

# 变分数阶Allen-Cahn方程的自适应指数时间差分方法

陈心悦, 梁霄\*

湖北文理学院数学与统计学院, 湖北 襄阳

收稿日期: 2026年4月15日; 录用日期: 2026年5月8日; 发布日期: 2026年5月19日

## 摘要

本文提出变分数阶Allen-Cahn方程的自适应指数时间差分方法, 在空间方向上采用有限差分法和分步傅里叶谱方法进行离散, 在时间方向上采用自适应控制与指数时间差分结合的方法, 构建数值格式并通过数值实验验证在计算精度、效率和稳定性方面的优势。

## 关键词

变分数阶方程, Allen-Cahn方程, 指数时间差分, 自适应方法, 误差估计

# Adaptive Exponential Time Discretization Method for the Variable-Order Fractional Allen-Cahn Equation

Xinyue Chen, Xiao Liang\*

College of Mathematics and Statistics, Hubei University of Arts and Sciences, Xiangyang Hubei

Received: April 15, 2026; accepted: May 8, 2026; published: May 19, 2026

## Abstract

This paper proposes an adaptive exponential time differencing method for the variable-order fractional Allen-Cahn equation. Finite difference and split-step Fourier spectral methods are used for spatial discretization, while an adaptive control combined with the exponential time differencing is employed for time discretization. The numerical scheme is constructed and its advantages in computational accuracy,

\*通讯作者。

efficiency, and stability are verified through numerical experiments.

## Keywords

Variable-Order Fractional Equation, Allen-Cahn Equation, Exponential Time Differencing, Adaptive Method, Error Estimates

Copyright © 2026 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

随着整数阶微积分理论体系日渐成熟, 分数阶微积分理论也逐步形成, 至 1993 年变分数阶微积分理论被正式提出并成为复杂系统建模的新数学工具[1]。变分数阶微分能够准确反映复杂系统的变记忆特性, 在处理非平稳和非线性系统时具有独特的优势[2]。变分数阶 Allen-Cahn 方程通过引入分数阶导数来描述具有记忆效应、非局部效应和反常扩散的过程, 其一般形式为:

$$\frac{\partial u}{\partial t} = L(t)u + N(u), \quad (1)$$

其中  $L(t)u = -(-\Delta)^{\alpha(t)/2}u$  为线性算子,  $\alpha(t)$  为时变阶数,  $N(u)$  为非线性项。

变分数阶 Allen-Cahn 方程(1)的非局部性和非线性项导致解析解难以获取, 数值解法成为研究重点[3]。本文采用谱方法(Spectral Methods, SM)和有限差分法(Finite Difference Method, FDM)来进行求解, 谱方法精度高但适用问题类型较窄, 有限差分法是最常用的且需要对分数阶导数设计高精度的离散格式[4]-[8]。本文构建自适应指数时间差分(ETD)方法求解变分数阶 Allen-Cahn 方程框架, 建立数值分析体系并设计系统化的数值实验方案。

## 2. 空间离散

求解偏微分方程数值的步骤之一是对连续的空间域进行离散化处理, 将无限维的问题转化为有限维的代数方程组。本文采用有限差分法和分步傅里叶谱方法, 考虑一维空间域  $\Omega = [0, 1]$  和二维矩形域  $\Omega = [0, 1] \times [0, 1]$ 。

### 2.1. 有限差分法

有限差分法是用离散网格点上函数值的线性组合来逼近偏微分方程(PDE)中的空间导数。一维情形下将域均匀划分为  $n$  个步长为  $h = \frac{1}{n-1}$  的均匀节点, 在网格点  $x_i$  处分数阶拉普拉斯算子  $(-\Delta)^{\alpha/2}$  基于 Grünwald-Letnikov 的离散定义为:

$$(-\Delta)^{\alpha/2} u(x_i, t) \approx \frac{1}{h^\alpha} \sum_{k=-M}^M g_k^{(\alpha)} u(x_{i+k}, t)$$

其中  $M$  为截断长度, 分数阶中心差分权重系数  $g_k^{(\alpha)}$  定义为:

$$g_k^{(\alpha)} = (-1)^k \frac{\Gamma(\alpha+1)}{\Gamma\left(\frac{\alpha}{2}-k-1\right)\Gamma\left(\frac{\alpha}{2}+k+1\right)}, \quad k = -M, \dots, M$$

其中  $\Gamma$  为 Gamma 函数, 离散化后分数阶拉普拉斯算子转化为带状矩阵。二维情形下将域沿  $x$  方向划分为  $n_x$  个步长为  $h_x = \frac{1}{n_x - 1}$  的等距节点, 沿  $y$  方向划分  $n_y$  个步长为  $h_y = \frac{1}{n_y - 1}$  的等距节点, 在节点  $(x_i, y_j)$  处分数阶拉普拉斯算子  $(-\Delta)^{\alpha/2}$  离散定义为:

$$(-\Delta)^{\alpha/2} u(x_i, y_j, t) \approx \sum_{k=-M_x}^{M_x} \sum_{l=-M_y}^{M_y} g_k^{(\alpha)} g_l^{(\alpha)} u_{i+k, j+l}$$

