方法, PPE 方法最大的优势在于可将时间积分模块完全看成“黑匣子”: 流速$\boldsymbol{u}$是 (44) 式中唯一的演化变量, 而压强 p 可视为$\boldsymbol{u}$的隐函数. 在任一时刻, 压强都可以按照 (44c) 式从流速中提取出来, 并且 p 和$\boldsymbol{u}$的边界条件不再与时间积分方法的细节相耦合. 当应用一个时间积分方法的时候, 只要给出了 (44a) 式右端项在该方法指定时刻的“样本”值, 就可以根据该方法的“契约”得到流速在时间上的高阶解. 在这个过程中, 无需知道该时间积分方法的任何细节.

PPE 方法 (44) 式也存在一些不足之处. 首先, 其推导过程假设了流速的不可压条件总是成立的; 由于 (44c) 式的推导过程依赖于 (1b) 式, 要分析近似 (1b) 式产生的误差对 INSE 解的影响是很困难的. 更重要的是, (44a) 式和 (44c) 式意味着$\dfrac{\partial\nabla\cdot\boldsymbol{u}}{\partial t} = 0$, 即流速散度的耗散是退化的, 也就是说在应用该方法时对流速散度是没有控制的. 数值实验结果显示, 直接对 PPE 方法的控制方程进行 MOL 离散得到的四阶方法是不稳定的.

3.6 UPPE 方法

利用 2.3 节中的 Laplace-Leray 交换子估计, Liu 等[22]提出了以下 UPPE (Unconstrained PPE) 方法:

$$\frac{\partial\boldsymbol{u}}{\partial t} = \boldsymbol{g} - \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nabla p + \nu\Delta\boldsymbol{u} \ \text{ in } \ \Omega, \tag{45a}$$

$$\boldsymbol{u} = \boldsymbol{0} \ \text{ on } \ \partial\Omega, \tag{45b}$$

$$\Delta p = \nabla\cdot(\boldsymbol{g} - \boldsymbol{u}\cdot\nabla\boldsymbol{u}) \ \text{ in } \ \Omega, \tag{45c}$$

$$\boldsymbol{n}\cdot\nabla p = \boldsymbol{n}\cdot(\boldsymbol{g} + \nu\Delta\boldsymbol{u} - \nu\nabla\nabla\cdot\boldsymbol{u}) \ \text{ on } \ \partial\Omega. \tag{45d}$$

该方法相对 PPE 方法的优势在于对 (45a) 式取散度后结合 (45c) 式可得

$$\frac{\partial(\nabla\cdot\boldsymbol{u})}{\partial t} = \nu\Delta(\nabla\cdot\boldsymbol{u}). \tag{46}$$

因此流速的散度由一个热方程所控制, 根据热方程的极大值原理, UPPE 格式 (45) 式对流速散度 $\nabla\cdot\boldsymbol{u}$ 的控制要比原始 PPE 格式 (44) 式可靠得多. 在有限元离散的框架下, 数值实验结果[23] 印证了 UPPE方法的稳定性.

4 GePUP: 时空一致四阶精度的广义投影方法

在把 (45) 式中的 UPPE 表述应用于基于 MOL 的有限差分和有限体积方法的时候我们遇到了困难. 首先, 直接对控制方程 (45) 进行 MOL 离散再求解常微分方程组得到的算法是不稳定的. 虽然在时间积分的过程中对流速的中间变量进行投影有时可以得到稳定的收敛结果, 但是由于控制方程 (45) 中并没有出现任何投影算子, 这个对流速中间变量进行投影的额外步骤是直接违背原方程的. 另外, 即使可以把控制方程 (45) 中的一些 $\boldsymbol{u}$ 用 $\mathscr{P}\boldsymbol{u}$ 进行替换, 也很难找到一个离散的四阶投影算子 $\boldsymbol{P}$ 满足 Leray-Helmholtz 投影性质 (10) 式. 换言之, 离散四阶投影 $\boldsymbol{P}$ 难以满足以下任何条件:

$$\boldsymbol{P}^2=\boldsymbol{P},\quad \boldsymbol{D}\boldsymbol{P}\langle\boldsymbol{u}\rangle=0,\quad \boldsymbol{P}\boldsymbol{G}\langle\phi\rangle=\boldsymbol{0}, \tag{47}$$

其中 $\langle\boldsymbol{u}\rangle$ 和 $\langle\phi\rangle$ 是定义在控制体上的平均值, $\boldsymbol{D}$ 和 $\boldsymbol{G}$ 分别是离散散度算子和离散梯度算子. 由于计算域的非周期性和四阶精度要求, 这一困难在交错网格上仍然存在.

UPPE 表述 (45) 式的推导过程依赖于 Leray-Helmholtz 投影的性质 (10) 式. 在这种情况下, 如果仍然把不可压流速场作为动力系统的演化变量, 那么用一个不满足 (47) 式的离散投影算子 $\tilde{\boldsymbol{P}}$ 来近似 Leray-Helmholtz 投影算子产生的离散误差将对数值稳定性造成怎样的影响呢? 这个问题在 UPPE 表述下是难以回答的, 我们认为这也是导致 UPPE 表述的 MOL 离散格式不稳定的根本原因.

为了在理论上刻画不满足 (47) 式的离散投影算子, 4.1 节定义了一个广义投影算子 (generic projection). 在 4.2 节中, 我们首先推导 Laplace 算子和广义投影算子的交换子, 然后在 4.3 节中将它作用在 INSE 上得到 GePUP 表述. 这三小节理论推导背后的主要思想是, 既然满足 (47) 式的离散投影算子难以找到, 投影以后的流速场也不一定能完全满足不可压条件, 那么索性将散度不为零的流速场作为动力系统的主变量. 之后四小节内容讨论在 GePUP 表述基础上的离散算法. 其中 4.4 节介绍经典的有限体积离散, 4.5 节列举了一些四阶离散算子的形式, 4.6 节简述了如何把 4.4 节和 4.5 节中对于规则区域的空间离散推广到不规则区域上, 4.7 节引入时间积分, 总结了 GePUP-IMEX 算法的主要步骤.

4.1 广义投影算子

广义投影算子是向量空间上的一个线性算子, 其输入向量与输出向量之差为一个梯度场:

$$\mathcal{P}\boldsymbol{v}^*=\boldsymbol{v}:=\boldsymbol{v}^*-\nabla\psi\quad\text{in }\ \Omega, \tag{48}$$

其中 $\nabla\cdot\boldsymbol{v}=0$ 不一定成立. 对于非周期边界条件的有界区域, 还要求广义投影算子 $\mathcal{P}$ 保持输入流场的法向边界条件, 即

$$\boldsymbol{n}\cdot\mathcal{P}\boldsymbol{v}^*=\boldsymbol{n}\cdot\boldsymbol{v}^*\ \text{on}\ \partial\Omega. \tag{49}$$

无穿透边界上的 Leray-Helmholtz 投影算子显然是一个广义投影算子.

对一个给定的广义投影算子 $\mathcal{P}$, 由 (48) 式和 (49) 式知以下 Poisson 方程

$$\Delta\psi=\nabla\cdot(\boldsymbol{v}^*-\mathcal{P}\boldsymbol{v}^*)\quad\text{in }\ \Omega,\quad \boldsymbol{n}\cdot\nabla\psi=0\ \text{ on }\partial\Omega \tag{50}$$

满足定理 1 中的可解条件, 从而可唯一确定 $\nabla\psi$. 但是通常

$$\nabla\psi\neq\boldsymbol{v}^*-\mathscr{P}\boldsymbol{v}^*=\nabla\phi.$$

例如 $\psi-\phi$ 可以是 2.3 节中提及的 Stokes 压强.

因此, (48) 式中的 $\mathcal{P}\boldsymbol{v}^*=\boldsymbol{v}^*-\nabla\psi$ 并不是对 $\boldsymbol{v}^*$ 的唯一分解, 而是由给定的 $\mathcal{P}$ 直接决定了梯度场 $\nabla\psi$. 由于许多向量场无法写成梯度场的形式, 因此 (48) 式和 (49) 式定义的广义投影算子是一种特殊的线性算子, 其特殊性在于总是为输入的 $\boldsymbol{v}^*$ 选择一个输出 $\mathcal{P}\boldsymbol{v}^*$, 使得两者之差为一个梯度场.

为什么可以用广义投影算子来刻画那些不满足 (47) 式的离散投影呢? 这是因为离散投影通常是 Leray-Helmholtz 投影的一个高阶 ($p\geqslant 4$) 近似, 从而有

$$\boldsymbol{v}^*=\mathscr{P}\boldsymbol{v}^*+\nabla\phi=\boldsymbol{P}\boldsymbol{v}^*+O(h^p)+\nabla\phi.$$

和 $\nabla\phi$ 相比, 高阶小量 $O(h^p)$ 往往可以忽略.

4.2 Laplace 算子和广义投影算子的交换子 $[\nabla,\mathcal{P}]$

首先定义梯度-散度-投影交换子为

$$[\nabla\nabla\cdot,\mathcal{P}]:=\nabla\nabla\cdot\mathcal{P}-\mathcal{P}\nabla\nabla\cdot. \tag{51}$$

由 (48) 式、Δ和∇的交换性以及 (13) 式可得

$$\Delta\mathcal{P}\boldsymbol{v}^* = \Delta(\boldsymbol{v}^* - \nabla\phi) = \Delta\boldsymbol{v}^* - \nabla\nabla\cdot\nabla\phi$$
$$= \Delta\boldsymbol{v}^* - \nabla\nabla\cdot(\boldsymbol{v}^* - \boldsymbol{v}) = \mathcal{B}\boldsymbol{v}^* + \nabla\nabla\cdot\mathcal{P}\boldsymbol{v}^*. \quad (52)$$

因此

$$\Delta\mathcal{P} = \mathcal{B} + \nabla\nabla\cdot\mathcal{P}. \quad (53)$$

对 (52) 式中第三项和最后一项应用$\mathcal{P}$, 有

$$\mathcal{P}\Delta - \mathcal{P}\nabla\nabla\cdot(\mathcal{I} - \mathcal{P}) = \mathcal{P}\mathcal{B} + \mathcal{P}\nabla\nabla\cdot\mathcal{P},$$

整理后得到

$$\mathcal{P}\Delta = \mathcal{P}\mathcal{B} + \mathcal{P}\nabla\nabla\cdot. \quad (54)$$

由 (53) 式, (54) 式和 (51) 式可得 Laplace 算子和广义投影算子的交换子为

$$[\Delta, \mathcal{P}] := \Delta\mathcal{P} - \mathcal{P}\Delta = (\mathcal{I} - \mathcal{P})\mathcal{B} + [\nabla\nabla\cdot, \mathcal{P}]. \quad (55)$$

若$\mathcal{P}$满足 (10) 式, 则$[\nabla\nabla\cdot, \mathcal{P}] \equiv \boldsymbol{0}$, 此时 (55) 式简化为 (17) 式.

