

物理实验

一种由 Brusselator 模型稳态解引出的
偏微分方程含时演化算法黄得清¹ 方爱平¹ 易思睿¹ 李屹山¹ 蒋臣威¹ 张修兴² 王小力¹(¹ 西安交通大学物理学院, 陕西 西安 710049; ² 渭南师范学院物理与电气工程学院, 陕西 渭南 714099)

摘 要 以二维 Brusselator 模型为研究对象, 本文利用含时演化算法得到了图灵斑图稳态解。基于对 Brusselator 模型的研究, 我们提出了一种新的偏微分方程(PDE)边值问题含时演化算法: 通过引入虚拟的时间参量, 将原本的边值问题转化为一个初值问题, 结合有限差分法与隐式迭代算法模拟系统演化, 从而逼近稳态解, 也就是原方程组的边值问题解。研究发现, 与传统 Jacobi 迭代法和 Gauss-Seidel 迭代法相比, 含时演化算法在复杂非线性问题中展现出更强的稳定性与收敛性。进一步将该方法拓展至泊松方程和 Helmholtz 方程的求解, 结果表明: 尽管该含时演化算法在一般线性问题中的精度与效率略逊于 Gauss-Seidel 迭代法, 但在大波数 Helmholtz 方程等特定场景下, 其收敛速度与误差精度表现更优。含时演化算法为非线性 PDE 边值问题提供了新的求解思路, 未来可通过结合高阶差分格式以及并行计算等策略进一步提升其性能。

关键词 偏微分方程; 含时演化算法; Brusselator 模型; 图灵斑图; 有限差分法

DOI: 10.27024/j.wlygc.2025.03.10.01

A TIME-EVOLUTION ALGORITHM FOR PARTIAL DIFFERENTIAL EQUATIONS DERIVED FROM THE STEADY-STATE SOLUTION OF THE BRUSSELATOR MODEL

HUANG Deqing¹ FANG Aiping¹ Yi Sirui¹ Li Yishan¹ JIANG Chenwei¹
ZHANG Xiuxing² WANG Xiaoli¹

(¹ School of Physics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049 ;

² School of Physics and Electrical Engineering, Weinan Normal University, Weinan, Shaanxi 714099)

Abstract For the two-dimensional Brusselator model, this paper obtains the steady-state solution of Turing patterns by using a time-evolution algorithm. Based on the research of the Brusselator model, we propose a new time-evolution algorithm for the boundary value problem of partial differential equations (PDEs). By introducing a virtual time parameter, the original boundary value problem is transformed into an initial value problem. Combining the finite difference method and the implicit iterative algorithm, the system evolution is simulated to approximate the steady-state solution, that is, the solution of the boundary value problem of the original equation set. The research shows that compared with the traditional Jacobi iter-

收稿日期: 2025-03-10

基金项目: 西安交通大学 2024 年课程思政专项研究项目资助; 2023 年基层教学组织教学改革研究专项(基础课程); 渭南师范学院教育科学研究项目(2020JYKX021)。

通信作者: 蒋臣威, jiangcw@mail.xjtu.edu.cn。

引文格式: 黄得清, 方爱平, 易思睿, 等. 一种由 Brusselator 模型稳态解引出的偏微分方程含时演化算法[J]. 物理与工程, 2026, 36(1): 163-173.

Cite this article: HUANG D Q, FANG A P, YI S R, et al. A time-evolution algorithm for partial differential equations derived from the steady-state solution of the Brusselator model[J]. Physics and Engineering, 2026, 36(1): 163-173. (in Chinese)

ative method and Gauss-Seidel iterative method, the time-evolution algorithm exhibits stronger stability and convergence in complex non-linear problems. Further, this method is extended to the solution of Poisson's equation and Helmholtz equation. The results show that although the accuracy and efficiency of the time-evolution algorithm in general linear problems are slightly inferior to those of the Gauss-Seidel iterative method, in specific scenarios such as the Helmholtz equation with large wave numbers, its convergence speed and error accuracy are better. The time-evolution algorithm provides a new solution idea for the boundary value problem of non-linear PDEs. In the future, its performance can be further improved by combining high-order difference schemes and parallel computing strategies.