分数阶中心差分权重系数定义为:

$$g_k^{(\alpha)} = (-1)^k \frac{\Gamma(\alpha + 1)}{\Gamma\left(\frac{\alpha}{2} - k - 1\right)\Gamma\left(\frac{\alpha}{2} + k + 1\right)}, \quad k = -M, \dots, M$$

$$g_l^{(\alpha)} = (-1)^l \frac{\Gamma(\alpha + 1)}{\Gamma\left(\frac{\alpha}{2} - l - 1\right)\Gamma\left(\frac{\alpha}{2} + l + 1\right)}, \quad l = -M, \dots, M$$

### 2.2. 分步傅里叶谱方法

分步傅里叶谱方法是基于快速傅里叶变换(FFT)且用于处理具有周期性边界条件或规则区域问题的高精度数值方法。一维情形下解函数  $u(x, t)$  在网格点上的近似值记为  $u_j(t) \approx u(x_j, t)$ , 离散傅里叶变换定义为:

$$\hat{u}_k(t) = \mathcal{F}[u]_k = \sum_{j=0}^{n-1} u_j(t) \cdot e^{-i\xi_k x_j} = \sum_{j=0}^{n-1} u_j(t) \cdot e^{-i2\pi k j/n},$$

其中  $\xi_k = \frac{2\pi k}{L}$ ,  $k = -\frac{n}{2}, -\frac{n}{2} + 1, \dots, \frac{n}{2} - 1$ 。二维情形下解函数  $u(x, y, t)$  的离散傅里叶变换为:

$$\hat{u}_{pq}(t) = \mathcal{F}[u]_{pq} = \sum_{i=0}^{n_x-1} \sum_{j=0}^{n_y-1} u_{ij}(t) \cdot e^{-i(\xi_p x_i + \eta_q y_j)},$$

其中  $\xi_p = \frac{2\pi p}{L_x}$ ,  $\eta_q = \frac{2\pi q}{L_y}$ ,  $p = -\frac{n_x}{2}, \dots, \frac{n_x}{2} - 1$ ,  $q = -\frac{n_y}{2}, \dots, \frac{n_y}{2} - 1$ 。

对于任意函数  $u(\vec{x}, t)$ , 其分数阶拉普拉斯算子在谱空间中的表示为:

$$\mathcal{F}\left[(-\Delta)^{\alpha/2} u\right](\vec{k}, t) = |\vec{k}|^\alpha \hat{u}(\vec{k}, t)$$

### 3. 自适应指数时间差分方法

指数时间差分(ETD)方法适合用来求解具有刚性问题的分数阶微分方程, 本文主要介绍四阶指数时间差分方法(ETDRK4)。Cox 和 Matthews 提出的 ETDRK4 方法具有四阶精度且可以通过多个中间阶段来逼近非线性项[9]:

$$\text{阶段 1: } a_n = e^{L_n \tau_n / 2} u_n + \frac{\tau_n}{2} \phi_1(L_n \tau_n / 2) \mathcal{N}(u_n),$$

$$\text{阶段 2: } b_n = e^{L_n \tau_n / 2} u_n + \frac{\tau_n}{2} \phi_1(L_n \tau_n / 2) \mathcal{N}(a_n),$$

$$\text{阶段 3: } c_n = e^{L_n \tau_n / 2} a_n + \frac{\tau_n}{2} \phi_1(L_n \tau_n / 2) [2\mathcal{N}(b_n) - \mathcal{N}(u_n)],$$

$$\text{最终步: } u_{n+1} = e^{L_n \tau_n} u_n + \tau_n \left[ \phi_1(L_n \tau_n) \mathcal{N}(u_n) + 2\phi_2(L_n \tau_n) (\mathcal{N}(a_n) + \mathcal{N}(b_n)) + \phi_3(L_n \tau_n) \mathcal{N}(c_n) \right],$$

其中  $\phi_3$  函数定义为:  $\phi_3(z) = \frac{e^z - I - z - z^2/2}{z^3}$ ,  $\phi_3(0) = \frac{1}{6} I$ 。

### 3.1. Padé 近似

ETD 方法处理高维问题时, 直接计算矩阵指数函数  $e^{L\tau_n}$  和相关  $\phi$  函数无法很好地保持稳定性和高效性, 本文采用 Padé 近似提高效率, Padé 近似通过有理分式(即两个多项式的比值)来逼近这些函数。

