

Stokes-Cahn-Hilliard方程的二阶并行解耦算法

刘 鑫

成都理工大学数学科学学院, 四川 成都

收稿日期: 2025年3月24日; 录用日期: 2025年4月17日; 发布日期: 2025年4月27日

摘 要

本文为Stokes-Cahn-Hilliard相场模型开发了一种具有二阶时间精度的时间并行解耦算法, 在时间上基于谱延迟校正方法和时间并行Parareal算法。此外, 我们使用稳定化方法处理Cahn-Hilliard方程, 运用了压力投影方法处理Stokes方程中压力 p 和速度 u 的耦合。相较于解耦的方案, 进一步提高了算法的计算效率。最后, 通过数值实验证明了它的稳定性和高效性。

关键词

Stokes-Cahn-Hilliard模型, 时间并行方法, 谱延迟校正方法, 有限元方法

Second-Order Parallel Decoupled Algorithms for the Stokes-Cahn-Hilliard Equation

Xin Liu

School of Mathematical Sciences, Chengdu University of Technology, Chengdu Sichuan

Received: Mar. 24th, 2025; accepted: Apr. 17th, 2025; published: Apr. 27th, 2025

Abstract

In this paper, we develop a time-parallel decoupled algorithm with second-order temporal accuracy for the Stokes-Cahn-Hilliard phase-field model, based in time on the spectral-deferred correction method and the time-parallel Parareal algorithm. In addition, we use a stabilisation method for the Cahn-Hilliard equation and employ a pressure projection method for the coupling of pressure p and velocity u in the Stokes equation. Compared with the decoupled scheme, the computational efficiency of the algorithm is further improved. Finally, numerical experiments demonstrate its stability and efficiency.

Keywords

Stokes-Cahn-Hilliard Model, Time Parallel Method, Spectral Deferred Correction Method, Finite Element Methods

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

流体、固体和气体混合物在多孔介质中的流动发生在各种物理、生物和工业过程中。因此，了解和准确预测多孔介质中的多相多组分流动对许多科学和工程应用都具有重大意义。相场方法是建模和模拟多相流问题的常用工具，相关概述见[1]-[4]。Stokes-Cahn-Hilliard (SCH)模型[5]由 Cahn-Hilliard 方程和不可压缩的 Stokes 方程构成，是相场模型中的一种，该模型描述了两相不可压缩不相容流体的混合物的相场演化过程。但是想要建立有效的数值格式来求解该耦合相场模型仍然面临刚性大和耦合项的问题。凸分裂方法和稳定线性隐式方法[6]是构造无条件能量稳定格式的两种常用方法。Eyre [7]推广的凸分裂格式是一种能保持能量耗散定律的数值格式。该格式隐式地处理化学势的凸部分和凹部分，并建立了唯一可解的和无条件能量稳定的格式。Kim 等[8]提出了具有对数自由能的 Cahn-Hilliard 方程的非线性凸分裂傅里叶谱格式，这是对多项式自由能泛函凸分裂思想的适当推广。不幸的是，凸分裂方法通常会导致非线性格式。同时，如何将算法推广到更高阶仍是一个问题。

谱延迟校正(SDC)方法是一种具有良好的稳定性和高精度的高阶数值方法，它在 2000 年由 Dutt 等人[9]首次提出。然而，SDC 方法由于本身的数值积分节点和迭代校正次数导致了较高的计算代价。为了降低计算成本，Minion [10]提出了一种半隐式含刚性和非刚性的常微分方程的谱延迟校正方法(SISDC)，该方法对非刚性项显式处理，刚性项隐式处理。在这个半隐式框架内，利用一阶欧拉法求解修正方程。假设数值积分是精确的，那么修正方程每迭代一次，数值解的精度就会提高一阶。之后，为了提高精度和迭代速度，郭瑞晗[11]开发了一类基于二阶时间积分法的半隐式 SDC 方法，每增加一次迭代，精度就会提高两阶。这为更快地构建高阶的数值方案提供了一种方法。

Parareal 方法作为时间并行计算领域[12]的重要突破，自 2001 年由 Lions 等人提出以来，受到了学术界和工业界的广泛关注。该方法巧妙地将时间区间剖分为若干个子区间，然后利用多重处理器，在每个时间子区间上并行求解，是一种在时间方向上并行计算常微分方程或时间离散的偏微分方程数值解的迭代方法。这种方法也被称作多重射击法，或者时间多重网格法，Gander 等人[13]曾针对该方法展开过探讨。自从 Parareal 方法问世，其再度激发了学界对于构建时间并行方法的研究热情[14]-[16]。在此背景下，一系列更适配高性能并行计算机的数值算法相继涌现并不断发展，进而在实际应用中诸如航空航天领域，加快了飞行器的设计和优化进程，降低研发成本等。此外，文献[17] [18]中提出了不同的改进算法来提高算法收敛阶。特别地，由于 Parareal 方法和 SDC 方法都是预测校正方法，Minion 等人[19]将两者结合起来。在文献[20]中给出该混合算法的详细分析，该融合算法的优势主要体现在两个方面：首先，将时间并行的 Parareal 算法与高阶 SDC 算法相结合，可以有效地平摊 SDC 算法在每个时间步上较高的计算成本；另一方面，相较于其它高阶算法，例如高阶龙格库塔方法等，选取高阶 SDC 算法作为 Parareal 算法中的高阶格式可以利用低阶数值方法迭代求解得到高阶精度，减少了 Parareal 算法中每次迭代的计算成本，提高了并行效率。因此，在本文中，我们基于谱延迟校正方法和时间并行 Parareal 算法，