Key words partial differential equations; time-evolution algorithm; Brusselator model; Turing patterns; finite difference method

偏微分方程(PDE)能够描述诸如流体流动、电磁场分布、热传导以及波动传播等物理现象,为理解和预测自然与工程系统的行为提供了强有力的工具,在物理、化学、工程及金融等多个领域有着广泛应用^[1,2]。

尽管从理论上可以分析出某些 PDE 解的存在及唯一性,完全从数学演算推导解的解析形式往往面临巨大挑战:复杂的边界条件、不规则几何形状以及非线性项的存在,使得许多情况下难以找到解析解。因此,数值解法成为解决这些问题的关键途径^[3,4]。其中,有限差分法是一种最基本、最常见且简单易用的 PDE 数值求解方法,其基本思想是将 PDE 的自变量区间离散化,从而将原先拥有连续自由度的微分方程转化为有限个离散自由度的差分方程,进而通过计算机进行数值求解。然而,不同的有限差分方法有不同的精度、效率和适用条件,单一的有限差分方法并不能很好地应对所有问题,因此研究特定情况下新的有限差分方法是很有价值的。

Brusselator 模型是一组模拟自催化反应的非线性微分方程,由 Ilya Prigogine 及其合作者于 1968 年在布鲁塞尔自由大学提出,用于求解化学反应中各组分浓度随时间的变化,在化学反应中的自组织现象研究中具有广泛的应用^[5]。该模型不仅在验证耗散结构理论上比较有效,在生物化学反应过程中的一些催化反应上也应用广泛。对 Brusselator 模型的研究,无论是从广度上还是从深度上,都是令人感兴趣的。

在特定参数下,Brusselator 模型能够演化出一些稳定存在但又不单调的分布模式,一般为斑点状、细粒状或者条纹状,非常类似于生物斑纹。图灵最先用非线性方程组来描述生物斑纹,很好

地解释了它们生成的动力学机制,此后由该机制生成的斑纹便被学术界命名为“图灵斑图”。本文的研究的 Brusselator 模型参数也是在生成图灵斑图的范围内,有助于观察系统的收敛状况进而探讨算法的适用性与性能。

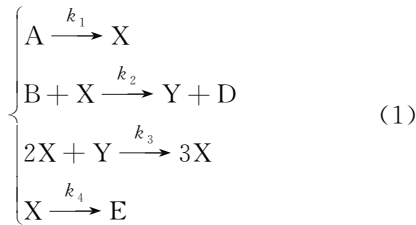
本文利用数值方法求解了二维 Brusselator 模型对应的偏微分方程组,得到了其稳态解-图灵斑图。此外,基于对 Brusselator 模型的研究,我们提出了一种新的 PDE 边值问题含时演化算法:通过引入虚拟的时间参量,将原本的边值问题转化为一个初边值问题,该方法通过模拟系统随时间的演化,能够逐步逼近稳态解,同时也是原方程组的边值问题解。与传统的 PDE 边值问题数值求解方法(如 Jacobi 迭代法和 Gauss-Seidel 迭代法)相比,这种含时演化算法通过引入时间变量使得迭代计算的稳定性和收敛性都得到显著增强。此外,通过将该含时演化算法应用于泊松方程和 Helmholtz 方程的数值求解,我们验证了该算法在更广泛的 PDE 问题中的适用性。

大学物理和其他物理课程学习中不仅要面对很多可以解析计算的物理问题,还存在很多只能通过数值计算才能解决的物理问题。通过这些较复杂物理问题的数值求解培养学生的数值计算思维与能力,对学生将来利用物理知识解决复杂科学和工程问题至关重要。因此,本文的研究结果对于大学物理和其他物理课程教学也是有益的。