4.3 INSE 的 GePUP 表述

由 4.1 节中的讨论可知, 当广义投影算子$\mathcal{P}$作用在一个散度为零的流场$\boldsymbol{u}$上时, 可以看成是对$\boldsymbol{u}$加一个梯度扰动项使其散度不一定为零. 在文献 [5] 中, 我们对动量方程 (1a) 应用广义投影算子$\mathcal{P}$并结合 (55) 式和 (51) 式推导了 INSE 的 GePUP 表述. 本文的推导过程在形式上略有不同.

由 (6) 式, (53) 式, (5) 式以及 (15) 式可得

$$\mathcal{P}\frac{\partial\boldsymbol{u}}{\partial t} = \mathcal{P}(\boldsymbol{a}^* - \nabla p) - \nu\mathcal{B}\boldsymbol{u} + \nu\Delta\mathcal{P}\boldsymbol{u} - \nu\nabla\nabla\cdot\mathcal{P}\boldsymbol{u} \quad (56)$$

$$\Rightarrow \frac{\partial\mathcal{P}\boldsymbol{u}}{\partial t} = \boldsymbol{g} - \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nabla q + \nu\Delta\mathcal{P}\boldsymbol{u} - \nu\nabla\nabla\cdot\mathcal{P}\boldsymbol{u}, \quad (57)$$

其中梯度场∇q定义为

$$\nabla q := \boldsymbol{a}^* - \mathcal{P}\boldsymbol{a} - \nu\nabla\nabla\cdot\boldsymbol{u}. \quad (58)$$

由 (6) 式及广义投影算子的定义可知 (58) 式的右端确实是个梯度场. 标量 q 可由如下 Neumann 边值问题确定:

$$\Delta q = \nabla\cdot(\boldsymbol{g} - \boldsymbol{u}\cdot\nabla\boldsymbol{u}) - \nabla\cdot\mathcal{P}\boldsymbol{a} \quad \text{in } \Omega, \quad (59a)$$

$$\boldsymbol{n}\cdot\nabla q = \boldsymbol{n}\cdot(\boldsymbol{a}^* - \nu\nabla\nabla\cdot\boldsymbol{u}) - \boldsymbol{n}\cdot\mathcal{P}\boldsymbol{a} \quad \text{on } \partial\Omega, \quad (59b)$$

其中 (59a) 式是对 (58) 式取散度后由$\nabla\cdot$和Δ的交换性得到的, 而 (59b) 式是 (58) 式的法向分量. 在无穿透边界条件下, 由 (5) 式和 (49) 式可得

$$\boldsymbol{n}\cdot\mathcal{P}\boldsymbol{a} = \boldsymbol{n}\cdot\mathcal{P}\left(\frac{\partial\boldsymbol{u}}{\partial t}\right) = \boldsymbol{n}\cdot\frac{\partial\mathcal{P}\boldsymbol{u}}{\partial t}$$
$$= \boldsymbol{n}\cdot\frac{\partial\boldsymbol{u}}{\partial t} = 0 \ \text{ on } \partial\Omega. \quad (60)$$

另外, 无滑移条件 (2) 式意味着对流项在边界上为零:

$$\boldsymbol{u}\cdot\nabla\boldsymbol{u} = \boldsymbol{0} \ \text{ on } \partial\Omega. \quad (61)$$

再结合 (5) 式知可用$\boldsymbol{g} + \nu\Delta\boldsymbol{u}$替换 (59b) 式中的$\boldsymbol{a}^*$.

定义投影流速

$$\boldsymbol{w} := \mathcal{P}\boldsymbol{u}, \quad (62)$$

并假设其散度满足

$$\nabla\nabla\cdot\boldsymbol{w} = \boldsymbol{0}, \quad (63a)$$

$$\frac{\partial\nabla\cdot\boldsymbol{w}}{\partial t} = 0, \quad (63b)$$

可以得到无滑移边界条件下 INSE 的 GePUP 表述:

$$\frac{\partial\boldsymbol{w}}{\partial t} = \boldsymbol{g} - \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nabla q + \nu\Delta\boldsymbol{w} \quad \text{in } \Omega, \quad (64a)$$

$$\boldsymbol{w} = \boldsymbol{0} \quad \text{on } \partial\Omega, \quad (64b)$$

$$\boldsymbol{u} = \mathscr{P}\boldsymbol{w} \quad \text{in } \Omega, \quad (64c)$$

$$\boldsymbol{u}\cdot\boldsymbol{n} = 0 \quad \text{on } \partial\Omega, \quad (64d)$$

$$\Delta q = \nabla\cdot(\boldsymbol{g} - \boldsymbol{u}\cdot\nabla\boldsymbol{u}) \quad \text{in } \Omega, \quad (64e)$$

$$\boldsymbol{n}\cdot\nabla q = \boldsymbol{n}\cdot(\boldsymbol{g} + \nu\Delta\boldsymbol{u} - \nu\nabla\nabla\cdot\boldsymbol{u}) \quad \text{on } \partial\Omega, \quad (64f)$$

其中 (64a) 式是由 (57) 式和假定 (63a) 式得到的, (64e) 式是由 (59a) 式和假定 (63b) 式得到的, 边界条件 (64f) 式来自于 (59b) 式, (60) 式以及 (61) 式. 由于广义投影算子对不可压流场$\boldsymbol{u}$的扰动是增加了一个梯度项, 而任何梯度都在 Leray-Helmholtz 投影算子的零空间中, 因此 (64c) 式显然成立.

在 (64) 式这个三变量系统中, 投影流速$\boldsymbol{w}$是唯一演化变量, 无源流速$\boldsymbol{u}$由$\boldsymbol{w}$唯一确定, 而压强梯度∇q又由$\boldsymbol{u}$唯一确定. 因此 (64) 式可以看成是关于$\boldsymbol{w}$的动力过程, 对其进行空间 MOL 离散之后得到一个常微分方程组, 而求解这个常微分方程组的时间积分方法可作为“黑匣子”使用, 这样就可以实现第 1 节中的 (Goal-3).

上述 GePUP 表述保持了 UPPE 表述的核心优势, 即投影流速的散度由一个热方程所控制. 确实, 对 (64a) 式取散度并结合 (64e) 式可得

$$\frac{\partial(\nabla\cdot\boldsymbol{w})}{\partial t} = \nu\Delta(\nabla\cdot\boldsymbol{w}), \quad (65)$$

若在$\partial\Omega$上指定边界条件$\nabla\cdot\boldsymbol{w}=0$, 则由热方程的极大值原理可知$\nabla\cdot\boldsymbol{w}$在$\Omega$上随时间呈指数衰减①.

GePUP 表述相对 UPPE 表述的优势有以下几点:

(I) 推导过程不依赖于任何 Leray-Helmholtz 投影算子$\mathscr{P}$的性质 (10) 式;

(II) 演化变量$\boldsymbol{w}$没有不可压限制条件;

(III) 投影算子$\mathscr{P}$出现在 (64) 式的右端, 因此用离散投影算子来近似$\mathscr{P}$时对数值精度和算法稳定性的影响非常清晰.

对一个满足不可压条件的$\boldsymbol{w}$的初始值来说, 如果 INSE 的解存在且唯一, 热方程 (65) 将使得假定 (63) 式成立. 但如果 INSE 的解本身将爆破的话, 其散度必先爆破. 因此只有当 INSE 在理论上确实有解的时候我们的计算才有意义, 换言之, 我们无法通过重新表述 INSE 的方式回避其解存在性的千禧年问题. 这是假定 (63) 式背后的主要思想.

4.4 矩形区域上的空间离散

用正方形网格划分计算域Ω, 并以多元索引$\boldsymbol{i}\in\mathbb{Z}^D$代表第$\boldsymbol{i}$个控制体$\mathcal{C}_{\boldsymbol{i}}$, 其区域为

$$\mathcal{C}_{\boldsymbol{i}}:=\left[\boldsymbol{x}_O+\boldsymbol{i}h,\ \boldsymbol{x}_O+(\boldsymbol{i}+\mathbb{1})\,h\right], \tag{66}$$

控制体$\boldsymbol{i}$在维度d上的高侧界面表示为

$$\mathcal{F}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}:=\left[\boldsymbol{x}_O+\left(\boldsymbol{i}+\boldsymbol{e}^d\right)h,\ \boldsymbol{x}_O+(\boldsymbol{i}+\mathbb{1})\,h\right], \tag{67}$$

其中$\boldsymbol{x}_O\in\mathbb{R}^D$是固定的坐标原点, h是均匀网格间距, $\mathbb{1}\in\mathbb{Z}^D$是分量全为 1 的多元索引, $\boldsymbol{e}^d\in\mathbb{Z}^D$表示第$d$个分量为 1, 其余分量全为 0 的多元索引.

如图 1 所示, ϕ在控制体$\mathcal{C}_{\boldsymbol{i}}$上的平均值定义为

$$\langle\phi\rangle_{\boldsymbol{i}}:=\frac{1}{h^D}\int_{\mathcal{C}_{\boldsymbol{i}}}\phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \tag{68}$$

而ϕ在控制体边界面$\mathcal{F}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$上的平均值定义为

$$\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}:=\frac{1}{h^{D-1}}\int_{\mathcal{F}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}}\phi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{69}$$

这两种平均值之间的关系可通过微积分基本定理得到[25]:

$$\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}=\frac{1}{12}\Big(-\langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}+7\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d}+7\langle\phi\rangle_{\boldsymbol{i}}-\langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}\Big)+O(h^4), \tag{70a}$$

$$\left\langle\frac{\partial\phi}{\partial x_d}\right\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}=\frac{1}{12h}\Big(-\langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}+15\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d}-15\langle\phi\rangle_{\boldsymbol{i}}+\langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}\Big)+O(h^4). \tag{70b}$$

由于控制体中心的点值和控制体平均值之间相差$O(h^2)$, 二阶有限体积方法不需要对它们进行区分, 而在四阶方法中则必须考虑这一差异. 此外, 作用于平均值上的有限体积算子与作用于点值上的有限差分算子是不同的.