提出了 SCH 模型的二阶时间并行算法, 该算法在提高计算效率和精度的同时, 保证离散算法的稳定性。

本文的主要内容安排如下。在第 2 节中, 首先介绍了 SCH 模型和 Parareal 时间并行方法。然后给出求解 SCH 系统的二阶 SDC 解耦算法, 然后基于 SDC 算法和 Parareal 算法构造了 SCH 模型的二阶时间并行解耦算法。在第 3 节中, 通过数值示例来验证了二阶时间并行解耦算法的准确性、稳定性和高效性。第 4 节做出总结。

2. 模型与算法

2.1. Stokes-Cahn-Hilliard 模型

两种不可压缩不相容流体的混合物的相场演化模型是由在时间区域 $(0, T)$ 和有界的 Lipschitz 空间域 $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) 的 Stokes-Cahn-Hilliard 方程所描述的:

$$\begin{cases} c_t + \nabla \cdot (cu) = M \Delta \omega, & (0, T) \times \Omega \\ \omega = \lambda (-\Delta c + f(c)), & (0, T) \times \Omega \\ u_t + \nabla p - \nu \Delta u = -c \nabla \omega, & (0, T) \times \Omega \\ \nabla \cdot u = 0, & (0, T) \times \Omega \end{cases} \quad (1)$$

其中, M 是与松弛时间尺度相关的迁移率常数, ν 是粘度系数, λ 与表面张力相对应, u 是速度场, p 是修正压力, ω 是化学势。相场函数 c 起着观察微观浓度的作用。比如在纯流体的区域内, 相场函数 c 可以达到数值 -1 和 1, 而在扩散界面内, $|c| < 1$ 。对于非线性项 $f(c) = F'(c)$, 其中 $F(c) = \frac{1}{4\eta^2}(c-1)^2$ 表示双阱势能, 用于惩罚偏离物理约束 $|c| \leq 1$ 的行为。方程(1)的前两项是 Cahn-Hilliard 方程, $\nabla \cdot (cu)$ 项模拟平流效应。方程(1)的后两项是不可压缩的 Stokes 方程, 其中的非线性项 $c \nabla \omega$ 模拟了表面张力效应。

下面将讨论模型(1)的边界条件和初始条件, 我们假设

$$(u(0), c(0)) = (u^0, c^0), \quad (2)$$

和

$$u|_{\partial\Omega} = 0, \partial_n c^{n+1}|_{\partial\Omega} = 0, \partial_n \omega^{n+1}|_{\partial\Omega} = 0. \quad (3)$$

此外, 若是假设该系统除重力外没有其他外力, 则 Stokes-Cahn-Hilliard 方程满足以下能量定律:

$$\frac{d}{dt} \left\{ W(\phi, \nabla \phi) + \int_{\Omega} \frac{1}{2} |u|^2 dx \right\} = - \int_{\Omega} M |\nabla \mu|^2 + \nu |\nabla u|^2 dx. \quad (4)$$

2.2. 时间并行 Parareal 算法

基于 Parareal 算法的理念, 当时间区间 $[0, T]$ 被分割为 N 个互不重叠的子区间时, 每个子区间将被分配给不同的处理器。为简化表述, 我们将这 N 个处理器记为 $P_1, \dots, P_N$ 。在 Parareal 时间并行算法[20]里, 会用到两个数值逼近算法, 分别记为传播函数 $\mathcal{G}$ 与 $\mathcal{F}$ 。为确保算法在并行计算时具备高效性, $\mathcal{G}$ 的计算成本必须低于 $\mathcal{F}$ 。另外, 在时间区间 $[t^n, t^{n+1}]$ 内, 若以 t^n 时刻的 $\hat{u}$ 作为初始值, 通过 $\mathcal{G}$ 和 $\mathcal{F}$ 计算得到 t^{n+1} 时刻的数值解, 则分别用 $\mathcal{G}(t^{n+1}, t^n, \hat{u})$ 与 $\mathcal{F}(t^{n+1}, t^n, \hat{u})$ 来表示。

在 Parareal 方法中, $\mathcal{G}$ 常被用来串行计算所有节点的临时逼近解, 即

$$u_1^{n+1} = \mathcal{G}(t^{n+1}, t^n, u_1^n), n = 0, 1, \dots, N-1.$$

只要每个处理器 P_{n+1} 获得初始值 u_1^n 和临时逼近解 u_1^{n+1} , 它们就能并行计算出更精确的近似解

$\mathcal{F}(t^{n+1}, t^n, u_1^n)$ 。然后使用如下的校正步串行得到最终的逼近解

$$u_2^{n+1} = \mathcal{G}(t^{n+1}, t^n, u_2^n) + \mathcal{F}(t^{n+1}, t^n, u_1^n) - \mathcal{G}(t^{n+1}, t^n, u_1^n). \quad (5)$$

2.3. SCH 方程的二阶算法

在本节中，我们首先给出了 SCH 方程的二阶解耦 SDC 算法。然后，基于 Parareal 并行算法，建立了一种求解 SCH 系统的二阶解耦时间并行算法。首先我们通过添加稳定项(7)得到 SCH 方程的一阶解耦算法。然后在一阶解耦算法的基础上通过谱延迟校正方法构造二阶解耦 SDC 算法。