1 Brusselator 模型含时演化数值模拟

1.1 反应式及浓度方程

Brusselator 模型,又称三分子模型^[5],包含以下 4 步化学反应



反应发生在固定的一块区域内,各种物质浓度 n_i 在这块区域内都有一定的场分布 $n_i(\mathbf{r}, t)$ 。在反应中不断供给反应物 A 和 B,使其浓度保持恒定(n_A, n_B 为非零常数),并不断将生成物 D 和 E 排除($n_D = n_E = 0$),于是反应单向进行(逆反应在 A、B 充足, D、E 为 0 时可以忽略),系统处在远离平衡的状态。在这种情形下,只有 X 和 Y 的浓度(n_x, n_y)随时间变化。设时间参数为 t' ,根据化学反应速率计算方法:由式(1)中第一个方程可得 $\frac{\partial n_x}{\partial t'} = k_1 n_A$,由第二个方程可得 $-\frac{\partial n_x}{\partial t'} = \frac{\partial n_y}{\partial t'} = k_2 n_B n_x$,由第三个方程(看成消耗一份 Y 生成一份 X)可得 $\frac{\partial n_x}{\partial t'} = -\frac{\partial n_y}{\partial t'} = k_3 n_x^2 n_y$,由第四个方程可得 $\frac{\partial n_x}{\partial t'} = -k_4 n_x$,最后加上扩散项 $\frac{\partial n}{\partial t'} = -\nabla \cdot \mathbf{J} = -\nabla \cdot (-D \nabla n) = D \nabla^2 n$,耦合得到浓度变化速率方程组为

$$\begin{cases} \frac{\partial n_x}{\partial t'} = k_1 n_A - (k_2 n_B + k_4) n_x + k_3 n_x^2 n_y + D_1' \nabla^2 n_x \\ \frac{\partial n_y}{\partial t'} = k_2 n_B n_x - k_3 n_x^2 n_y + D_2' \nabla^2 n_y \end{cases} \quad (2)$$

采用变量代换

$$\begin{cases} t = k_4 t' \\ X = \left(\frac{k_3}{k_4}\right)^{\frac{1}{2}} n_x; Y = \left(\frac{k_3}{k_4}\right)^{\frac{1}{2}} n_y \\ A = \left(\frac{k_1 k_3}{k_4^3}\right)^{\frac{1}{2}} n_A; B = \left(\frac{k_2}{k_4}\right) n_B \\ D_1 = \frac{D_1'}{k_4}; D_2 = \frac{D_2'}{k_4} \end{cases} \quad (3)$$

方程组(2)可转化为

$$\begin{cases} \frac{\partial X}{\partial t} = A - (B + 1) X + X^2 Y + D_1 \nabla^2 X \\ \frac{\partial Y}{\partial t} = B X - X^2 Y + D_2 \nabla^2 Y \end{cases} \quad (4)$$

这里进行的只是标度变换,不影响我们对系统演化的分析,于是也称 X, Y 为物质浓度, t 为反

应时间。

在这个反应中, X, Y 是反应的中间产物,也是重要的演化物质,我们希望它是无源的(系统中不存在除化学反应外直接产生 X, Y 的事物,这排除了固定边界条件这种在边界上有源的情况)、守恒的(系统中的 X, Y 不扩散出系统外,这表明 X, Y 浓度在边界处梯度为 0),于是采用无通量边界条件

$$\mathbf{e}_n \cdot \nabla X = \mathbf{e}_n \cdot \nabla Y = 0 \quad (5)$$

1.2 有限差分法离散化

本文仅研究二维 Brusselator 模型问题,因其更具有一般性和代表性。

将方程组(4)中的时间和空间都离散化,时间步长 τ (用上标 n 标记),空间步长 h (用下标 i, j 标记),将拉普拉斯算符 ∇^2 拆分成沿 x 轴的算符 ∇_i^2 和沿 y 轴的算符 ∇_j^2 ,并运用平均隐式差格式