函数的  $[p/q]$  阶 Padé 近似定义为:  $R_{[p/q]}(z) = \frac{P_p(z)}{Q_q(z)}$ , 其中  $P_p(z)$  和  $Q_q(z)$  分别是阶数为  $p$  和  $q$  的多项式, 本文主要运用  $[2/2]$  阶 Padé 近似:

$$R_{[2/2]}(z) = \frac{1 + z/2 + z^2/12}{1 - z/2 + z^2/12}$$

### 3.2. 自适应 ETD 算法

自适应算法的核心是精确估计局部截断误差(LTE)并调整步长, 本文采用 Richardson 外推法进行误差估计。Richardson 外推法通过比较不同步长的计算结果来估计误差, 设步长  $\tau_n$  时的解为  $u_{n+1}^{(\tau)}$ , 步长  $\tau_n/2$  经两次半步计算的解为  $u_{n+1}^{(\tau/2)}$ , 局部误差估计为:

$$\text{err}_n = \left\| u_{n+1}^{(\tau)} - u_{n+1}^{(\tau/2)} \right\|_{\infty}$$

本文采用自适应步长控制策略确保自适应 ETD 方法在计算效率和精度间取得平衡, 步长调整公式为:

$$\tau_{n+1} = \rho \cdot \tau_n \cdot \min \left( f_{\max}, \max \left( f_{\min}, \gamma \left( \frac{\text{tol}}{\text{err}_n} \right)^{1/(p+1)} \right) \right),$$

其中 tol 为用户定义的容差,  $\rho \approx 0.9$  为安全因子,  $\gamma \approx 0.8$  为阻尼因子,  $f_{\min} = 0.5$  和  $f_{\max} = 2.0$  为步长变化限幅,  $p$  为格式阶数。

### 3.3. 与现有方法的对比

本节从时间离散格式、步长策略、计算成本和适用问题类型这四个维度与标准的四阶显式龙格-库塔方法(经典 RK4)和固定步长的四阶指数时间差分方法(固定步长 ETD RK4)进行对比。

**Table 1.** Comparison of characteristics of discrete methods at different times

**表 1.** 不同时间离散方法特性对比

| 方法           | 时间离散格式               | 步长策略                | 计算成本                           | 适用问题类型                                |
|--------------|----------------------|---------------------|--------------------------------|---------------------------------------|
| 经典 RK4       | 基于右端项的四阶显式格式         | 固定步长                | 单步成本低, 但对刚性系统需极小步长, 总步数多, 效率低  | 非刚性或轻度刚性问题                            |
| 固定步长 ETD RK4 | 对线性部分进行精确积分的四阶指数积分格式 | 固定步长                | 单步成本高, 固定步长可能导致总计算量冗余          | 刚性/非线性问题 (如分数阶方程)                     |
| 自适应 ETD RK4  | 同固定步长 ETD RK4        | 自适应步长, 基于局部误差估计动态调整 | 单步成本略高, 但总步数显著减少, 在保证精度下整体效率更高 | 刚性/非线性问题, 并适用于刚度变化或解行为剧烈变化问题(如变分数阶方程) |

从表 1 可以看出, 用经典 RK4 解强刚性方程时需要使用极小的固定步长来维持数值稳定性, 导致在解平滑的区域也需要大量的计算, 而固定步长 ETD RK4 的时间离散格式使其在求解刚性方程时稳定性优

于经典 RK4。然而, 固定步长 ETDRK4 的固定步长策略无法根据解的变化来简化计算, 故选择固定步长 ETDRK4 来解 Allen-Cahn 方程也并非最优选择, 故本文提出自适应 ETDRK4 方法。

自适应 ETDRK4 通过 Richardson 外推法估计局部截断误差来调整步长, 因此可以根据解的变化分配计算资源来减少不必要的计算, 进而简化整个计算过程, 相较于固定步长 ETDRK4 更适合用来求解 Allen-Cahn 方程。

### 3.4. 自适应 ETD 方法的截断误差分析

本节从理论上分析自适应方法的精度, 为后文数值实验验证提供理论依据。分析方程(1)在平衡点  $u_0$  处的线性化。设  $v = u - u_0$  为小扰动, 其中  $u_0$  满足  $L(t)u_0 + N(u_0) = 0$ 。非线性项  $N(u)$  在  $u_0$  处线性化:

$$N(u) \approx N(u_0) + N'(u_0)v, \tag{2}$$

其中  $N'(u_0) = \frac{\partial N}{\partial u} \Big|_{u=u_0} = 1 - 3u_0^2$ 。为简化分析, 取平衡点  $u_0 = 0$  (即  $N(u_0) = 0$ ), 则  $N'(u_0) = 1$ 。将(2)代入方程(1), 得到线性化方程:

$$\frac{\partial v}{\partial t} = L(t)v + \lambda v, \quad \lambda = 1. \tag{3}$$