算法 1. (SCH 方程的二阶 SDC 解耦算法)

1) (一阶解耦算法)给定初值 $(c^0, \omega^0, u^0, p^0)$ ，求解 $(c_1^{n+1}, \omega_1^{n+1}, u_1^{n+1}, p_1^{n+1}), n \geq 0$ 满足

$$\begin{cases} \frac{c_1^{n+1} - c_1^n}{\Delta t} + \nabla \cdot (c_1^n \mathbf{u}_1^n) - M \Delta \omega_1^{n+1} = 0, \\ \omega_1^{n+1} - \frac{\lambda}{\eta^2} (c_1^{n+1} - c_1^n) + \lambda (\Delta c_1^{n+1} - f(c_1^n)) = 0; \end{cases} \quad (6)$$

$$\mathbf{u}_{*1}^n = \mathbf{u}_1^n - \Delta t c_1^n \nabla \omega_1^{n+1}; \quad (7)$$

$$\frac{\tilde{\mathbf{u}}_1^{n+1} - \mathbf{u}_1^n}{\Delta t} = \nu \Delta \tilde{\mathbf{u}}_1^{n+1} - \nabla p_1^n - c_1^n \nabla \omega_1^{n+1}; \quad (8)$$

$$\begin{cases} -\Delta(p_1^{n+1} - p_1^n) = -\frac{1}{\Delta t} \nabla \cdot \tilde{\mathbf{u}}_1^{n+1}, & \mathbf{u}_1^{n+1} = \tilde{\mathbf{u}}_1^{n+1} - \Delta t \nabla (p_1^{n+1} - p_1^n), \\ \partial_n(p_1^{n+1} - p_1^n)|_{\partial\Omega} = 0, \end{cases} \quad (9)$$

2) (二阶解耦 SDC 算法)由一阶解耦算法求得 $(c_1^{n+1}, \omega_1^{n+1}, u_1^{n+1}, p_1^{n+1})$ ，求解 $(c_2^{n+1}, \omega_2^{n+1}, u_2^{n+1}, p_2^{n+1})$ 满足

$$\begin{cases} \frac{c_2^{n+1} - c_2^n}{\Delta t} + \nabla \cdot (c_2^n \mathbf{u}_2^n) - M \Delta \omega_2^{n+1} = \frac{1}{2} (\nabla \cdot (c_1^n \mathbf{u}_1^n) - \nabla \cdot (c_1^{n+1} \mathbf{u}_1^{n+1})) + \frac{1}{2} M (\Delta \omega_1^n - \Delta \omega_1^{n+1}), \\ \omega_2^{n+1} - \frac{\lambda}{\eta^2} (c_2^{n+1} - c_2^n) + \lambda (\Delta c_2^{n+1} - f(c_2^n)) \\ = \frac{1}{2} (\omega_1^n - \omega_1^{n+1}) + \frac{\lambda}{\eta^2} (c_1^{n+1} - c_1^n) + \frac{1}{2} \lambda (\Delta c_1^n - \Delta c_1^{n+1}) - \frac{1}{2} \lambda (f(c_1^n) - f(c_1^{n+1})), \end{cases} \quad (10)$$

$$\begin{cases} \frac{\tilde{\mathbf{u}}_2^{n+1} - \mathbf{u}_2^n}{\Delta t} - \nu \Delta \tilde{\mathbf{u}}_2^{n+1} = \frac{1}{2} \nu (\Delta \tilde{\mathbf{u}}_1^n - \Delta \tilde{\mathbf{u}}_1^{n+1}) - \nabla p_2^n \\ + \frac{1}{2} \nabla (p_1^n - \nabla p_1^{n+1}) - \frac{1}{2} (c_2^n \nabla \omega_2^n + c_2^{n+1} \nabla \omega_2^{n+1}), \end{cases} \quad (11)$$

$$\begin{cases} -\Delta(p_2^{n+1} - p_2^n - p_1^{n+1} + p_1^n) = -\frac{1}{\Delta t} \nabla \cdot \tilde{\mathbf{u}}_2^{n+1}, \\ \partial_n(p_2^{n+1} - p_2^n - p_1^{n+1} + p_1^n)|_{\partial\Omega} = 0, \end{cases} \quad (12)$$

$$\mathbf{u}_2^{n+1} = \tilde{\mathbf{u}}_2^{n+1} - \Delta t \nabla (p_2^{n+1} - p_2^n - p_1^{n+1} + p_1^n).$$

为了进一步提高数值求解 SCH 系统的计算效率，我们将基于 SDC 方法和并行算法，提出一种具有时间二阶精度且能量稳定的时间并行算法。该算法是对上述解耦算法的进一步优化，相较于一般的 SDC 算法，降低了计算规模，大大提高了计算效率。在本文中，我们采用一阶解耦算法作为传播函数 $\mathcal{G}$ ，并采用类似(10)~(12)式的 SDC 方法作为传播函数 $\mathcal{F}$ ，在确保精度和能量稳定性的前提下，利用时间步并

行技巧提高了计算效率。

算法 2. (SCH 方程的二阶解耦并行算法)