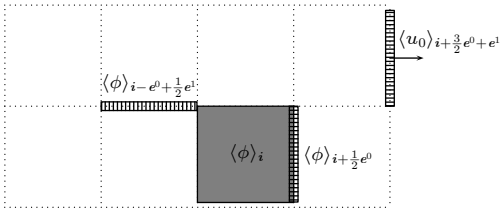


图 1 面平均值和控制体平均值. 面平均值用带$\langle\cdot\rangle$且下标为分数的符号表示, 而控制体平均值对应下标为整数. 因此$\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^0}$, $\langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^0+\frac{1}{2}\boldsymbol{e}^1}$和$\langle u_0\rangle_{\boldsymbol{i}+\frac{3}{2}\boldsymbol{e}^0+\boldsymbol{e}^1}$表示面平均值, 而$\langle\phi\rangle_{\boldsymbol{i}}$是控制体平均值[5]

Fig. 1. Notation of face-averaged and cell-averaged values. A symbol with angled brackets denotes either a cell-averaged value if the subscript is an integer multi-index, or a face-averaged value if the subscript is a fractional multi-index. Hence $\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^0}$, $\langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^0+\frac{1}{2}\boldsymbol{e}^1}$, and $\langle u_0\rangle_{\boldsymbol{i}+\frac{3}{2}\boldsymbol{e}^0+\boldsymbol{e}^1}$ are face-averaged values while $\langle\phi\rangle_{\boldsymbol{i}}$ is a cell-averaged value. Horizontal and vertical hatches represent the averaging processes over a vertical cell face and a horizontal cell face, respectively. Light gray area represents averaging over a cell[5].

4.5 四阶离散算子

离散梯度、离散散度以及离散 Laplace 算子分别定义如下:

$$\boldsymbol{G}_d\langle\phi\rangle_{\boldsymbol{i}}:=\frac{1}{12h}\Big(-\langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}+8\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d}-8\langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}+\langle\phi\rangle_{\boldsymbol{i}-2\boldsymbol{e}^d}\Big), \tag{71}$$

$$\boldsymbol{D}\langle\boldsymbol{u}\rangle_{\boldsymbol{i}}:=\frac{1}{12h}\sum_d\Big(-\langle u_d\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}+8\langle u_d\rangle_{\boldsymbol{i}+\boldsymbol{e}^d}-8\langle u_d\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}+\langle u_d\rangle_{\boldsymbol{i}-2\boldsymbol{e}^d}\Big), \tag{72}$$

$$\boldsymbol{L}\langle\phi\rangle_{\boldsymbol{i}}:=\frac{1}{12h^2}\sum_d\Big(-\langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}+16\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d}-30\langle\phi\rangle_{\boldsymbol{i}}+16\langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}-\langle\phi\rangle_{\boldsymbol{i}-2\boldsymbol{e}^d}\Big). \tag{73}$$

① 在数值算法中, 将$\nabla\cdot\boldsymbol{w}=0$和$\nabla\cdot\boldsymbol{u}=0$附加到 (64) 式中对流项以及扩散项的计算中. 在不规则边界附近, 这些条件耦合了不同流速分量之间的多维高阶多项式拟合.

离散散度算子 $\boldsymbol{D}$ 还可作用在张量平均值上

$$\boldsymbol{D}\langle \boldsymbol{uu}\rangle_{\boldsymbol{i}} \coloneqq \frac{1}{h}\sum_{d}\left(\boldsymbol{F}\langle u_d,\boldsymbol{u}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}-\boldsymbol{F}\langle u_d,\boldsymbol{u}\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}\right), \tag{74}$$

其中两标量乘积的面平均值可近似为

$$\begin{aligned}\boldsymbol{F}\langle\varphi,\psi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} &:= \langle\varphi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\langle\psi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\\ &+\frac{h^2}{12}\sum_{d'\neq d}\left(\boldsymbol{G}^{\perp}_{d'}\varphi\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\left(\boldsymbol{G}^{\perp}_{d'}\psi\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d},\end{aligned} \tag{75}$$

而 $\boldsymbol{G}^{\perp}_{d'}$ 表示横向离散梯度算子

$$\left(\boldsymbol{G}^{\perp}_{d'}\varphi\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} := \frac{1}{2h}\left(\langle\varphi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d+\boldsymbol{e}^{d'}}-\langle\varphi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d-\boldsymbol{e}^{d'}}\right). \tag{76}$$

命题 4 (71) 式—(75) 式中的离散算子都具有四阶精度:

$$\boldsymbol{G}_d\langle\phi\rangle_{\boldsymbol{i}} = \frac{1}{h^D}\int_{\mathcal{C}_{\boldsymbol{i}}}\frac{\partial\phi}{\partial x_d}\mathrm{d}\boldsymbol{x}+O(h^4), \tag{77a}$$

$$\boldsymbol{D}\langle\boldsymbol{u}\rangle_{\boldsymbol{i}} = \frac{1}{h^D}\int_{\mathcal{C}_{\boldsymbol{i}}}\nabla\cdot\boldsymbol{u}\mathrm{d}\boldsymbol{x}+O(h^4), \tag{77b}$$

$$\boldsymbol{L}\langle\phi\rangle_{\boldsymbol{i}} = \frac{1}{h^D}\int_{\mathcal{C}_{\boldsymbol{i}}}\Delta\phi\mathrm{d}\boldsymbol{x}+O(h^4), \tag{77c}$$

$$\begin{aligned}&\boldsymbol{D}\langle\boldsymbol{u}\rangle_{\boldsymbol{i}} = O(h^4)\Rightarrow\boldsymbol{D}\langle\boldsymbol{uu}\rangle_{\boldsymbol{i}}\\ =&\frac{1}{h^D}\int_{\mathcal{C}_{\boldsymbol{i}}}(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\mathrm{d}\boldsymbol{x}+O(h^4),\end{aligned} \tag{77d}$$

$$\boldsymbol{F}\langle\varphi,\psi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{h^{D-1}}\int_{\mathcal{F}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}}\varphi\psi\mathrm{d}\boldsymbol{x}+O(h^4). \tag{77e}$$

证明见文献 [4].

如图 2 所示, 我们在计算域边界之外再设置两层虚拟单元. 如果是周期性边界条件则直接把计算域内对应控制体的值复制到虚拟单元中, 否则由给定边界条件外插得到虚拟单元的平均值. 例如, 对于 Dirichlet 边界条件 $\psi=0$, 用以下五阶公式来填充虚拟单元 $\boldsymbol{i}+\boldsymbol{e}^d$ 及 $\boldsymbol{i}+2\boldsymbol{e}^d$:

$$\begin{aligned}\langle\psi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} &= \frac{1}{12}(-77\langle\psi\rangle_{\boldsymbol{i}}+43\langle\psi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}\\ &-17\langle\psi\rangle_{\boldsymbol{i}-2\boldsymbol{e}^d}+3\langle\psi\rangle_{\boldsymbol{i}-3\boldsymbol{e}^d})+O(h^5),\\ \langle\psi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d} &= \frac{1}{12}(-505\langle\psi\rangle_{\boldsymbol{i}}+335\langle\psi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}\\ &-145\langle\psi\rangle_{\boldsymbol{i}-2\boldsymbol{e}^d}+27\langle\psi\rangle_{\boldsymbol{i}-3\boldsymbol{e}^d})+O(h^5).\end{aligned} \tag{78}$$

在确定了所有虚拟单元上的平均值之后, 再用格式 (71) 式—(74) 式计算离散算子. 这样做的好处是将边界条件和连续算子的离散形式完全解耦.

用 (71) 式—(73) 式可定义离散投影算子

$$\boldsymbol{P} := \boldsymbol{I}-\boldsymbol{G}\boldsymbol{L}^{-1}\boldsymbol{D}. \tag{79}$$

由 $\boldsymbol{L}\neq\boldsymbol{DG}$ 可知 $\boldsymbol{P}^2\neq\boldsymbol{P}$. 在周期边界条件下, $\boldsymbol{P}$ 是对 Leray-Helmholtz 投影算子 $\mathscr{P}$ 的四阶近似, 并且可以证明其谱半径和 2-范数都是 1[4]. 在无滑移边界下, $\boldsymbol{P}$ 不是对称矩阵, 但是数值实验验证了 $\boldsymbol{P}$ 的谱半径不大于 1[5]. 总之, $\boldsymbol{P}$ 可以看成是有限维线性空间上的广义投影算子, 作为对 GePUP 表述 (64) 式中 Leray-Helmholtz 投影算子的近似.

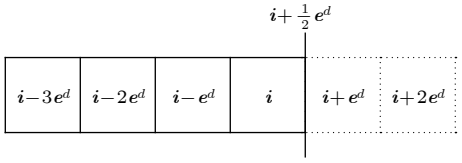


图 2 计算域边界上的两层虚拟单元 ($d=1$). 粗实线代表域边界, 细实线表示内部控制体, 而虚线表示虚拟单元[5]

Fig. 2. An example ($d=1$) of ghost filling for the domain boundary. The thick line represents a physical boundary, the thin solid lines interior cells, and the dotted lines ghost cells[5].

4.6 不规则区域上的空间离散

如果计算域 Ω 是不规则的, 我们仍然使用正方形控制体离散一个包含了 Ω 的矩形区域, 并定义控制体流相 $\mathcal{V}_{\boldsymbol{i}} := \mathcal{C}_{\boldsymbol{i}}\cap\Omega$. 当 $\mathcal{V}_{\boldsymbol{i}}=\mathcal{C}_{\boldsymbol{i}},\emptyset$ 时, 分别称 $\mathcal{C}_{\boldsymbol{i}}$ 为纯流相和无流相控制体, 这两种以外的情况称为边界控制体. 记 $\|\mathcal{V}\|$ 为控制体流相 $\mathcal{V}$ 的体积, 定义 ϕ 在 $\mathcal{V}$ 上的平均值为

$$\langle\phi\rangle_{\mathcal{V}} := \frac{1}{\|\mathcal{V}\|}\int_{\mathcal{V}}\phi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}. \tag{80}$$

对于远离边界的纯流相控制体, 空间算子 $\mathcal{L}$ 在其上的平均值 $\langle\mathcal{L}\phi\rangle_{\boldsymbol{i}}$ 可以用 4.5 节中的离散算子近似到四阶精度. 在图 3(a) 的例子中, $\mathcal{L}=\Delta$. 对于边界控制体 $\mathcal{C}_{\boldsymbol{i}}$ 或其附近的纯流相控制体, 我们也希望用一些控制体平均值的线性组合来近似 $\langle\mathcal{L}\phi\rangle_{\mathcal{C}_{\boldsymbol{i}}}$. 如图 3(b) 所示, 主要思路是用选定控制体的平均值以及边界条件拟合一个多维完全多项式. 其核心难点是如何选择 $\mathcal{V}_{\boldsymbol{i}}$ 附近的一组控制体流相使得多项式拟合的方程组有唯一解, 这个问题最终归结为在 $\mathbb{Z}^D$ 中找到 $\binom{n+D}{D}$ 个点满足一些适定条件. 我们提出了一个基于对称群作用的回溯算法[27], 见图 4.