处理方程(3)的时变性, 在局部时间区间  $[t_n, t_{n+1}]$  内, 将  $\alpha(t)$  冻结为  $a_n = \alpha(t_n)$ , 线性算子固定为  $L_n = L(t_n)$ , 因此, 线性测试方程为:

$$\frac{\partial v}{\partial t} = L_n v + \lambda v, \tag{4}$$

其中精确解为  $v(t_{n+1}) = e^{(t_n + \lambda)\tau_n} v(t_n)$ , 其中  $\tau_n = t_{n+1} - t_n$  为时间步长。

将 ETDRK4 格式应用于线性测试方程(4), 在波数空间(谱空间)中线性算子  $L_n$  可对角化, 设波数  $k$  对应的线性算子特征值为  $L_n(k) = -|k|^{2\alpha_n}$ , 令  $x = \lambda\tau_n$ ,  $y = L_n(k)\tau_n$ , ETDRK4 格式的放大因子  $r(x, y)$  满足  $\hat{v}_k^{n+1} = r(x, y)\hat{v}_k^n$ 。通过代入简化得:

$$r(x, y) = e^y + \phi_1(y)x + 2\phi_2(y)x(e^{y/2} + \phi_1(y/2)x) + \phi_3(y)x(e^{y/2} + \phi_1(y/2)x),$$

其中  $\phi$  函数为:

$$\phi_1(z) = 1 + \frac{z}{2} + \frac{z^2}{6} + O(z^3), \quad \phi_2(z) = \frac{1}{2} + \frac{z}{6} + \frac{z^2}{24} + O(z^3), \quad \phi_3(z) = \frac{1}{6} + \frac{z}{24} + O(z^2)$$

局部截断误差(LTE)定义为数值解与精确解在一步内的差异:

$$\text{LTE} = \|v_{n+1} - v(t_{n+1})\| = |r(x, y) - e^{x+y}| \cdot |v_n|,$$

为分析误差, 将放大因子进行泰勒展开。假设  $x$  和  $y$  为小量, 展开至四阶项:

$$r(x, y) = 1 + (x + y) + \frac{(x + y)^2}{2} + \frac{(x + y)^3}{6} + \frac{(x + y)^4}{24} + O(\tau_n^5),$$

$$e^{x+y} = 1 + (x + y) + \frac{(x + y)^2}{2} + \frac{(x + y)^3}{6} + \frac{(x + y)^4}{24} + O(\tau_n^5)$$

因此, 局部截断误差为:  $\text{LTE} = O(\tau_n^5)$ 。LTE 与  $\tau_n^5$  成正比表明 ETDRK4 格式在时间离散上的四阶精度(即误差阶数  $O(\tau^5)$  对应四阶方法), 线性化处理冻结时变算子保证误差分析的通用性。

### 3.5. 自适应 ETD 方法的稳定性分析

本节从理论上分析自适应 ETD 方法的稳定性, 为后文数值实验验证提供理论依据。考虑非线性方程 (1) 的线性化测试方程:

$$\frac{\partial u}{\partial t} = L_n u + \lambda u,$$

其中  $\lambda=1$  (基于平衡点  $u_0=0$ )。在谱空间中, 令  $x = \lambda \tau_n$ ,  $y = L_n(k) \tau_n$ , 数值方法的稳定性由放大因子  $r(x, y)$  决定。ETDRK4 的  $r(x, y)$  可表示为有理函数形式:

$$r(x, y) = \frac{c_0(y) + c_1(y)x + c_2(y)x^2 + c_3(y)x^3 + c_4(y)x^4}{D(y)},$$

其中  $D(y) = (48 + 12y + y^2)^3 (12 + 6y + y^2)$ ,

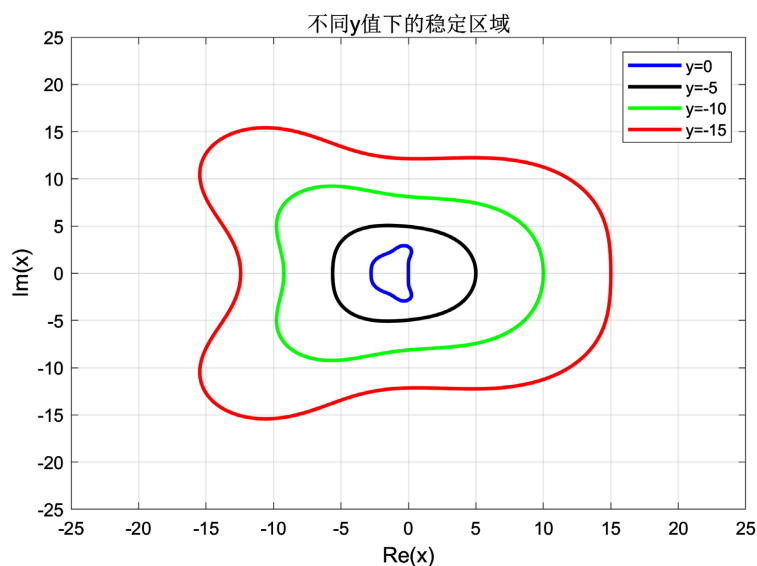
$$c_0 = 1327104 + 331776y - 55296y^2 - 20736y^3 + 3456y^4 + 2160y^5 + 372y^6 + 30y^7 + y^8,$$