1) 串行求解 $\mathcal{G}(t^{n+1}, t^n, c_1^n, \omega_1^n, u_1^n, p_1^n)$, $n=0, \dots, N-1$, 即在时间方向依次计算(10)式得到 $(c_1^{n+1}, \omega_1^{n+1}, u_1^{n+1}, p_1^{n+1})$ 。

2) 利用二阶解耦 SDC 算法的校正步并行求解 $\mathcal{F}(t^{n+1}, t^n, c_1^n, \omega_1^n, u_1^n, p_1^n)$, $n=0, \dots, N-1$, 以第 $n+1$ 个处理器 P_{n+1} 为例: 求解 $(c_F^{n+1}, \omega_F^{n+1}, u_F^{n+1}, p_F^{n+1})$ 满足

$$\begin{cases} \frac{c_F^{n+1} - c_1^n}{\Delta t} + \nabla \cdot (c_1^n \mathbf{u}_1^n) - M \Delta \omega_F^{n+1} = \frac{1}{2} (\nabla \cdot (c_1^n \mathbf{u}_1^n) - \nabla \cdot (c_1^{n+1} \mathbf{u}_1^{n+1})) + \frac{1}{2} M (\Delta \omega_1^n - \Delta \omega_1^{n+1}), \\ \omega_F^{n+1} - \frac{\lambda}{\eta^2} (c_F^{n+1} - c_1^n) + \lambda (\Delta c_F^{n+1} - f(c_1^n)) \\ = \frac{1}{2} (\omega_1^n - \omega_1^{n+1}) + \frac{\lambda}{\eta^2} (c_1^{n+1} - c_1^n) + \frac{1}{2} \lambda (\Delta c_1^n - \Delta c_1^{n+1}) - \frac{1}{2} \lambda (f(c_1^n) - f(c_1^{n+1})), \end{cases} \quad (13)$$

$$\mathbf{u}_1^n = \mathbf{u}_1^n - \Delta t c_1^n \nabla \omega_1^{n+1}; \quad (14)$$

$$\frac{\tilde{\mathbf{u}}_F^{n+1} - \mathbf{u}_1^n}{\Delta t} - \nu \Delta \tilde{\mathbf{u}}_F^{n+1} = \frac{1}{2} \nu (\Delta \tilde{\mathbf{u}}_1^n - \Delta \tilde{\mathbf{u}}_1^{n+1}) - \frac{1}{2} (\nabla p_1^n + \nabla p_1^{n+1}) - c_1^n \nabla \omega_1^{n+1}, \quad (15)$$

$$\begin{cases} -\Delta (p_F^{n+1} - p_1^n - p_1^{n+1} + p_1^n) = -\frac{1}{\Delta t} \nabla \cdot \tilde{\mathbf{u}}_F^{n+1}, \\ \partial_n (p_F^{n+1} - p_1^n - p_1^{n+1} + p_1^n) \Big|_{\partial \Omega} = 0, \end{cases} \quad (16)$$

$$\mathbf{u}_F^{n+1} = \tilde{\mathbf{u}}_F^{n+1} - \Delta t \nabla (p_F^{n+1} - p_1^n - p_1^{n+1} + p_1^n)$$

3) 串行求解 $\mathcal{G}(t^{n+1}, t^n, c_2^n, \omega_2^n, u_2^n, p_2^n)$, $n=0, \dots, N-1$, 即在时间方向依次求解 $(c_G^{n+1}, \omega_G^{n+1}, u_G^{n+1}, p_G^{n+1})$, 满足

$$\begin{cases} \frac{c_G^{n+1} - c_2^n}{\Delta t} + \nabla \cdot (c_2^n \mathbf{u}_2^n) - M \Delta \omega_G^{n+1} = 0, \\ \omega_G^{n+1} - \frac{\lambda}{\eta^2} (c_G^{n+1} - c_2^n) + \lambda (\Delta c_G^{n+1} - f(c_2^n)) = 0; \end{cases} \quad (17)$$

$$\begin{cases} c_{2,h}^{n+1} = c_G^{n+1} + c_F^{n+1} - c_{1,h}^{n+1}, \\ \omega_{2,h}^{n+1} = \omega_G^{n+1} + \omega_F^{n+1} - \omega_{1,h}^{n+1}; \end{cases} \quad (18)$$

$$\frac{\tilde{\mathbf{u}}_G^{n+1} - \mathbf{u}_2^n}{\Delta t} = \nu \Delta \tilde{\mathbf{u}}_G^{n+1} - \nabla p_2^n - \frac{1}{2} (c_2^n \nabla \omega_2^n + c_2^{n+1} \nabla \omega_2^{n+1}); \quad (19)$$

$$\begin{cases} -\Delta (p_G^{n+1} - p_2^n) = -\frac{1}{\Delta t} \nabla \cdot \tilde{\mathbf{u}}_G^{n+1}, \\ \partial_n (p_G^{n+1} - p_2^n) \Big|_{\partial \Omega} = 0; \end{cases} \quad (20)$$

$$\mathbf{u}_G^{n+1} = \tilde{\mathbf{u}}_G^{n+1} - \Delta t \nabla (p_G^{n+1} - p_2^n); \quad (21)$$

$$\begin{cases} \tilde{\mathbf{u}}_{2,h}^{n+1} = \tilde{\mathbf{u}}_G^{n+1} + \tilde{\mathbf{u}}_F^{n+1} - \tilde{\mathbf{u}}_{1,h}^{n+1}, \\ \mathbf{u}_{2,h}^{n+1} = \mathbf{u}_G^{n+1} + \mathbf{u}_F^{n+1} - \mathbf{u}_{1,h}^{n+1}, \\ p_{2,h}^{n+1} = p_G^{n+1} + p_F^{n+1} - p_{1,h}^{n+1}. \end{cases} \quad (22)$$