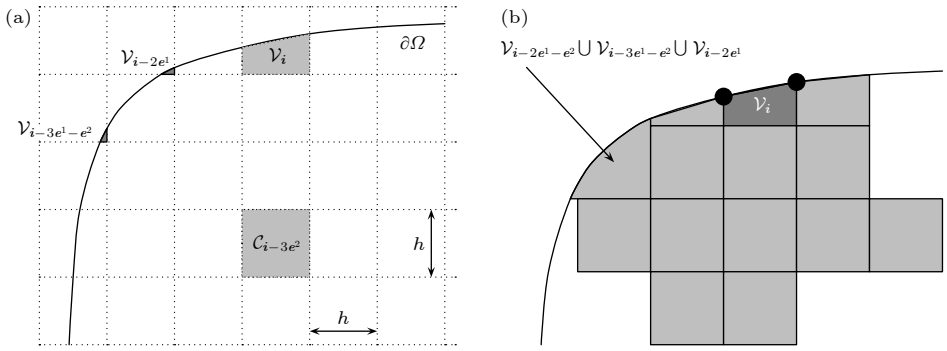


图 3 在不规则计算域上用正交网格离散 Laplace 算子 Δ[26]

Fig. 3. Discretizing the Laplacian on irregular domains with rectangular grids[26].

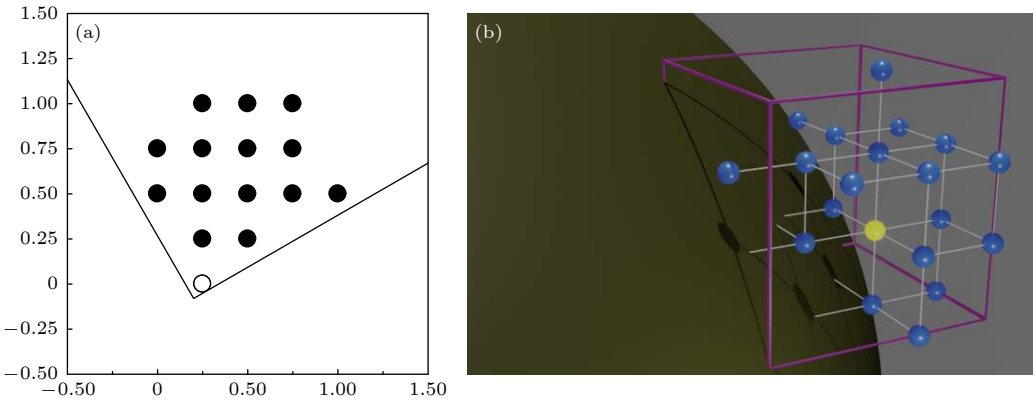


图 4 从给定起始点开始选择一组格点以拟合多维 n 次完全多项式. 用拟合的完全多项式可以离散空间算子, 并且能很方便地满足边界条件[26]

Fig. 4. For a given starting point, we choose a set of nodes to fit a high-order, multi-variate, complete polynomial, which facilitates the discretization of spatial operators and the enforcement of boundary conditions[26].

4.7 时间积分

对 (64) 式在控制体上积分, 并用离散算子近似连续算子的积分, 可得常微分方程组:

$$\frac{\mathrm{d}\langle \boldsymbol{w}\rangle}{\mathrm{d}t} = \langle \boldsymbol{g}\rangle - \boldsymbol{D}\langle \boldsymbol{u}\boldsymbol{u}\rangle - \boldsymbol{G}\langle q\rangle + \nu \boldsymbol{L}\langle \boldsymbol{w}\rangle, \tag{81a}$$

$$\boldsymbol{L}\langle q\rangle = \boldsymbol{D}\big(\langle \boldsymbol{g}\rangle - \boldsymbol{D}\langle \boldsymbol{u}\boldsymbol{u}\rangle\big), \tag{81b}$$

$$\langle \boldsymbol{u}\rangle = \boldsymbol{P}\langle \boldsymbol{w}\rangle, \tag{81c}$$

其中外力项 $\langle \boldsymbol{g}\rangle$ 已知, 未知量是控制体平均值向量 $\langle \boldsymbol{w}\rangle$, 而 $\langle \boldsymbol{u}\rangle$ 和 $\langle q\rangle$ 都可以通过 $\langle \boldsymbol{w}\rangle$ 得到. (81) 式的初始条件为 $\langle \boldsymbol{w}\rangle(t_0) = \langle \boldsymbol{u}\rangle(t_0)$, 边界条件可由 (64) 式得到.

由于 $\boldsymbol{P}$ 是对 Leray-Helmholtz 投影算子 $\mathscr{P}$ 的四阶近似, 结合命题 4 可知常微分方程组 (81) 是对 GePUP 表述 (64) 式的四阶离散.

在求解半离散系统 (81) 式时, 时间积分方法可以完全作为黑匣子使用. 在文献 [5] 中我们把 GePUP 和 IMEX 方法 [28] 耦合起来得到了一组求解 INSE的半隐式时间推进算法, 称为 GePUP-IMEX. 这里以 Kennedy 与 Carpenter[29] 提出的 ERK-ESDIRK 方法作为一个例子来阐述 GePUP-IMEX 算法.

对于

$$\frac{\mathrm{d}\psi}{\mathrm{d}t} = \boldsymbol{X}^{[\mathrm{E}]}(\psi, t) + \boldsymbol{X}^{[\mathrm{I}]}(\psi), \tag{82}$$

ERK-ESDIRK 方法在每个时间步的求解步骤如下:

$$\psi^{(1)} = \psi^n \approx \psi(t^n), \tag{83a}$$

for $s = 2,\ 3,\ \cdots,\ n_{\mathrm{s}}$,

$$\big(I - \Delta t\gamma \boldsymbol{X}^{[\mathrm{I}]}\big)\psi^{(\mathrm{s})} = \psi^n + \Delta t\sum_{j=1}^{s-1} a_{s,j}^{[\mathrm{E}]}\boldsymbol{X}^{[\mathrm{E}]}\left(\psi^{(j)}, t^{(j)}\right) + \Delta t\sum_{j=1}^{s-1} a_{s,j}^{[\mathrm{I}]}\boldsymbol{X}^{[\mathrm{I}]}\psi^{(j)}, \tag{83b}$$

$$\psi^{n+1}=\psi^{(n_{\mathrm{s}})}+\Delta t\sum_{j=1}^{n_{\mathrm{s}}}\left(b_j-a^{[\mathrm{E}]}_{n_{\mathrm{s}},j}\right)\boldsymbol{X}^{[\mathrm{E}]}\left(\psi^{(j)},t^{(j)}\right), \tag{83c}$$

其中上标 $^{(\mathrm{s})}$ 代表中间阶段, $t^{(\mathrm{s})}=t^n+c_s\Delta t$ 是该阶段对应时刻; n_s 是总阶段数; $A,\boldsymbol{b},\boldsymbol{c}$ 是 Butcher 表中的系数. $A^{[\mathrm{E}]},\boldsymbol{b}$ 和 $\boldsymbol{c}$ 表示一个 ERK 方法, 而 $A^{[\mathrm{I}]},\boldsymbol{b}$ 和 $\boldsymbol{c}$ 表示一个 ESDIRK 方法. 在每个中间阶段, $\psi^{(\mathrm{s})}$ 通过求解线性方程组 (83b) 得到, 其中右端项是结合前面步骤值计算得到的, 故 (83) 式确实属于半隐式格式. 另外 (83c) 式是由 $b^{[\mathrm{E}]}_j=b^{[I]}_j=b_j=a^{[\mathrm{I}]}_{n_{\mathrm{s}},j}$ 得到的. 有关该方法的更多细节请参考文献 [29].

对 GePUP 半离散系统 (81) 定义

$$\boldsymbol{X}^{[\mathrm{E}]}:=\langle\boldsymbol{g}\rangle-\boldsymbol{D}\langle\boldsymbol{uu}\rangle-\boldsymbol{G}\langle q\rangle;\quad \boldsymbol{X}^{[\mathrm{I}]}:=\nu\boldsymbol{L}\langle\boldsymbol{w}\rangle, \tag{84}$$

并套用 ERK-ESDIRK 方法 (83) 式的契约, 我们得到如下的 GePUP-IMEX 算法:

$$\langle\boldsymbol{w}\rangle^{(1)}=\langle\boldsymbol{w}\rangle^{n}, \tag{85a}$$

$$\begin{cases}\text{for } s=2,\ 3,\ \cdots,\ n_{\mathrm{s}},\\ \left(\boldsymbol{I}-\Delta t\nu\gamma\boldsymbol{L}\right)\langle\boldsymbol{w}\rangle^{(s)}\\ =\langle\boldsymbol{w}\rangle^{n}+\Delta t\sum\limits_{j=1}^{s-1}a^{[\mathrm{E}]}_{s,j}\boldsymbol{X}^{[\mathrm{E}]}\left(\langle\boldsymbol{u}\rangle^{(j)},t^{(j)}\right)\\ \quad+\Delta t\nu\sum\limits_{j=1}^{s-1}a^{[\mathrm{I}]}_{s,j}\boldsymbol{L}\langle\boldsymbol{w}\rangle^{(j)},\\ \langle\boldsymbol{u}\rangle^{(s)}=\boldsymbol{P}\langle\boldsymbol{w}\rangle^{(s)},\end{cases} \tag{85b}$$

$$\begin{cases}\langle\boldsymbol{w}\rangle^{*}=\langle\boldsymbol{w}\rangle^{(n_{\mathrm{s}})}+\Delta t\sum\limits_{j=1}^{n_{\mathrm{s}}}\left(b_j-a^{[\mathrm{E}]}_{n_{\mathrm{s}},j}\right)\\ \qquad\times\boldsymbol{X}^{[\mathrm{E}]}\left(\langle\boldsymbol{u}\rangle^{(j)},t^{(j)}\right),\\ \langle\boldsymbol{u}\rangle^{n+1}=\boldsymbol{P}\langle\boldsymbol{w}\rangle^{*},\\ \langle\boldsymbol{w}\rangle^{n+1}=\langle\boldsymbol{u}\rangle^{n+1},\end{cases} \tag{85c}$$

在初始时刻 $t=t_0$, 有 $\langle\boldsymbol{w}\rangle^{0}=\langle\boldsymbol{u}\rangle^{0}\approx\langle\boldsymbol{u}(t_0)\rangle$.

GePUP-IMEX 算法 (85) 式的具体步骤直接来自于将 ERK-ESDIRK 方法应用到 GePUP 的半离散系统 (81) 式上. 其中 (85b) 式的第一步对应于 (81a) 式和 (81b) 式, 而 (85b) 式的第二步对应于 (81c) 式, 即用离散投影 $\boldsymbol{P}$ 将 $\langle\boldsymbol{w}\rangle^{(s)}$ 映成 $\langle\boldsymbol{u}\rangle^{(s)}$, 以便为下一阶段计算 $\boldsymbol{X}^{[\mathrm{E}]}(\langle\boldsymbol{u}\rangle^{(j)},t^{(j)})$ 做好准备. 在 (85c) 式中, 算子 $\boldsymbol{X}^{[\mathrm{E}]}$ 中的非线性对流项导致 $\langle\boldsymbol{w}\rangle^{*}$ 的散度要远大于中间阶段 $\langle\boldsymbol{w}\rangle^{(s)}$ 的散度, 因此我们将该时间步的最终解 $\langle\boldsymbol{w}\rangle^{n+1}$ 设置为 $\langle\boldsymbol{u}\rangle^{n+1}$.