$$c_1 = 1327104 + 442368y - 23040y^3 - 4608y^4 - 384y^5 - 12y^6,$$

$$c_2 = 552960 + 110592y - 25344y^2 - 6912y^3 - 336y^4 + 24y^5,$$

$$c_3 = 221184 + 55296y - 9216y^2 + 1152y^3,$$

$$c_4 = 55296 + 27648y,$$



**Figure 1.** The stable region when  $y$  is fixed at certain non-positive real values  
**图 1.** 当  $y$  固定为某些非正实数值时的稳定区域

图 1 展示当  $|y| \rightarrow \infty$  时  $|r(x, y)| \rightarrow 0$ , 当  $y$  取负实数时 (即  $L_n(k) \leq 0$ ) ETDRK4 方法  $|r(x, y)|$  的稳定区域 (即绿色区域), 变阶问题中阶数  $\alpha(t)$  的变化会影响  $y = -|k|^{\alpha} \tau_n$  的分布, 当  $\alpha(t)$  增大时,  $|y|$  增大, 系统刚性增强, 反之减小。

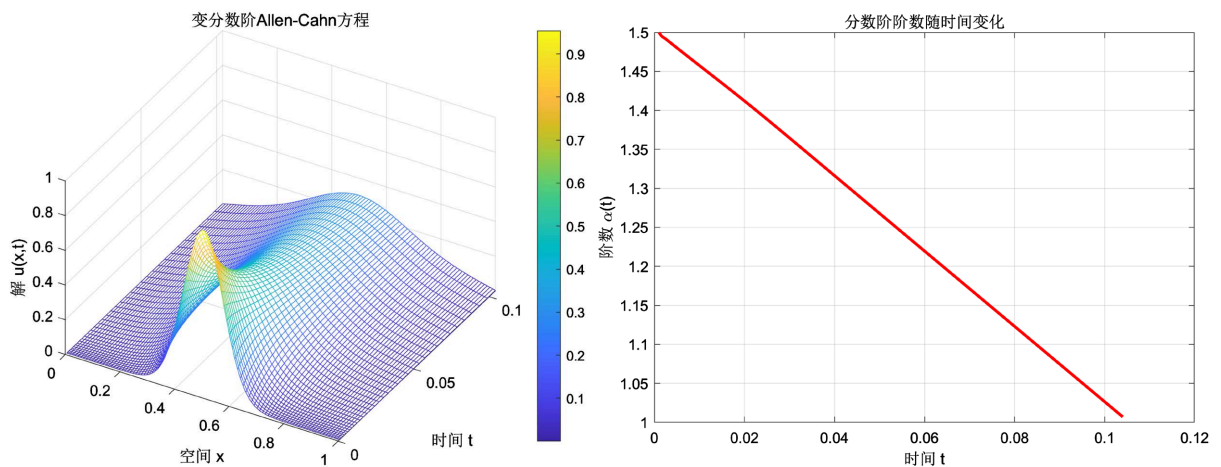
自适应 ETDRK4 方法通过调整  $\tau_n$  确保  $y$  落在稳定区域内并保证局部截断误差控制为  $O(\tau_n^5)$ , 表明此方法适用于变分数阶 Allen-Cahn 方程。

## 4. 数值实验

为验证方法性能, 考虑一维和二维变分数阶 Allen-Cahn 方程, 设  $N(u) = u - u^3$ , 阶数函数设为  $\alpha(t) = 1.5 - 0.5t/T$ 。为验证方法的有效性, 将自适应 ETDRK4 方法与经典 RK4、固定步长 ETDRK4 方法进行对比。由于该方程为非线性方程, 通常难以获得解析解, 因此, 本文中所有“最大误差”的计算, 均采用超高精度数值方法获得的数值解作为参考解。该参考解的具体生成方式如下: 空间离散采用分步傅里叶谱方法, 并使用足够多的网格点(一维:  $N = 1024$ ; 二维:  $N_x = N_y = 512$ )以确保空间误差可忽略; 时间离散采用固定步长的四阶指数时间差分方法(ETDRK4), 并选取极小步长  $\tau = 1e-7$  进行积分, 以保证时间离散误差远小于其他对比方法。将待评估方法(经典 RK4、固定步长 ETDRK4、自适应 ETDRK4)在最终时刻  $t = T$  得到的数值解, 与上述参考解在相同空间网格点上的值进行比较, 取其无穷范数( $\infty$ -范数)作为“最大误差”。