随后将介绍数值实验, 以证明二阶解耦并行算法在求解 SCH 方程时确实具有高阶精确性、能

量稳定性和高效性。

3. 数值实验

在这一部分，我们将借助数值实验来检验针对 SCH 方程的二阶时间并行解耦算法的有效性。所有数值运算均基于 MATLAB 软件开展。就空间离散化而言，对于 Stokes 方程，我们运用 Taylor-Hood (P2-P1)有限元进行近似处理；对于 Cahn-Hilliard 方程，则采用二次多项式有限元(P2)来加以近似。

3.1. 收敛性实验

我们通过数值实验验证算法 2 的时间收敛阶。我们考虑在计算区域为 $\Omega = [0,1] \times [0,1]$ ，最终时间为 $T=0.1$ ，物理参数为 $\eta = 0.04, \lambda = 0.04^2, M = 0.1, \nu = \frac{1}{100}$ 。初始条件为

$$\begin{aligned}\phi_0 &= 0.24 \cos(2\pi x) \cos(2\pi y) + 0.4 \cos(\pi x) \cos(3\pi y), \\ u_0 &= (-\sin^2(\pi x) \sin(2\pi y), \sin^2(\pi y) \sin(2\pi x)).\end{aligned}$$

在文献[21]中，给出了以下计算收敛阶的方法：假定

$$v_h^{\Delta t}(x, t_m) \approx v(x, t_m) + C_1(x, t_m) \Delta t^\gamma + C_2(x, t_m) h^\mu,$$

定义

$$\rho_{v, \Delta t, i} = \frac{\left| v_h^{\Delta t}(x, t_m) - v_h^{\frac{\Delta t}{2}}(x, t_m) \right|_i}{\left| v_h^{\frac{\Delta t}{2}}(x, t_m) - v_h^{\frac{\Delta t}{4}}(x, t_m) \right|_i} \approx \frac{4^\gamma - 2^\gamma}{2^\gamma - 1},$$

Table 1. The temporal convergence order of Algorithm 2 with the fixed grid size $h = \frac{1}{32}$, the final moment $T = 0.1$ and different time steps $\Delta t = \frac{T}{80}, \frac{T}{160}, \frac{T}{320}, \frac{T}{640}$

表 1. 算法 2 的时间收敛阶，其中固定网格大小为 $h = \frac{1}{32}$ ，最终时刻 $T = 0.1$ ，并选取不同的时间步长

$$\Delta t = \frac{T}{80}, \frac{T}{160}, \frac{T}{320}, \frac{T}{640}$$

Δt	$\ \delta_{u_1}^{\Delta t}\ $	order	$\ \delta_{u_1}^{\Delta t}\ _{H^1(\Omega)}$	order	$\ \delta_{u_2}^{\Delta t}\ $	order	$\ \delta_{u_1}^{\Delta t}\ _{H^1(\Omega)}$	order
$T/80$	2.9932e-05	1.9036	4.8890e-04	1.8604	3.0472e-05	1.9030	4.4057e-04	1.8399
$T/160$	7.9998e-06	1.9518	1.3464e-04	1.9275	8.1478e-06	1.9518	1.2307e-04	1.9175
$T/320$	2.0678e-06	1.9760	3.5393e-05	1.9631	2.1062e-06	1.9760	3.2578e-05	1.9581
$T/640$	5.2565e-07		9.0774e-06		5.3539e-07		8.3846e-06	
Δt	$\ \delta_\phi^{\Delta t}\ $	order	$\ \delta_\phi^{\Delta t}\ _{H^1(\Omega)}$	order	$\ \delta_p^{\Delta t}\ $	order		
$T/80$	7.0617e-04	1.8723	1.4863e-02	1.8732	1.8003e-04	1.8902		
$T/160$	1.9288e-04	1.9349	4.0571e-03	1.9357	4.8567e-05	1.9436		
$T/320$	5.0447e-05	1.9671	1.0605e-03	1.9676	1.2626e-05	1.9715		
$T/640$	1.2903e-05		2.7114e-04		3.2194e-06			

$$\rho_{v,h,i} = \frac{\left| v_h^{\Delta t}(x, t_m) - v_{\frac{h}{2}}^{\Delta t}(x, t_m) \right|_i}{\left| v_{\frac{h}{2}}^{\Delta t}(x, t_m) - v_{\frac{h}{4}}^{\Delta t}(x, t_m) \right|_i} \approx \frac{4^\mu - 2^\mu}{2^\mu - 1}.$$

这里, v 可以为 u 、 p 或 φ , 且 $i=0,1$ 。如果 $\log_2 \rho_{v,\Delta t,0}$, $\log_2 \rho_{v,h,0} \approx 2$, 则表明 L^2 误差的空间收敛阶和时间收敛阶分别为 $O(\Delta t^2)$ 和 $O(h^2)$ 。同样, 通过计算 $\log_2 \rho_{v,\Delta t,1}$ 和 $\log_2 \rho_{v,h,1}$ 可以得到 H^1 误差的空间和时间收敛阶。为方便记述, 我们将定义 $\delta_v^{\Delta t} = v_h^{\Delta t} - v_{\frac{h}{2}}^{\Delta t}$ 和 $\delta_v^h = v_h^{\Delta t} - v_{\frac{h}{2}}^{\Delta t}$ 。