$$\begin{cases} \frac{X_{ij}^{(n)} - X_{ij}^{(n-1)}}{\tau} = A - (B + 1) X_{ij}^{(n-1)} + (X_{ij}^{(n-1)})^2 Y_{ij}^{(n-1)} + \frac{D_1}{2} (\nabla_i^2 X_{ij}^{(n-1)} + \nabla_i^2 X_{ij}^{(n)} + \nabla_j^2 X_{ij}^{(n-1)} + \nabla_j^2 X_{ij}^{(n)}) \\ \frac{Y_{ij}^{(n)} - Y_{ij}^{(n-1)}}{\tau} = B X_{ij}^{(n-1)} - (X_{ij}^{(n-1)})^2 Y_{ij}^{(n-1)} + \frac{D_2}{2} (\nabla_i^2 Y_{ij}^{(n-1)} + \nabla_i^2 Y_{ij}^{(n)} + \nabla_j^2 Y_{ij}^{(n-1)} + \nabla_j^2 Y_{ij}^{(n)}) \end{cases} \quad (6)$$

采用变量代换

$$\begin{cases} k_1 = \frac{\tau D_1}{2h^2} \\ k_2 = \frac{\tau D_2}{2h^2} \\ \alpha_i^{(n-1)} = \tau [A - (B + 1) X_i^{(n-1)} + (X_i^{(n-1)})^2 Y_i^{(n-1)}] \\ \beta_i^{(n-1)} = \tau [B X_i^{(n-1)} - (X_i^{(n-1)})^2 Y_i^{(n-1)}] \end{cases} \quad (7)$$

则方程组(6)转化为

$$\begin{cases} X_{ij}^{(n)} - X_{ij}^{(n-1)} = \alpha_{ij}^{(n-1)} + k_1 h^2 [\nabla_i^2 X_{ij}^{(n-1)} + \nabla_i^2 X_{ij}^{(n)} + \nabla_j^2 X_{ij}^{(n-1)} + \nabla_j^2 X_{ij}^{(n)}] \\ Y_{ij}^{(n)} - Y_{ij}^{(n-1)} = \beta_{ij}^{(n-1)} + k_2 h^2 [\nabla_i^2 Y_{ij}^{(n-1)} + \nabla_i^2 Y_{ij}^{(n)} + \nabla_j^2 Y_{ij}^{(n-1)} + \nabla_j^2 Y_{ij}^{(n)}] \end{cases} \quad (8)$$

1.3 解离散方程的迭代计算

以下展示如何由方程组(8)的第一个方程从

$X_i^{(n-1)}$ 求出 $X_i^{(n)}$, 思路类似于交替方向隐式算法^[6], 是一种改进版的多维度隐式迭代算法。

注意到在空间离散化的情况下方程组(8)其实是一系列线性方程组, 可以先形式化地解这个线性方程组, 算符的倒数代表求逆算符, 得

$$X_{ij}^{(n)} = \frac{1}{1 - k_1 h^2 \nabla_i^2 - k_1 h^2 \nabla_j^2} (\alpha_{ij}^{(n-1)} + 2X_{ij}^{(n-1)}) - X_{ij}^{(n-1)} \quad (9)$$

记包含逆算符的项

$$\phi_{ij} = \frac{1}{1 - k_1 h^2 \nabla_i^2 - k_1 h^2 \nabla_j^2} (\alpha_{ij}^{(n-1)} + 2X_{ij}^{(n-1)}) \quad (10)$$

则

$$(1 - k_1 h^2 \nabla_i^2 - k_1 h^2 \nabla_j^2) \phi_{ij} = \alpha_{ij}^{(n-1)} + 2X_{ij}^{(n-1)} \quad (11)$$

由于 $k_1 h^2 = \frac{\tau D_1}{2}$ 很小, 忽略高阶算符 $\nabla_i^2 \nabla_j^2$ 项, 式(11)可分解为

$$(1 - k_1 h^2 \nabla_i^2) (1 - k_1 h^2 \nabla_j^2) \phi_{ij} = \alpha_{ij}^{(n-1)} + 2X_{ij}^{(n-1)} \quad (12)$$