根据 (85b) 式和 (81) 式, 在每个中间阶段, 需要求解三个线性方程组.

(HLS-1) Neumann 边界条件下的 Poisson 方程 (81b): 用来从 $\langle\boldsymbol{u}\rangle$ 中提取 $\langle q\rangle$;

(HLS-2) 无滑移边界条件下的 Helmholtz 方程 (85b): 用来在时间上推进 $\langle\boldsymbol{w}\rangle$;

(HLS-3) Neumann 边界条件下的 Poisson 方程 (81c): 用来对 $\langle\boldsymbol{w}\rangle^{*}$ 进行投影.

求解这些方程组的具体算法见文献 [5]. 在 (85c) 式中, 需要求解 (HLS-1) 和 (HLS-3), 因此在每个时间步内, GePUP-IMEX 算法 ($n_{\mathrm{s}}=6$) 总共需要求解 17 个线性方程组. 我们使用的方法是标准的多重网格 V 循环, 其中光滑迭代为加权 Jacobi 方法 ($\omega=2/3$), 网格转移算子为单射算子和常值限制算子 [30]. 由于 $-\boldsymbol{L}$ 属于本性正型 [31, 32], 具有 $\boldsymbol{M}$-矩阵的良好性质, 例如正定性等. (HLS-2) 中的算子 $\left(\boldsymbol{I}-\Delta t\nu\gamma\boldsymbol{L}\right)$ 的特征值的实部全为正, 因此经典几何多重网格方法对三个线性方程组的求解都相当高效.

为了估算稳定的库朗数范围, 我们将对流项线性化, 再用与处理对流扩散方程 [25] 同样的方法可估算稳定的库朗数范围为

$$\mathrm{Cr}\leqslant 2.91/D. \tag{86}$$

综上所述, GePUP-IMEX 算法是将 IMEX 时间积分方法直接应用到 GePUP 半离散系统上得到的. 这个简单的思路对于其他时间积分方法如显式 Runge-Kutta 方法也是适用的, 具体步骤往往会更简单. 例如, 我们将经典四阶 Runge-Kutta 方法应用到 GePUP 半离散系统上得到的算法称为 GePUP-ERK 算法 [5].

5 数值实验

5.1 不规则区域上求解 INSE

在不规则区域 (倾斜正方形)

$$\boldsymbol{\Omega}=\left\{\begin{bmatrix}\cos\dfrac{\pi}{6} & -\sin\dfrac{\pi}{6}\\ \sin\dfrac{\pi}{6} & \cos\dfrac{\pi}{6}\end{bmatrix}\begin{pmatrix}\xi\\ \eta\end{pmatrix}+\begin{pmatrix}\dfrac{1}{2}\\ 0\end{pmatrix}:(\xi,\eta)\in[0,1]^2\right\} \tag{87}$$

上给定流速和压强的精确解

$$\boldsymbol{u}(x,y,t)=\begin{pmatrix}\sin^2(\pi x)\sin(2\pi y)\\ -\sin(2\pi x)\sin^2(\pi y)\end{pmatrix}\pi\cos t, \tag{88}$$

$$p(x,y,t)=-\cos(\pi x)\sin(\pi y)\cos t, \tag{89}$$

可以根据动量方程 (1a) 推导出本节测试中使用的外力项$\boldsymbol{g}$.

在计算域 (87) 上, 用 (88) 式设置流速场的初始条件和无滑移边界条件, 用 GePUP-ERK 算法将 INSE 的解从$t_0=0$推进到$t_e=0.1$. 表 1 列出了最终时刻结果的误差和收敛阶, 显然流速$\boldsymbol{u}$和压强 p 都能达到四阶收敛.

5.2 三维顶盖驱动方腔流

三维顶盖驱动方腔内的不可压缩流动几乎表现出流体力学中所有常见的现象: 漩涡、二次流、混沌粒子运动、不稳定性、过渡、湍流和如展向涡的复杂三维流动结构[33], 因此其数值模拟具有重要的科学意义. 另外由于几何设定非常简单, 顶盖驱动方腔流是检验 INSE 数值求解方法最常使用的基准测试之一.

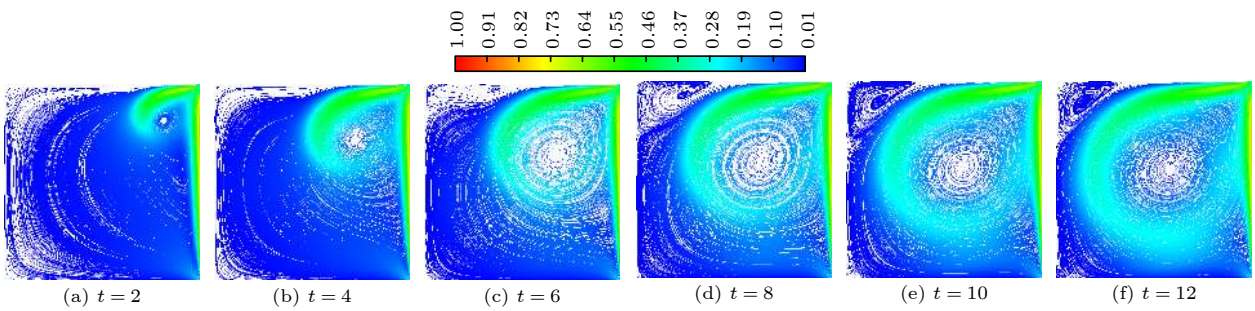
在规则计算区域$[0,1]\times[0,1]\times[0,2]$上, 用 GePUP-IMEX 算法 (85) 式对三维顶盖驱动方腔流进行数值模拟. 选择的具体参数为: 流速场$\boldsymbol{u}=(u,v,w)$的边界条件为在$x=1$处, $v=1$, 其余边界上$\boldsymbol{u}=\boldsymbol{0}$; 而在初始时刻, $\boldsymbol{u}$在计算域内全为$\boldsymbol{0}$, $t_0=0$及 $t_e=12$. 雷诺数 $Re=1000$, 库朗数 $\mathsf{Cr}=0.5$, 均匀网格间距$h=1/128$. 下面的数值结果可以在文献 [5] 中找到.

图 5 给出了在时刻$t=2,4,6,8,10,12$, 对称平面$z=1$内流速场数值解的流线. 数值模拟刚开始时, 盖子的移动会沿腔盖产生很大的剪切应力, 并在$t=2$之前形成一个漩涡: 位于漩涡中心控制体的流速要比附近所有控制体流速小 50 倍左右. 随着模拟继续进行, 逆时针旋转漩涡附近的高速流体逐渐穿透最初静止的流体, 在$t=4$之后不久, 又有另外一个顺时针旋转的漩涡开始形成, 并在$t=6$变得显著. 然后二次涡开始增强并移动到计算域左上角. 最终在$t=12$时刻, 主涡、二次涡和流速场达到稳定状态. 这些特征与文献 [34] 中的实验结果完全符合. 三维与二维顶盖驱动方腔流动的一个主要区别是沿翼展方向壁端涡旋的演化和对称面附近的 Taylor-Görtler 型涡旋, 图 6 清晰地显示了时刻$t=8,10,12$的这些突出特征.

表 1 用 GePUP-ERK 算法在不规则计算域 (87) 上求解以 (88) 式和 (89) 式为解析解的 INSE 得到的误差和收敛阶. $Re=1000$, $t_0=0$, $t_e=0.1$, $\mathsf{Cr}=0.2$[5]

Table 1. Errors and convergence rates of GePUP-ERK for solving the INSE on the irregular domain (87) with Eq. (88) and Eq. (89) as the analytic solutions. $Re=1000$, $t_0=0$, $t_e=0.1$, $\mathsf{Cr}=0.2$[5].

h	1/32	Rate	1/64	Rate	1/128	Rate	1/256
$\boldsymbol{u}$ L_∞	2.63×10^{-4}	3.94	1.71×10^{-5}	3.98	1.08×10^{-6}	4.00	6.78×10^{-8}
$\boldsymbol{u}$ L_1	1.63×10^{-4}	3.87	1.12×10^{-5}	3.97	7.14×10^{-7}	3.99	4.48×10^{-8}
$\boldsymbol{u}$ L_2	1.49×10^{-4}	3.89	1.01×10^{-5}	3.98	6.38×10^{-7}	4.00	4.00×10^{-8}
p L_∞	1.63×10^{-3}	3.99	1.02×10^{-4}	4.00	6.39×10^{-6}	4.01	3.97×10^{-7}
p L_1	3.26×10^{-4}	4.00	2.04×10^{-5}	4.03	1.25×10^{-6}	3.97	7.97×10^{-8}
p L_2	4.60×10^{-4}	3.99	2.89×10^{-5}	4.01	1.80×10^{-6}	3.99	1.13×10^{-7}



(a) $t=2$ (b) $t=4$ (c) $t=6$ (d) $t=8$ (e) $t=10$ (f) $t=12$

图 5 三维顶盖驱动方腔流测试中均匀网格上 ($h=1/128$) 对称平面$z=1$内的流线快照. 不同颜色代表不同的$\|\boldsymbol{u}\|_2$[5]

Fig. 5. Snapshots of streamlines for the 3D lid-driven cavity test inside the symmetry plane $z=1$ on a uniform grid with $h=1/128$. Different colors represent different values of $\|\boldsymbol{u}\|_2$[5].

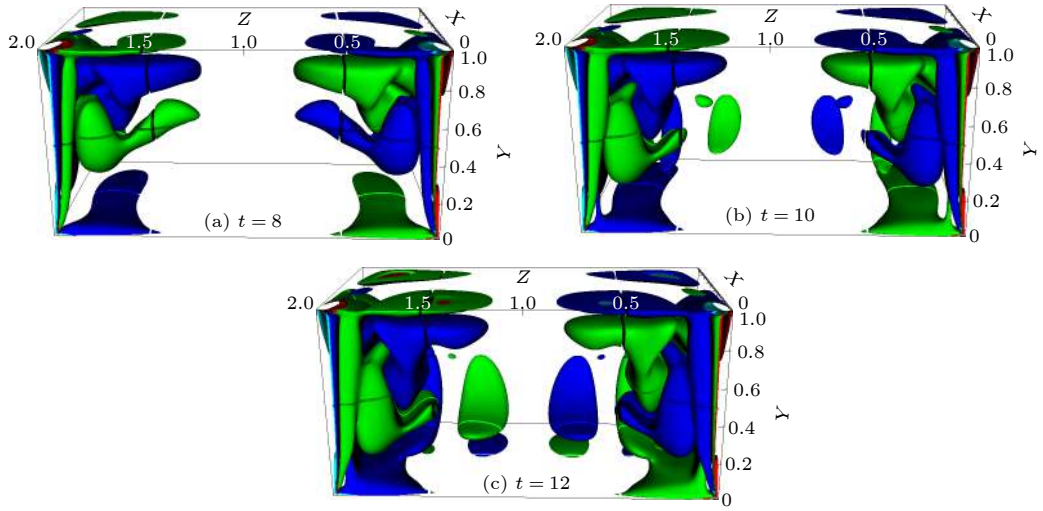


图 6 三维顶盖驱动方腔流测试中均匀网格上 ($h=1/128$)涡量的 x-分量等值面. 红色, 橙色, 蓝色和青色对应的涡量 x-分量值分别为–0.50, –0.25, 0.25 和 0.50[5]

Fig. 6. Isosurfaces of the x component of the vorticity vector for the 3D lid-driven cavity test on a uniform grid with $h=1/128$. The values of the vorticity component for the red, orange, blue, cyan surfaces are –0.50, –0.25, 0.25, and 0.50, respectively[5].