### 4.1. 一维 Allen-Cahn 方程的数值模拟

考虑方程(1),  $T = 0.01$ , 初始条件为  $u(x, 0) = \exp(-100(x - 0.5)^2)$ , 边界条件为  $u(0, t) = u(1, t) = 0$ 。本实验将记录计算过程中的“接受步数”与“拒绝步数”, 观察步长  $\tau_n$  随时间的演化序列, 直接验证自适应 ETDRK4 方法能否自动调整步长来保证精度和效率。



**Figure 2.** The evolution of the one-dimensional Allen-Cahn equation solution and the change of its order over time  
**图 2.** 一维 Allen-Cahn 方程解的演化和阶数随时间的变化

图 2 展示初始高斯脉冲在变分数阶扩散算子与非线性反应项耦合作用下的演化行为, 整个演化曲面平滑且无数值振荡。图 2 给出函数  $\alpha(t)$  从初始的  $\alpha(0) = 1.5$  线性下降至  $\alpha(0.1) = 1.0$ , 即系统的“超扩散”强度减弱而“记忆效应”增强, 表明自适应 ETD 方法的稳定性。

本节通过一维 Allen-Cahn 方程性能对比证明自适应 ETDRK4 方法的优势, 实验过程保证空间离散格式、终止时间和计算环境相同, 固定步长 ETDRK4 方法选取三个具有代表性的步长: 一个较小的步长、一个较大的步长和一个与自适应 ETDRK4 方法平均步长相近的步长。

表 2 中“最大误差”为数值解与解析解在最终时刻的无穷范数误差。固定步长 ETDRK4 在  $\tau = 1e-4$  时误差为  $2.89e-4$ , 耗时 0.21s, 而自适应 ETDRK4 的误差( $2.45e-4$ )更小, 总时间步数(85 步)更小, 计算时间(0.15 s)更短, 效率提升约 28.6%。固定步长 ETDRK4 在  $\tau = 1.18e-4$  时计算时间(0.13 s)与自适应 ETDRK4 (0.15 s)相近, 但误差( $5.12e-4$ )远大于自适应 ETDRK4 ( $2.45e-4$ )。

**Table 2.** One-dimensional example performance comparison results  
**表 2.** 一维算例性能对比结果

| 方法类型         | 步长策略/容差          | 总时间步数 | 计算时间(s) | 最大误差      |
|--------------|------------------|-------|---------|-----------|
| 经典 RK4       | $\tau = 1e-5$    | 1000  | 0.95    | $3.21e-4$ |
| 固定步长 ETD RK4 | $\tau = 2e-4$    | 50    | 0.08    | $1.02e-3$ |
| 固定步长 ETD RK4 | $\tau = 1.18e-4$ | 85    | 0.13    | $5.12e-4$ |
| 固定步长 ETD RK4 | $\tau = 1e-4$    | 100   | 0.21    | $2.89e-4$ |
| 自适应 ETD RK4  | tol = $1e-4$     | 85    | 0.15    | $2.45e-4$ |

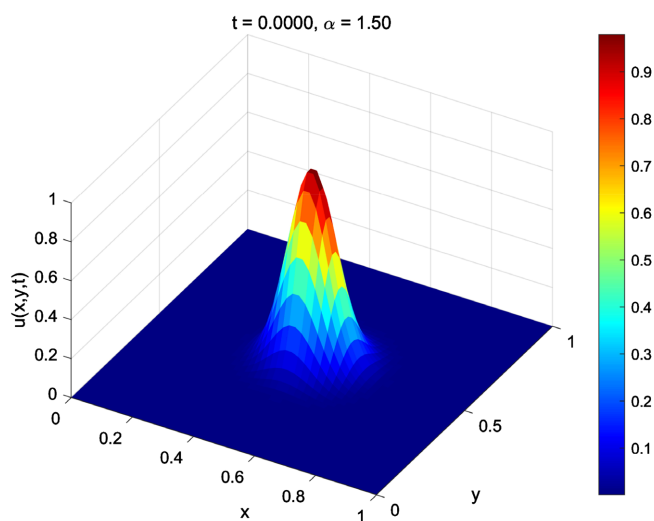
## 4.2. 二维 Allen-Cahn 方程的数值模拟

考虑方程(1),  $T = 0.02$ , 边界条件为齐次狄利克雷边界条件, 初始条件为  $u(x, y, 0) = \exp\left(-\frac{(x-0.5)^2 + (y-0.5)^2}{2\sigma^2}\right)$ ,  $\sigma = 0.1$ 。监测  $t = 0.000$ 、 $t = 0.010$ 、 $t = 0.015$ 、 $t = 0.020$  四个时间点的解  $u(x, y, t)$  的形态变化, 分析阶数递减对扩散行为的影响。本实验将记录并分析在整个积分过程中, 自适应方法随时间  $t$  或解的变化程度的动态调整时间步长序列  $\{\tau_n\}$  的过程。