由上, 可先求中间变量 φ_{ij} , 其满足

$$(1 - k_1 h^2 \nabla_i^2) \varphi_{ij} = \alpha_{ij}^{(n-1)} + 2X_{ij}^{(n-1)} \quad (13)$$

再求 ϕ_{ij} , 其满足

$$(1 - k_1 h^2 \nabla_j^2) \phi_{ij} = \varphi_{ij} \quad (14)$$

式(13)、式(14)都是一系列线性方程组。

在式(13)中, 对于固定的 j , 平均隐式差分格式为

$$\varphi_{ij} - k_1 (\varphi_{i+1,j} - 2\varphi_{ij} + \varphi_{i-1,j}) = \alpha_{ij}^{(n-1)} + 2X_{ij}^{(n-1)} \quad (15)$$

在式(14)中, 对于固定的 i , 平均隐式差分格式为

$$\phi_{ij} - k_1 (\phi_{i+1,j} - 2\phi_{ij} + \phi_{i-1,j}) = \varphi_{ij} \quad (16)$$

给定 $X_i^{(n-1)}, Y_i^{(n-1)}, \alpha_i^{(n-1)}, \beta_i^{(n-1)}$, 可由方程组(15)求出 φ_{ij} , 进而由方程组(16)求出 ϕ_{ij} 和 $X_i^{(n)}$ 。同理可以求出 $Y_i^{(n)}$ 。式(15)和式(16)这种系数矩阵为三对角形式的 n 维线性方程组有非常高效的算法, 不用求逆矩阵或者高斯消元, 时间复杂度可以达到 $O(n)$ ^[5,6]。

最后, 对于无通量边界条件, 只需要在每次迭代之后消除边界梯度, 对于 $\forall i, j$, 赋值

$$\begin{cases} X_{1,j}^{(n)} = X_{2,j}^{(n)}, X_{end,j}^{(n)} = X_{end-1,j}^{(n)} \\ Y_{1,j}^{(n)} = Y_{2,j}^{(n)}, Y_{end,j}^{(n)} = Y_{end-1,j}^{(n)} \\ X_{i,1}^{(n)} = X_{i,2}^{(n)}, X_{i,end}^{(n)} = X_{i,end-1}^{(n)} \\ Y_{i,1}^{(n)} = Y_{i,2}^{(n)}, Y_{i,end}^{(n)} = Y_{i,end-1}^{(n)} \end{cases} \quad (17)$$

综上, 我们可以从 $X_{ij}^{(0)}, Y_{ij}^{(0)}$ 经过多次迭代逐步生成 $X_{ij}^{(1)}, Y_{ij}^{(1)}, \dots, X_{ij}^{(N)}, Y_{ij}^{(N)}$ (N 由最大时间确定), 从而展现出系统随时间的变化图像。以上求解过程也在如图1所示的算法流程图中展现。