本文还对 GePUP-IMEX 算法结果和 Guermond 等[34] 的实验和计算结果进行了定量比较: 图 7 和图 8 分别给出了主漩涡的中心位置和在平面 $z=1, 0.5$内的流速剖面图. 可以看到, GePUP-IMEX 算法的计算结果和 Guermond 等[34] 的实验和计算结果都非常符合. 此外, GePUP-IMEX 算法的某些计算结果, 例如流速的 v 分量, 要比 Guermond 等[34] 的计算结果更加符合实验数据.

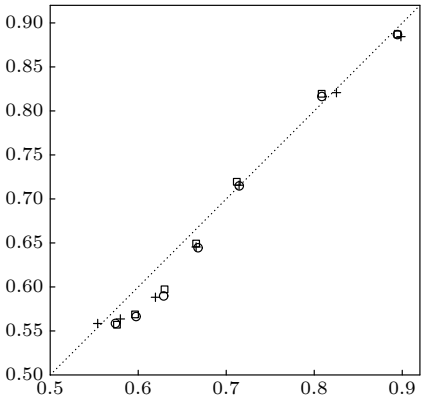


图 7 时刻 $t=1,2,4,6,8,10,12$ 对称平面 $z=0$ 上主旋涡的中心位置比较. "○"表示本文的数值结果 (由最小化 $\|\boldsymbol{u}\|_2$ 得到), 而"+"和"□"分别是 Guermond 等[34] 得到的实验和计算结果[5]

Fig. 7. A comparison of the center locations of the primary eddy in the symmetry plane $z=0$ at time instances $t=1,2,4,6,8,10,12$. The circles represent numerical results of this work, which are determined by minimizing the speed $\|\boldsymbol{u}\|_2$. The experimental and computational results of Guermond et. al.[34] are represented by the crosses and the squares, respectively[5].

5.1 节验证了基于 GePUP 表述的 INSE 投影方法能够达到时空一致四阶精度. 在这个基础上, 本节的结果进一步验证了 GePUP 方法精确捕捉实际物理过程的能力.

5.3 二维顶盖驱动方腔流

本节在倾斜正方形 ((87) 式) 上模拟二维顶盖驱动方腔流, 并将得到的数值结果与对应规则区域 $[0,1]^2$ 上的数值结果进行比较. 在由点 $P=\left(\dfrac{\sqrt{3}}{2},\dfrac{\sqrt{3}}{2}+\dfrac{1}{2}\right)$ 和点 $Q=\left(\dfrac{\sqrt{3}}{2}+\dfrac{1}{2},\dfrac{1}{2}\right)$ 确定的域边界线段上, 流速场 $\boldsymbol{u}$ 的边界条件为

$$\boldsymbol{u}(\boldsymbol{x},t)=\begin{pmatrix}-1/2\\ \sqrt{3}/2\end{pmatrix}\eta(r)(1-\mathrm{e}^{-\omega t})\mathrm{e}, \tag{90}$$

其中

$$\eta(\alpha)=\begin{cases}\mathrm{e}^{-\frac{1}{1-\alpha^2}} & \text{if } |\alpha|<1,\\ 0 & \text{otherwise,}\end{cases}$$
$$r=2\|\boldsymbol{x}-P\|_2-1,\quad \omega=2. \tag{91}$$

其余计算域边界上 $\boldsymbol{u}=\boldsymbol{0}$, 而在初始时刻, $\boldsymbol{u}$ 在计算域内全为 $\boldsymbol{0}$, $t_0=0$, $t_\mathrm{e}=60$, $\mathrm{Cr}=0.1$. 雷诺数 $Re=1000$, 均匀网格间距 $h=1/128$.

图 9 和图 10 分别给出了在时刻 $t=4,8,32$ 时的流速场和涡量快照. 在规则区域和不规则区域上, 这两个物理量的特征完全一致.

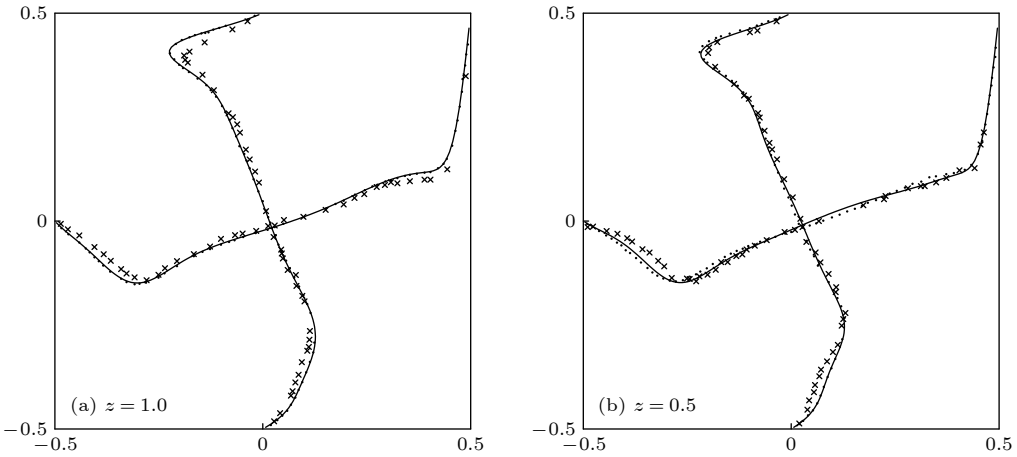


图 8 三维顶盖驱动方腔流在时刻 $t = 12$ 的流速剖面图比较 ($h = 1/128$). 实线表示本文的数值结果, 即 $-\frac{1}{2}u$ 和 $\frac{1}{2}v$ 分别作为 $\frac{1}{2} - y$ 和 $\frac{1}{2} - x$ 的函数. "×"和"."分别表示 Guermond 等[34] 得到的实验和计算结果[5]

Fig. 8. A comparison of velocity profiles for the three dimensional lid-driven cavity test ($h = 1/128$) at $t = 12$. Solid lines represent computational results of this work, i.e. $-\frac{1}{2}u$ as a function of $\frac{1}{2} - y$ and $\frac{1}{2}v$ as a function of $\frac{1}{2} - x$. The computational and experimental results of Guermond et. al.[34] are represented by the solid dots and crosses, respectively[5].

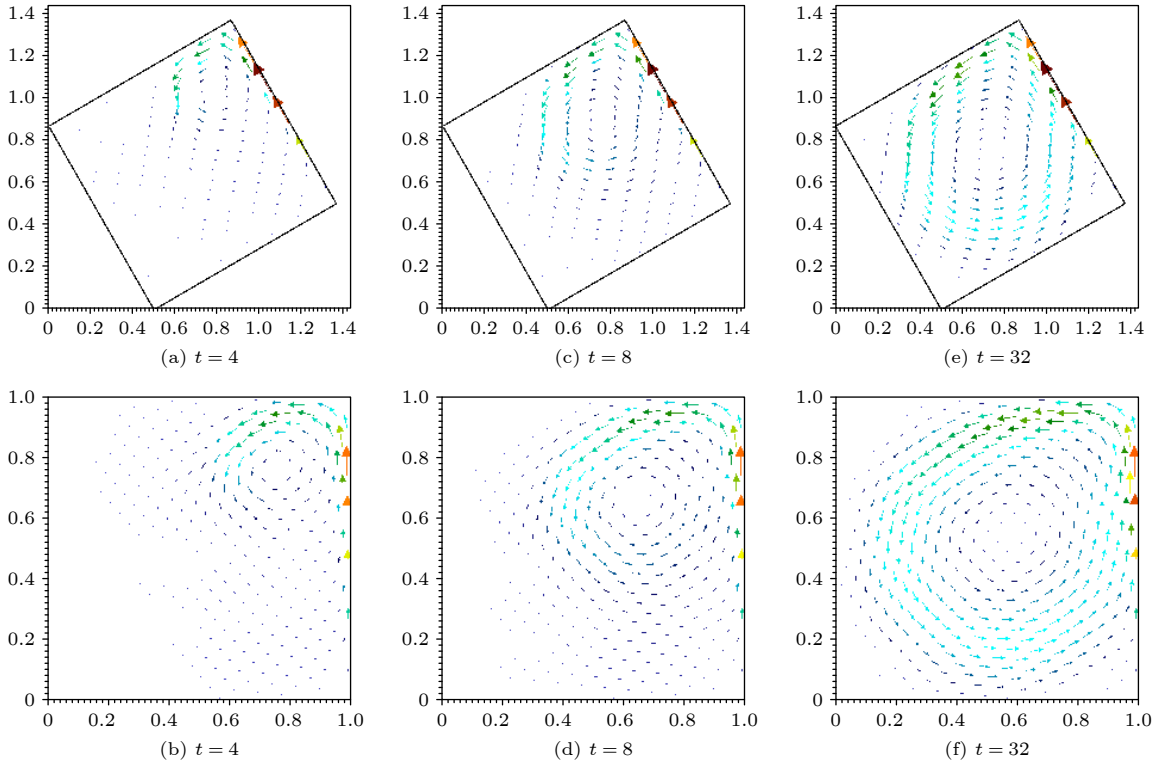


图 9 二维顶盖驱动方腔流测试中均匀网格上 ($h = 1/128$) 的流速场

Fig. 9. Numerical results of the velocity field in the two dimensional lid-driven cavity test on the unit box and the rotated box with $h = 1/128$.