图 3~6 展示了从中心峰值变为平滑分布的演化过程, 当阶数  $\alpha(t)$  从 1.5 降至 1.0, 系统的扩散速率减慢而记忆效应增强。自适应步长策略在解变化剧烈的初始阶段采用较小步长, 随着解趋于平稳随之增大步长, 通过动态调整时间步长保证计算精度并提高计算效率, 验证了自适应 ETD 方法对二维 Allen-Cahn 方程的适用性。

本节通过二维 Allen-Cahn 方程性能对比证明自适应 ETD RK4 方法的优势。

从表 3 可以看出, 固定步长 ETD RK4 在  $\tau = 5e-5$  时误差达到  $7.89e-4$ , 计算时间为 2.11 s, 而自适应 ETD RK4 精度( $7.05e-4$ )更高, 总时间步数(276 步)更少, 计算时间减少到 1.55 s, 效率提升约 26.5%。固定步长 ETD RK4 在  $\tau = 1e-4$  时计算时间(1.05 s)少于自适应 ETD RK4 (1.55 s), 但误差( $3.21e-3$ )显著大于自适应 ETD RK4 ( $7.05e-4$ ), 固定步长 ETD RK4 使用  $\tau = 5e-5$  时才能取得与自适应 ETD RK4 相近精度 ( $7.89e-4$ ), 此时计算时间增加到 2.11 s。