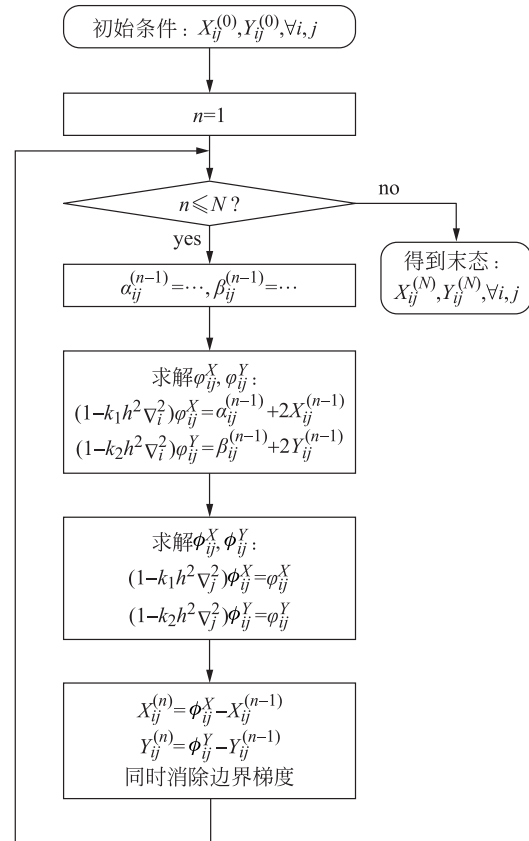


图1 含时演化算法迭代计算流程图

2 Brusselator 模型稳态解的计算

2.1 参数设置

本文旨在讨论 PDE 的含时演化算法, 不对 Brusselator 模型的参数规律进行深入的研究。对于方程组(4)的参数, 我们取

$$A = 2, B = 5, D_1 = 0.001, D_2 = 0.004 \quad (18)$$

空间坐标 x, y 范围为正方形区域 $0 \leq x, y \leq 1$ 。

初始条件为

$$\begin{cases} X = A \\ Y = \frac{B}{A} \left[1 + 0.01 \exp \left(-\frac{(x - 0.4)^2 + (y - 0.8)^2}{0.01} \right) \right] \end{cases} \quad (19)$$

此初始分布可视为初始时在定常平衡分布 $X=A, Y=\frac{B}{A}$ 的坐标 $(0.4, 0.8)$ 处加入一定量的物质 Y 。图 2 给出了 X, Y 的初始浓度分布。可以看出, 此分布可视为系统在平衡点附近的浓度微扰。

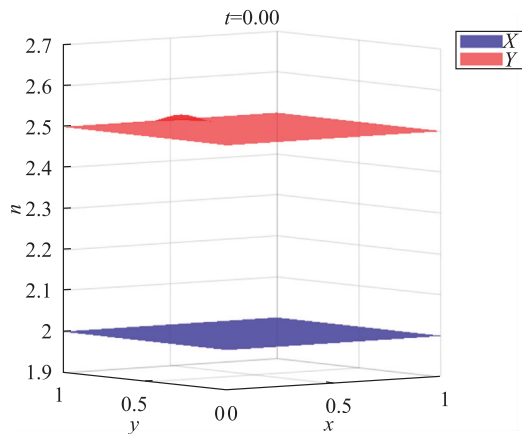


图 2 Brusselator 模型初始浓度分布曲面图

需要说明的是, 式(18)的参数设置和式(19)的初值形式都是为了使系统最终演化到非平凡的稳定解—图灵斑图。结合文献^[5,7]中对系统的线性稳定性分析, 可知当前系统的状态是不稳定的, 无法回到 $X=A, Y=\frac{B}{A}$ 的定常平衡态。当时间趋于无穷时, 将趋于非平凡的稳态解, 即为图灵斑图。