表 1 列出了算法 2 的时间收敛阶。我们固定空间步长为 $h = \frac{1}{32}$, 并选择不同的时间步长

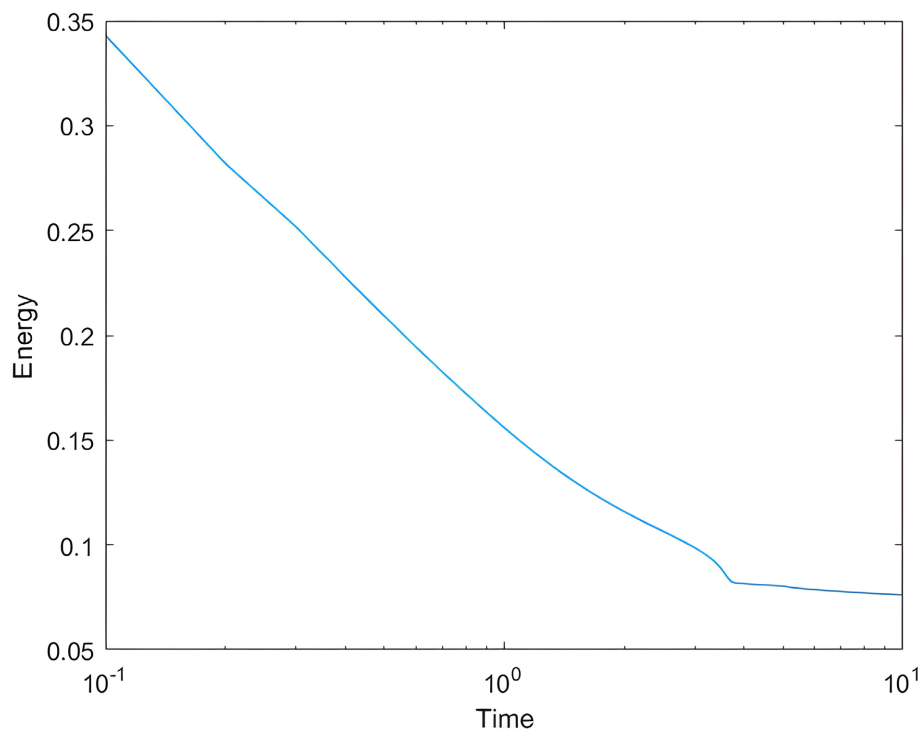
$\Delta t = \frac{T}{80}, \frac{T}{160}, \frac{T}{320}, \frac{T}{640}$ 。表 1 中的结果显示, 对于 c, p 和 u , 算法 2 的时间收敛阶在 L^2 范数下和 H^1 范数下均达到了二阶精度。

3.2. 稳定性实验

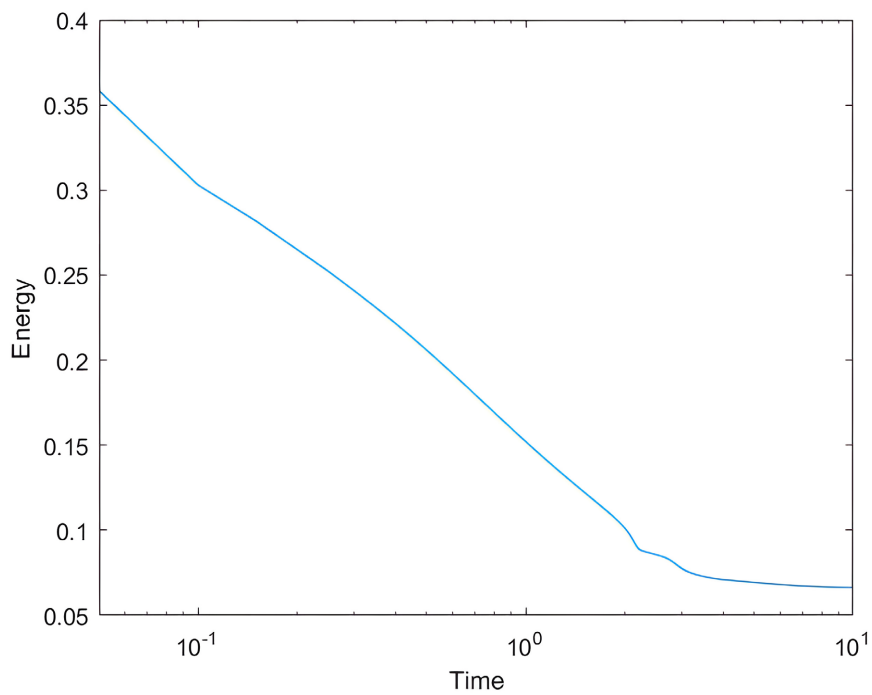
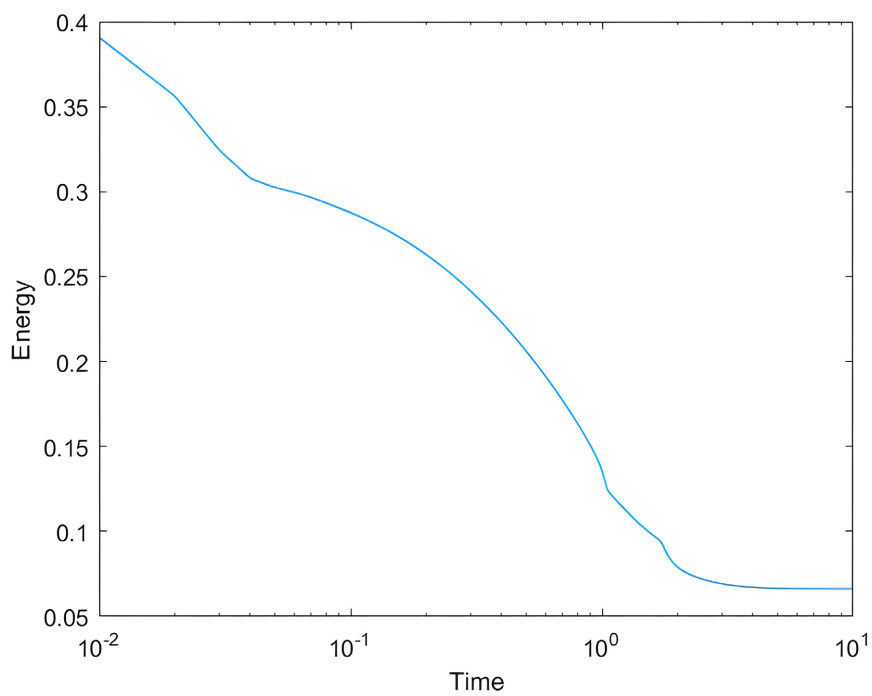
在这一部分中, 我们将验证算法 2 的能量稳定性, 首先, 先定义离散时间 $t=n\Delta t$ 的离散能量函数