5.4 GePUP 与一个经典二阶方法的性能比较

尽管在许多应用中[35–37], 高阶方法都被认为要优于二阶方法, 但有学者也报告了某些四阶方法的效率事实上是不尽如人意的[38–40], 原因是这些"四阶方法"[38–40] 仅能在空间上达到四阶精度, 在时间上的精度仍为二阶. 由于求解误差主要是由二阶时间误差主导的, 这些方法在求解非定常流动时精度和效率都相对较低. 因此即使一个方法在空间上具有高精度甚至谱精度, 如果时间上只有二阶精

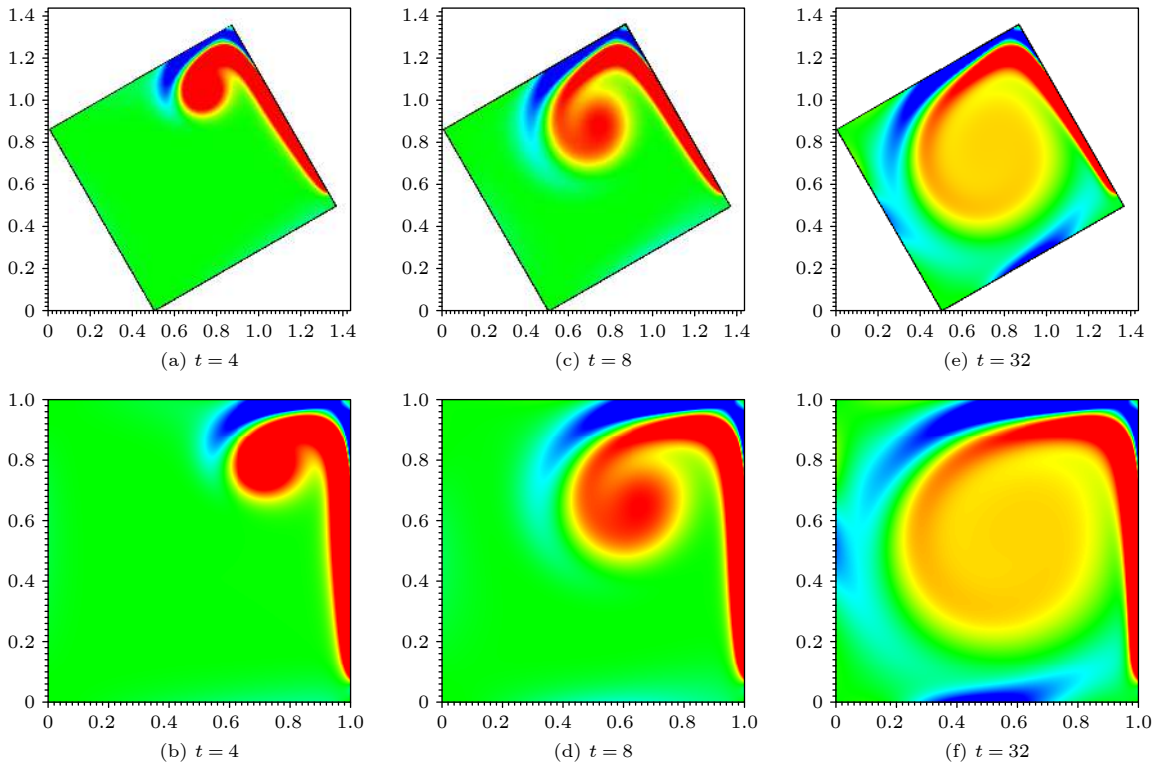


图 10 二维顶盖驱动方腔流测试中均匀网格上 ($h=1/128$) 的涡量快照

Fig. 10. Numerical results of the vorticity in the two dimensional lid-driven cavity test on the unit box and the rotated box with $h=1/128$.

度, 我们仍然认为该方法是一个二阶方法. 另一方面, 这表明要想获得一个真正的四阶方法并充分发挥其优势, 必须保证时间上的四阶精度. 现有文献中对四阶方法与二阶方法的比较缺乏系统的论述, 本节回顾文献 [5] 中给出的一个性能比较公式.

在比较具有不同收敛阶的 INSE 数值方法的效率时, 作以下两个假定:

(ECA-1) 求解线性方程组的时间远大于显式计算离散算子所耗费的时间;

(ECA-2) 求解单个线性方程组消耗的 CPU 时间与其未知量个数成线性关系.

基于这两个假定, 有:

命题 5 定义一个ξ阶方法相对于一个η阶方法的性能加速比为

$$S_{\xi|\eta}(\epsilon):=\frac{T_\eta(\epsilon)}{T_\xi(\epsilon)}, \tag{92}$$

其中$\xi,\eta\in\mathbb{R}^+$, $\xi\geqslant\eta$, $T_\xi(\epsilon)$和$T_\eta(\epsilon)$分别是ξ阶方法和η阶方法达到给定精度ϵ需要的 CPU 时间. 则对于任一给定问题有[5]:

$$S_{\xi|\eta}(\epsilon)=\frac{1}{n_{\rm T}}\left(\frac{E_\eta}{E_\xi}\right)^{\frac{D+1}{\eta}}\left(\frac{E_\xi}{\epsilon}\right)^{\frac{(\xi-\eta)(D+1)}{\xi\eta}}, \tag{93}$$

其中$n_{\rm T}$是在同一网格下求解该问题时 ξ阶方法与η阶方法所需 CPU 时间的比值, 而E_ξ和E_η分别是该固定网格下相应的误差, $E_\xi\leqslant E_\eta$.

证明 设$h_\xi(\epsilon)$和$h_\eta(\epsilon)$分别是该ξ阶方法和η阶方法为达到给定精度ϵ所取的网格间距, 则

$$h_\xi(\epsilon)=\Theta\left(\epsilon^{1/\xi}\right),\qquad h_\eta(\epsilon)=\Theta\left(\epsilon^{1/\eta}\right), \tag{94}$$

其中$f=\Theta(g)$是指存在常数$c_l,c_u>0$使得 $c_lg\leqslant f\leqslant c_ug$.

根据假定 (ECA-1) 和 (ECA-2), 可得

$$T_\xi(\epsilon)=c_\xi\left(\frac{1}{h_\xi(\epsilon)}\right)^{D+1},\quad T_\eta(\epsilon)=c_\eta\left(\frac{1}{h_\eta(\epsilon)}\right)^{D+1}, \tag{95}$$

其中c_ξ和c_η是两个正常数, $\left(\frac{1}{h_\xi(\epsilon)}\right)$和$\left(\frac{1}{h_\eta(\epsilon)}\right)$的幂次中的"+1"是由于初始值需要被推进$\Theta(1/h)$个时间步才能得到最终解. 由性能比$S_{\xi|\eta}(\epsilon)$的定义 (92) 式, (94) 式和 (95) 式可得

$$S_{\xi|\eta}(\epsilon)=\frac{T_\eta(\epsilon)}{T_\xi(\epsilon)}=c_{\xi|\eta}\epsilon^{-\frac{(\xi-\eta)(D+1)}{\xi\eta}}, \tag{96}$$

其中 $c_{\xi|\eta}$ 对于该给定问题是个常数. 在相同网格上, E_ξ 和 E_η 是运行 ξ 阶方法和 η 阶方法得到的精度, 而 $n_{\rm T}$ 是这两个方法消耗 CPU 时间的比值. 假设 η 阶方法的精度由 E_η 提升到 E_ξ 需要对网格加密 $n_{\eta\to\xi}$ 次, 且相邻两网格间距比率为 r, 则

$$n_{\eta\to\xi} = \log_{r^\eta}\frac{E_\eta}{E_\xi}, \tag{97a}$$

$$\frac{T_\eta(E_\xi)}{T_\xi(E_\xi)} = \frac{r^{(D+1)n_{\eta\to\xi}}}{n_{\rm T}}, \tag{97b}$$

其中 (97a) 式中 r^η 的幂次 η 是 η 阶方法的收敛阶, (97b) 式是由 (95) 式得到的. 在 (96) 式中取 ϵ 为 E_ξ, 再结合 (97b) 式可得

$$c_{\xi|\eta} = \frac{1}{n_{\rm T}}(E_\xi)^{\frac{(\xi-\eta)(D+1)}{\xi\eta}}\left(\frac{E_\eta}{E_\xi}\right)^{\frac{D+1}{\eta}}. \tag{98}$$

将 (98) 式代入 (96) 式即可得到 (92) 式.

利用 (93) 式, 可以对四阶投影方法 GePUP-IMEX 和 Martin 等 [7] 提出的经典二阶投影方法 MCG[①]的效率作比较.

表 2 给出了在不同的测试中为了达到同样的 L_2 流速计算精度 ϵ, GePUP-IMEX 相对于 MCG 的性能加速比 $S_{4|2}$. 对于较低的相对精度, 如 $\epsilon = 10^{-4}$, GePUP-IMEX 相对于 MCG 在效率上的优势还不太明显, 而要达到适中的相对精度, 如 $\epsilon = 10^{-6}$, 10^{-8}, GePUP-IMEX 的性能优势是相当明显的. 尽管对于其他问题, $S_{4|2}$ 的具体值可能与表 2 中的结果不同, 但是 $S_{4|2}(\epsilon)$ 是 $1/\epsilon$ 的幂函数这一增长模式是始终保持不变的. 此外, 即使数值精度不是首要考虑的因素, GePUP-IMEX 也允许使用更粗糙的网格来获得与 MCG 相同的精度, 因此在实际计算时 GePUP-IMEX 依然具有巨大的效率优势.

由于涡量 $\nabla\times\boldsymbol{u}$ 是由流速场的一阶导数得到的, GePUP-IMEX 和 MCG 对涡量的计算在 L_∞ 范数下分别具有三阶和一阶精度, 表 3 给出了在不同测试中为了达到同样的 L_∞ 涡量计算精度 ϵ, GePUP-IMEX 相对于 MCG 的性能加速比 $S_{3|1}$. 即使对于很低的相对精度, 如 $\epsilon = 10^{-2}$, GePUP-IMEX 相对于 MCG 在效率上的优势也是非常明显的.

以上展示的测试均是在作者的个人电脑 (Intel® i7-3930, 138 GFlop/s) 上进行的, 截至 2020 年 11 月,世界上最快的超级计算机是日本的 Fugaku (7630848 核, peak performance 537212 TFlop/s)[43], 因此由个人电脑切换到超级计算机上进行计算所带来的性能提升最大为 4×10^6. 而注意到表 2 和表 3 中许多结果是大于 4×10^6 的, 这说明为了在一些测试问题中达到给定精度要求, 在作者的个人电脑上运行四阶 GePUP-IMEX 算法比在国际上最快的超级计算机上运行二阶 MCG 方法都要快!

在数值模拟高雷诺数不可压流体的物理过程时, 往往要用到依赖于流速的一阶和二阶导数的物理量, 例如涡量和曲率等. 表 3 中的结果说明了四阶 GePUP-IMEX 投影方法对精确捕捉高雷诺数

表 2 不同的测试中为了达到同样的 L_2 流速计算精度 ϵ, GePUP-IMEX 相对于 MCG[7] 的性能加速比 $S_{4|2}$ [5]

Table 2. The speedup $S_{4|2}$ of GePUP-IMEX over MCG[7] for achieving the same L_2 accuracy ϵ of the velocity[5].