2.2 基于含时演化迭代的数值求解

根据本文第 1 部分的算法流程, 取 $h=0.01$, $\tau=0.01$, 按照式(19)给定的初始条件, 系统在不同时刻的浓度分布在图 3 中给出。

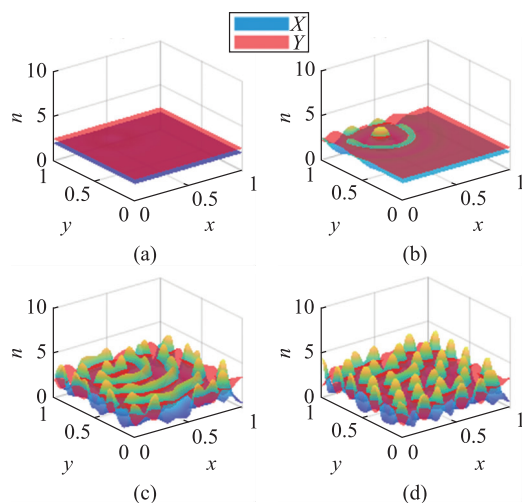


图 3 Brusselator 模型含时演化得到的浓度分布图像

从图 3 可以看出, 系统逐渐趋于一个在空间规律分布的稳态解, 这是典型的图灵斑生成过程^[7,8]。

从图 3(d)的 z 轴正向往 xOy 平面上看浓度曲面, 浓度大小对应不同颜色, 看到的即为图灵斑图, 如图 4 所示。

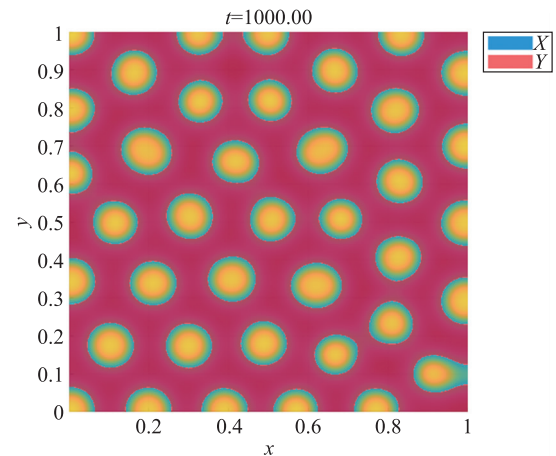


图 4 Brusselator 模型的稳态解——图灵斑图

在上面的求解过程中, 基于时间演化的隐式迭代算法稳定地给出了系统远离定常平衡的另一种稳定模式—图灵斑图。在浓度变化速率微分方程组(4)中, 将时间导数项全部设置为 0, 可得到只含有空间导数的偏微分方程组, 它是一个边值问题, 应该能够表征该图灵斑稳定的分布, 所以可以直接求解这个边值问题。但在后面的研究中发现, 虽然剔除时间维度后方程变得简单, 常见的迭代方法(如 Jacobi 迭代法 Gauss-Seidel 迭代法)并不能给出稳定收敛的解, 这说明了时间维度的重要性。

2.3 基于边值问题迭代的发散问题

当时间趋于无穷时

$$\frac{\partial X}{\partial t} = \frac{\partial Y}{\partial t} = 0 \quad (20)$$

方程组(4)转化为

$$\begin{cases} \nabla^2 X = -\frac{1}{D_1} (A - (B+1)X + X^2Y) \\ \nabla^2 Y = -\frac{1}{D_2} (BX - X^2Y) \end{cases} \quad (21)$$

方程组(21)是椭圆形偏微分方程, 可以使用椭圆形偏微分方程的常用数值方法 Jacobi 迭代法和 Gauss-Seidel 迭代法进行数值求解。Jacobi 迭代通过同步更新全部空间点, 天然适配并行架构; 而 Gauss-Seidel 迭代法利用异步更新实时整合最

新信息,效率相比 Jacobi 迭代有所提升,但不太容易实现并行计算。关于上述两种算法的更详细介绍可查阅参考文献[3-5]。

含松弛因子的 Jacobi 迭代法的表达式为

$$U_{i,j}^{(n)} = \frac{\omega}{4}(-S_{i,j}^{(n-1)} + U_{i+1,j}^{(n-1)} + U_{i-1,j}^{(n-1)} + U_{i,j+1}^{(n-1)} + U_{i,j-1}^{(n-1)}) + (1-\omega)U_{i,j}^{(n-1)} \quad (22)$$

含松弛因子的 Gauss-Seidel 迭代法的表达式为

$$U_{i,j}^{(n)} = \frac{\omega}{4}(-S_{i,j}^{(n-1)} + U_{i+1,j}^{(n-1)} + U_{i-1,j}^{(n)} + U_{i,j+1}^{(n-1)} + U_{i,j-1}^{(n)}) + (1-\omega)U_{i,j}^{(n-1)} \quad (23)$$

其中 $U=X, Y, \omega$ 为松弛因子 ($0 < \omega < 2$), S 为式(21)的右端项。关于 $\nabla^2 X$ 的方程: $S = -\frac{1}{D_1} \cdot (A - (B+1)X + X^2Y)$; 关于 $\nabla^2 Y$ 的方程: $S = -\frac{1}{D_2}(BX - X^2Y)$ 。