$$E_t = \lambda(F(\phi^n), 1) + \frac{\lambda}{2} \|\nabla \phi^n\|^2 + \frac{1}{2} \|u^n\|^2.$$

在本实验中, 取空间步长 $h = \frac{1}{64}$, 其他参数与上述收敛性实验中的参数相同。图 1 给出了算法 2 在 $\Delta t = 0.1, 0.05, 0.01$ 三个时间步长下的能量变化情况。图 1 显示, 算法 2 在不同的时间步长下, 离散能量函数 E_t 随时间是递减的。因此, 我们可以得到, 算法 2 具有良好的稳定性。



(A) $\Delta t = 0.1$

(B) $\Delta t = 0.05$ (C) $\Delta t = 0.01$ **Figure 1.** Evolution of discrete energy E_t for Algorithm 2 are shown at $\Delta t = 0.1, 0.05, 0.01$ **图 1.** 算法 2 在 $\Delta t = 0.1, 0.05, 0.01$ 时, 离散能量 E_t 变化情况

3.3. 运行时间比较

为了验证算法 2 和算法 3 的并行有效性, 我们比较了时间并行算法与串行二阶 SDC 算法, 即算法 1

的计算时间。在本实验中，计算区域和参数选择与实验 3.1 中相同。表 2 和表 3 中的数值结果表明，算法 2 和 3 的运算时间相近，且与串行算法 1 相比，都提升了运行效率，但通过实验 2 表明了算法 2 的稳定性更好。因此，在同等条件下，算法 2 在稳定性和计算效率均具有优势。

Table 2. The CPU time of the time-parallel and serial decoupling schemes with the fixed mesh $h = \frac{1}{32}$

表 2. 时间并行方案和串行解耦方案在固定网格 $h = \frac{1}{32}$ 时的 CPU 运行时间

数值算法	CPU (s)			
	$\Delta t = 1/40$	$\Delta t = 1/80$	$\Delta t = 1/160$	$\Delta t = 1/320$
二阶解耦算法	11.598	23.5438	45.2505	89.1838
二阶解耦并行算法	6.2058	10.9925	22.3059	44.272
加速比	1.9	2.1	2.0	2.0

Table 3. The CPU time of the time-parallel and serial decoupling schemes with the fixed mesh $h = \frac{1}{64}$

表 3. 时间并行方案和串行解耦方案在固定网格 $h = \frac{1}{64}$ 时的 CPU 运行时间

数值算法	CPU (s)			
	$\Delta t = 1/40$	$\Delta t = 1/80$	$\Delta t = 1/160$	$\Delta t = 1/320$
二阶解耦算法	62.912	121.499	229.6978	461.2383
二阶解耦并行算法	28.3711	56.9073	114.9398	234.079
加速比	2.2	2.1	2.0	2.0

4. 结论

为提高数值求解具有匹配密度的 Stokes-Cahn-Hilliard 相场模型的计算效率，本文基于半隐的 SDC 算法和 Parareal 算法构造了一个具有时间二阶精度的时间并行算法。我们在数值上验证了算法的收敛性和稳定性，同时通过不同算法的比较说明了我们的时间并行算法的高效性。

参考文献

[1] Rayleigh, L. (1892) XX. *On the Theory of Surface Forces—II. Compressible Fluids. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **33**, 209-220. <https://doi.org/10.1080/14786449208621456>