$S_{4\|2}$	$\epsilon=10^{-4}$	$\epsilon=10^{-6}$	$\epsilon=10^{-8}$	$\epsilon=10^{-10}$
单涡量测试, $Re=2\times10^4$	1.73	5.50×10^1	1.73×10^3	5.47×10^4
二维粘盒测试, $Re=10^4$	5.03	1.59×10^2	5.03×10^3	1.59×10^5
二维粘盒测试, $Re=100$	1.26×10^1	4.00×10^2	1.26×10^4	3.98×10^5
三维粘盒测试, $Re=10^4$	8.65×10^1	8.65×10^3	8.65×10^5	8.65×10^7
三维粘盒测试, $Re=100$	4.24×10^1	4.24×10^3	4.24×10^5	4.24×10^7

表 3 不同的测试中为了达到同样的 L_∞ 涡量计算精度 ϵ, GePUP-IMEX 相对于 MCG[7] 的性能加速比 $S_{3|1}$ [5]

Table 3. The speedup $S_{3|1}$ of GePUP-IMEX over MCG[7] for achieving the same L_∞ accuracy ϵ of the vorticity[5].

$S_{3\|1}$	$\epsilon=10^{-2}$	$\epsilon=10^{-3}$	$\epsilon=10^{-4}$	$\epsilon=10^{-5}$
单涡量测试, $Re=2\times10^4$	1.89×10^4	1.89×10^6	1.89×10^8	1.89×10^{10}
二维粘盒测试, $Re=10^4$	2.30×10^5	2.30×10^7	2.30×10^9	2.30×10^{11}
三维粘盒测试, $Re=10^4$	3.66×10^7	1.70×10^{10}	7.88×10^{12}	3.66×10^{15}

① MCG 是在 C++软件包 Chombo[41] 的基础上实现的, 其源代码见文献 [42].

流体涡量场的演化过程具有重要的科学意义.

随着流体力学的飞速发展, 我们越来越需要研究具有多长度尺度和多时间尺度的复杂物理问题, 这对数值模拟在精度上和效率上都提出了越来越高的要求. 四阶和更高阶精度的算法极大地提高了计算精度和计算效率, 若进一步将这些算法与并行计算和自适应网格结合, 我们将能解决更多的实际物理问题.

6 结 论

本文首先回顾了一些求解不可压 Navier-Stokes 方程的经典数值方法, 然后基于广义投影算子和 UPPE 表述推导了 GePUP 表述和相应的数值方法. GePUP 方法最突出的优势在于时间积分方法和空间离散格式是完全解耦的, 因此这两个算法模块都能够作为"黑匣子"进行处理. 时间积分方法的灵活自由选择便于实现 INSE 解在时间上的高阶精度, 并使得 GePUP 格式同时适用于高低雷诺数流体. 空间离散的灵活自由选择使得算法可以很好地适应不规则边界. 数值测试结果印证了该方法的四阶精度及其捕捉实际物理过程的能力, 再结合一个简单的效率比较公式就可以看出四阶 GePUP 方法相对于现有二阶投影方法在效率上和精度上都具有巨大优势.

目前我们已经将 GePUP 方法推广到静止不规则区域上, 我们计划将其与高阶界面追踪方法 [44,45] 耦合, 针对动边界不可压流体提出一个时空一致四阶精度的投影算法.

参考文献

[1] Smale S 1998 *Mathematical Intelligencer* **20** 7
[2] Devlin K 2005 *The Millennium Problems: the Seven Greatest Unsolved Mathematical Puzzles of Our Time* (Granta Books) pp103–136
[3] Carlson J, Jaffe A, Wiles A, Editors 2006 *The Millennium Prize Problems* (American Mathematical Society) pp57–67
[4] Zhang Q 2014 *Appl. Numer. Math.* **77** 16
[5] Zhang Q 2016 *J. Sci. Comput.* **67** 1134
[6] Bell J B, Colella P, Glaz H M 1989 *J. Comput. Phys.* **85** 257
[7] Martin D F, Colella P, Graves D 2008 *J. Comput. Phys.* **227** 1863
[8] Taylor M E 2011 *Partial Differential Equations I. No. 115 in Applied Mathematical Sciences* (2nd Ed.) (Springer) pp408–409
[9] Sanderse B, Koren B 2012 *J. Comput. Phys.* **231** 3041
[10] Griffth B E 2009 *J. Comput. Phys.* **228** 7565
[11] Benzi M, Golub G H, Liesen J 2005 *Acta Numer.* **14** 1
[12] Chorin A J 1968 *Math. Comput.* **22** 745
[13] Kim J, Moin P 1985 *J. Comput. Phys.* **59** 308
[14] Orszag S A, Israeli M, Deville M O 1986 *J. Sci. Comput.* **1** 75
[15] Weinan E, Liu J G 2003 *Comm. Math. Sci.* **1** 317
[16] Guermond J L, Minev P, Shen J 2006 *Comput. Methods Appl. Mech. Engrgy* **195** 6011
[17] Brown D L, Cortez R, Minion M L 2001 *J. Comput. Phys.* **168** 464
[18] Kleiser L, Schumann U 1980 *Proceedings of the Third GAMM-Conference on Numerical Methods in Fluid Mechanics, vol. 2 of Notes on Numerical Fluid Mechanics* (Springer) pp 165–173
[19] Gresho P M, Sani R L 1987 *Int. J. Numer. Methods Fluids* **7** 1115
[20] Henshaw W D 1994 *J. Comput. Phys.* **113** 13
[21] Johnston H, Liu J G 2004 *J. Comput. Phys.* **199** 221
[22] Liu J G, Liu J, Pego R L 2007 *Commun. Pure Appl. Math.* **60** 1443
[23] Liu J G, Liu J, Pego R L 2010 *J. Comput. Phys.* **229** 3428
[24] Shirokoff D, Rosales R R 2011 *J. Comput. Phys.* **230** 8619
[25] Zhang Q, Johansen H, Colella P 2012 *SIAM J. Sci. Comput.* **34** B179
[26] Li Z, Zhang Q 2021 *submitted for publication*
[27] Zhang Q, Li Z 2021 *submitted for publication*
[28] Ascher U M, Ruuth S J, Spiteri R J 1997 *Appl. Numer. Math.* **25** 151
[29] Kennedy C A, Carpenter M H 2003 *Appl. Numer. Math.* **44** 139
[30] Briggs W L, Henson V E, McCormick S F 2000 *A Multigrid Tutorial* (2nd Ed.) (SIAM) pp 31–43
[31] Brandt A 1986 *Appl. Math. Comput.* **19** 23
[32] Trottenberg U, Oosterlee C, Schuller A 2001 *Multigrid* (Elsevier Academic Press) pp439–441
[33] Shankar P N, Deshpande M D 2000 *Annu. Rev. Fluid Mech.* **32** 93
[34] Guermond J L, Migeon C, Pineau G, Quartapelle L 2002 *J. Fluid. Mech.* **450** 169
[35] Verstappen R, Veldman A 1997 *J. Eng. Math.* **32** 143
[36] Hokpunna A, Manhart M 2010 *J. Comput. Phys.* **229** 7545
[37] Almgren A, Aspden A, Bell J B, Minion M L 2013 *SIAM J. Sci. Comput.* **35** B25
[38] Gullbrand J 2000 *An Evaluation of a Conservative Fourth Order DNS Code in Turbulent Channel* Center for Turbulence Research, Stanford University
[39] Shishkina O, Wagner C 2007 *Comput. Fluids* **36** 484
[40] Knikker R 2009 *Int. J. Numer. Methods Fluids* **59** 1063
[41] Adams M, Colella P, Graves D T, Johnson J, Keen N, Ligocki T J, Martin D F, McCorquodale P, Modiano D, Schwartz P, Sternberg T, Straalen B V 2014 *Chombo Software Package for AMR Applications-Design Document* Lawrence Berkeley National Laboratory. LBNL-6616 E
[42] Chombo http://seesar.lbl.gov/ANAG/software.html. Version 3.2
[43] Meuer H, Strohmaier E, Dongarra J, Simon H http://www.top500.org [2020-11-25]
[44] Zhang Q, Fogelson A 2016 *SIAM J. Numer. Anal.* **54** 530
[45] Zhang Q 2018 *SIAM J. Sci. Comput.* **40** A3755

YOUNG SCIENTISTS' FORUM

Projection methods for the incompressible Navier-Stokes equations*

Zhang Qing-Hai† Li Yang

(*School of Mathematical Sciences, Zhejiang University, Hangzhou* 310027, *China*)

(Received 3 February 2021; revised manuscript received 9 March 2021)

Abstract

The incompressible Navier-Stokes equations (INSE) are the basic governing equations of fluid dynamics, and their numerical solutions are of great significance. In this review paper, we first recollect some classical projection methods and their relatives in the past 50 years and then fully explain the recent fourth-order projection method called GePUP [Zhang Q 2016 *J. Sci. Comput.* **67** 1134]. Based on a generic projection operator and the UPPE formulation of the INSE [Liu J G, Liu J, Pego R L 2007 *Comm. Pure Appl. Math.* **60** 1443], we derive the GePUP equations, which retain the advantage of UPPE that the velocity divergence is governed by a heat equation and is thus well under control. In comparison with UPPE, the GePUP formulation is advantageous in three aspects: (1) its derivation depends on none of the properties of the Leray-Helmholtz projection; (2) the evolutionary velocity can be divergent, thus it is directly applicable to numerical calculations with nonzero velocity divergence; (3) the Leray-Helmholtz projection does appear on the right-hand sides of the governing equations, thus making it transparent to analyze the accuracy and stability issues raised by numerically approximating the Leray-Helmholtz projection. As the most appealing feature of GePUP, temporal integration and spatial discretization are completely decoupled and can be treated as black boxes, so that the user can choose his favorite methods for the two parts to form his own GePUP method. In particular, high-order accuracy in time can be easily obtained since no internal details of the ODE solver are needed. The flexibility in time makes the GePUP method applicable to both low-Reynolds-number flow and high-Reynolds-number flow. The flexibility in space makes the GePUP method applicable to both rectangular boxes and irregular domains. The numerical results and elementary analysis show that the fourth-order GePUP method may be much more accurate and efficient than classical second-order projection methods by many orders of magnitude.

Keywords: incompressible Navier-Stokes equations, projection methods, fourth-order accuracy both in time and in space, no-slip boundary conditions, generic projection, pressure Poisson equation

PACS: 02.60.Cb, 47.10.ad, 47.11.Df **DOI:** 10.7498/aps.70.20210259

* Project supported by the Science Fund for Creative Research Groups of the National Natural Science Foundation of China (Grant No. 11621101) and the National Natural Science Foundation of China (Grant No. 11871429).

† Corresponding author. E-mail: qinghai@zju.edu.cn