**Figure 3.** Numerical solution of the two-dimensional equation at  $t = 0$   
**图 3.** 二维方程在  $t = 0$  时的数值解

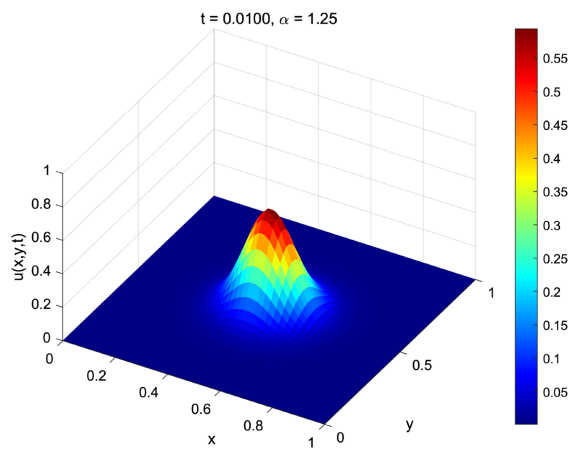


Figure 4. Numerical solution of the two-dimensional equation at  $t = 0.01$

图 4. 二维方程在  $t = 0.01$  时的数值解

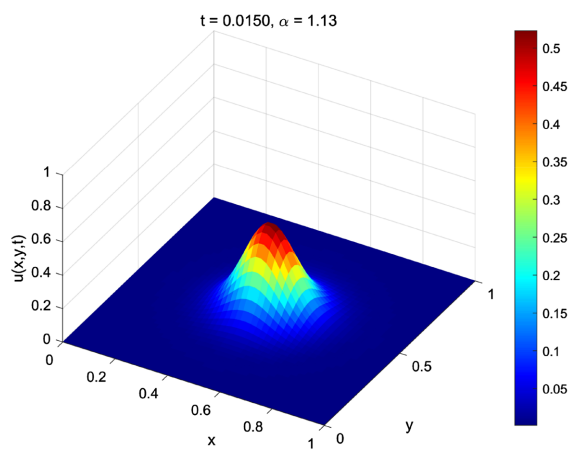


Figure 5. Numerical solution of the two-dimensional equation at  $t = 0.015$

图 5. 二维方程在  $t = 0.015$  时的数值解

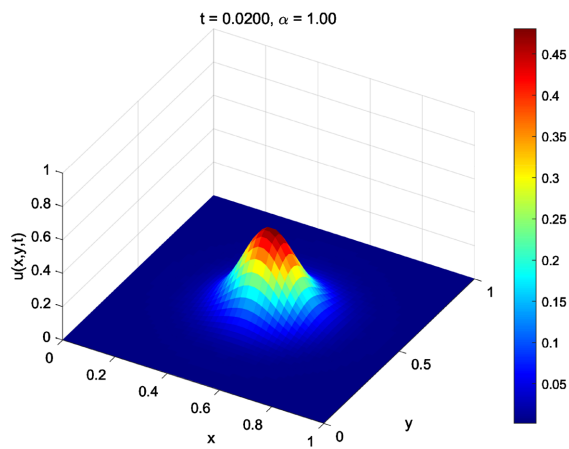


Figure 6. Numerical solution of the two-dimensional equation at  $t = 0.02$

图 6. 二维方程在  $t = 0.02$  时的数值解

**Table 3.** Two-dimensional example performance comparison results**表 3.** 二维算例性能对比结果

| 方法类型        | 步长策略/容差             | 总时间步数 | 计算时间(s) | 最大误差      |
|-------------|---------------------|-------|---------|-----------|
| 经典 RK4      | $\tau = 1e-6$       | 20000 | 48.7    | $4.12e-4$ |
| 固定步长 ETDRK4 | $\tau = 2e-4$       | 100   | 0.52    | $8.76e-3$ |
| 固定步长 ETDRK4 | $\tau = 1e-4$       | 200   | 1.05    | $3.21e-3$ |
| 固定步长 ETDRK4 | $\tau = 5e-5$       | 400   | 2.11    | $7.89e-4$ |
| 自适应 ETDRK4  | $\text{tol} = 5e-5$ | 276   | 1.55    | $7.05e-4$ |

一维和二维算例性能对比结果验证了自适应 ETDRK4 方法通过自适应步长调整策略简化计算过程, 在求解变分数阶 Allen-Cahn 方程时计算更快, 精度更高, 表明本文提出的自适应 ETD 方法的计算效率较经典 RK4 和固定步长 ETDRK4 明显提高。这主要得益于其动态步长策略: 在解变化剧烈的初始阶段分数阶高, 强扩散效应强, 算法自动采用小步长以保证精度与稳定性; 而当解趋于平缓、系统刚度减弱时, 则能够增大步长, 避免了固定步长方法在平滑区域进行的冗余计算。

## 5. 结论

本文针对变分数阶 Allen-Cahn 方程提出了自适应 ETD 方法, 通过将自适应步长控制机制与高阶 ETD 格式相结合, 在空间离散方面, 对比了有限差分法和分步傅里叶谱方法, 并结合 Padé 近似技术, 系统研究变阶问题中的数值计算。实验结果表明, 在设定相同容差的前提下, 自适应方法能自动调整步长, 其最终误差与使用更小固定步长的 ETDRK4 方法相当, 但总计算步数和时间更少。自适应 ETDRK4 方法特别适用于解的动态行为剧烈变化或系统刚度时变的科学计算问题, 如本文研究的变分数阶非线性方程。其核心优势在于能够根据局部误差估计智能分配计算资源, 在保证预设精度的同时, 显著提升整体计算效率。本文的数值结果验证了该方法的有效性、稳定性及相较于传统固定步长策略的优越性。

## 基金项目

湖北文理学院横向科研项目(HX2023030)。

## 参考文献

- [1] Samko, S.G. and Ross, B. (1993) Integration and Differentiation to a Variable Fractional Order. *Integral Transforms and Special Functions*, **1**, 277-300. <https://doi.org/10.1080/10652469308819027>
- [2] Sun, H.G., Chen, W., Wei, H. and Chen, Y.Q. (2011) A Comparative Study of Constant-Order and Variable-Order Fractional Models in Characterizing Memory Property of Systems. *The European Physical Journal Special Topics*, **193**, 185-192. <https://doi.org/10.1140/epjst/e2011-01390-6>
- [3] Wang, G.T., Pei, K., Agarwal, R.P., Zhang, L. and Ahmad, B. (2018) Nonlocal Hadamard Fractional Boundary Value Problem with Hadamard Integral and Discrete Boundary Conditions on a Half-Line. *Journal of Computational and Applied Mathematics*, **343**, 230-239. <https://doi.org/10.1016/j.cam.2018.04.062>
- [4] 尹修草, 周均, 胡兵. 分数阶对流-弥散方程的有限差分方法[J]. 四川大学学报(自然科学版), 2013, 50(3): 409-413.
- [5] 陶群群, 吴亚运. 变系数的分数阶非齐次线性微分方程[J]. 应用数学与计算数学学报, 2015, 29(2): 154-161.
- [6] 宋光珍, 赵维加, 黄健飞. 时间分数阶扩散方程的一种数值解法[J]. 青岛大学学报(自然科学版), 2015, 28(3): 9-14.
- [7] 曾宝思, 尹修草, 谢常平, 等. 带 Robin 知更鸟边界条件的分数阶对流-扩散方程的数值解法[J]. 四川大学学报

(自然科学版), 2018, 55(1): 13-17.

- [8] 李俊婵. 分数阶对流扩散方程的有限点方法研究[D]: [硕士学位论文]. 西安: 西安理工大学, 2019.
- [9] Cox, S.M. and Matthews, P.C. (2002) Exponential Time Differencing for Stiff Systems. *Journal of Computational Physics*, **176**, 430-455. <https://doi.org/10.1006/jcph.2002.6995>