[2] Shen, J. and Yang, X. (2015) Decoupled, Energy Stable Schemes for Phase-Field Models of Two-Phase Incompressible Flows. *SIAM Journal on Numerical Analysis*, **53**, 279-296. <https://doi.org/10.1137/140971154>

[3] Shen, J. and Yang, X. (2010) Energy Stable Schemes for Cahn-Hilliard Phase-Field Model of Two-Phase Incompressible Flows. *Chinese Annals of Mathematics, Series B*, **31**, 743-758. <https://doi.org/10.1007/s11401-010-0599-y>

[4] Shen, J. and Yang, X. (2010) A Phase-Field Model and Its Numerical Approximation for Two-Phase Incompressible Flows with Different Densities and Viscosities. *SIAM Journal on Scientific Computing*, **32**, 1159-1179. <https://doi.org/10.1137/09075860x>

[5] Bañas, E. and Mahato, H.S. (2017) Homogenization of Evolutionary Stokes-Cahn-Hilliard Equations for Two-Phase Porous Media Flow. *Asymptotic Analysis*, **105**, 77-95. <https://doi.org/10.3233/asy-171436>

[6] Shen, J. and Yang, X. (2010) Numerical Approximations of Allen-Cahn and Cahn-Hilliard Equations. *Discrete & Continuous Dynamical Systems—A*, **28**, 1669-1691. <https://doi.org/10.3934/dcds.2010.28.1669>

[7] Eyre, D.J. (1998) Unconditionally Gradient Stable Time Marching the Cahn-Hilliard Equation. *MRS Online Proceedings*

-
- Library, **529**, 39-46. <https://doi.org/10.1557/proc-529-39>
- [8] Kim, J. and Lee, H. (2019) A Nonlinear Convex Splitting Fourier Spectral Scheme for the Cahn-Hilliard Equation with a Logarithmic Free Energy. *Bulletin of the Korean Mathematical Society*, **56**, 265-276.
 - [9] Dutt, A., Greengard, L. and Rokhlin, V. (2000) Spectral Deferred Correction Methods for Ordinary Differential Equations. *Bit Numerical Mathematics*, **40**, 241-266. <https://doi.org/10.1023/a:1022338906936>
 - [10] Minion, M.L. (2003) Semi-Implicit Spectral Deferred Correction Methods for Ordinary Differential Equations. *Communications in Mathematical Sciences*, **1**, 471-500. <https://doi.org/10.4310/cms.2003.v1.n3.a6>
 - [11] Ruihan Guo, R.G. and Yan Xu, Y.X. (2024) Semi-Implicit Spectral Deferred Correction Methods Based on Second-Order Time Integration Schemes for Nonlinear PDEs. *Journal of Computational Mathematics*, **42**, 111-133.
 - [12] Gander, M.J. (2015) 50 Years of Time Parallel Time Integration. *Multiple Shooting and Time Domain Decomposition Methods*, Heidelberg, 6-8 May 2013, 69-113. https://doi.org/10.1007/978-3-319-23321-5_3
 - [13] Gander, M.J. and Vandewalle, S. (2007) Analysis of the Parareal Time-Parallel Time-Integration Method. *SIAM Journal on Scientific Computing*, **29**, 556-578. <https://doi.org/10.1137/05064607x>
 - [14] Maday, Y. and Mula, O. (2020) An Adaptive Parareal Algorithm. *Journal of Computational and Applied Mathematics*, **377**, Article 112915. <https://doi.org/10.1016/j.cam.2020.112915>
 - [15] Wu, S. and Zhou, T. (2017) Fast Parareal Iterations for Fractional Diffusion Equations. *Journal of Computational Physics*, **329**, 210-226. <https://doi.org/10.1016/j.jcp.2016.10.046>
 - [16] Pentland, K., Tamborrino, M., Samaddar, D. and Appel, L.C. (2022) Stochastic Parareal: An Application of Probabilistic Methods to Time-Parallelization. *SIAM Journal on Scientific Computing*, **45**, S82-S102. <https://doi.org/10.1137/21m1414231>
 - [17] Bu, S. and Lee, J. (2014) An Enhanced Parareal Algorithm Based on the Deferred Correction Methods for a Stiff System. *Journal of Computational and Applied Mathematics*, **255**, 297-305. <https://doi.org/10.1016/j.cam.2013.05.001>
 - [18] Gander, M. and Petcu, M. (2008) Analysis of a Krylov Subspace Enhanced Parareal Algorithm for Linear Problems. *ESAIM: Proceedings*, **25**, 114-129. <https://doi.org/10.1051/proc:082508>
 - [19] Minion, M.L. and Williams, S.A. (2008) Parareal and Spectral Deferred Corrections. *NUMERICAL ANALYSIS AND APPLIED MATHEMATICS: International Conference on Numerical Analysis and Applied Mathematics 2008*, Kos, 16-20 September 2008, 388-391. <https://doi.org/10.1063/1.2990941>
 - [20] Minion, M. (2010) A Hybrid Parareal Spectral Deferred Corrections Method. *Communications in Applied Mathematics and Computational Science*, **5**, 265-301. <https://doi.org/10.2140/camcos.2010.5.265>
 - [21] Mu, M. and Zhu, X. (2009) Decoupled Schemes for a Non-Stationary Mixed Stokes-Darcy Model. *Mathematics of Computation*, **79**, 707-731. <https://doi.org/10.1090/s0025-5718-09-02302-3>