上述 Jacobi 迭代法和 Gauss-Seidel 迭代法中的松弛因子是迭代算法中调节新旧解权重比例的超参数,其值等于 1 时为标准迭代(新值完全替代旧值),大于 1 时通过增强新值权重加速收敛(超松弛),小于 1 时保留更多旧值以避免振荡(欠松弛),核心作用是在收敛速度与稳定性之间取得平衡,防止算法因过度依赖历史数据而迟滞或因步长过大而发散。关于含松弛因子的 Jacobi 迭代法和 Gauss-Seidel 迭代法的更多知识可以查阅参考文献[3-5]。

设定 $h=0.01$, 空间间隔分别取 0.01、0.005、0.001, 松弛因子分别取 0.5、0.8、1.0、1.2 和 1.5, 用式(19)作为初始猜测, 试图通过式(22)和式(23)的迭代使 X, Y 趋于稳定, 结果发现均无法收敛。仅考查前面数次迭代的表现, 一种情况是 X, Y 浓度不停振荡、交替变化(见图 5), 另一种是浓度趋于无穷的发散(见图 6), 两种情况均不能给出稳定收敛的解。具体情况分别在表 1 和表 2 中列出。

表 1 Jacobi 迭代在求解 Brusselator 模型稳态解时的表现

松弛因子\空间间隔	0.01	0.005	0.001
0.5	振荡	振荡	振荡
0.8	无穷	振荡	振荡
1.0	无穷	无穷	振荡
1.2	无穷	无穷	无穷
1.5	无穷	无穷	无穷

表 2 Gauss-Seidel 迭代在求解 Brusselator 模型

稳态解时的表现

松弛因子\空间间隔	0.01	0.005	0.001
0.5	振荡	振荡	振荡
0.8	无穷	振荡	振荡
1.0	无穷	振荡	振荡
1.2	无穷	振荡	振荡
1.5	无穷	无穷	振荡

Gauss-Seidel 迭代通常比 Jacobi 迭代要更加稳定,且松弛因子越小、空间间隔越小,则稳定性越强。但是,即使是在两者均很小的情况下结果仍然振荡发散,说明这个边值问题无法通过这些简单的迭代方法给出含时演化所得到的图灵斑图稳态解。

图 5 展示了松弛因子为 0.5, 空间步长为 0.01 时的 Gauss-Seidel 迭代的典型振荡发散过程。在图 5 的后续迭代中, X, Y 的浓度呈现此消彼长的交替变化, 类似于化学钟。但是这是对空间差分的迭代而非时间差分的迭代, 所以其与含时演化的化学钟是不同的, 不是物理上的振荡解。

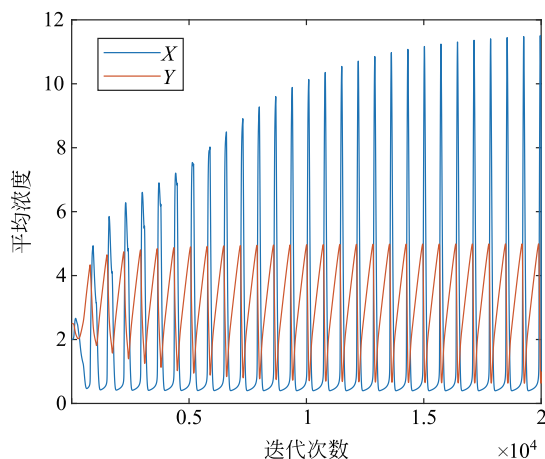


图 5 Gauss-Seidel 迭代振荡发散过程

图 6 展示了松弛因子为 1.0, 空间步长为 0.01 时 Gauss-Seidel 迭代的典型无穷发散过程。在图 6 中, X, Y 的浓度相继发散为无穷大。

为了检验是否由于初始条件的选取导致 Gauss-Seidel 迭代无法朝着正确的方向演化, 我们设计了另外一个程序, 该程序通过含时演化算法给出一个粗糙的稳态解作为 Gauss-Seidel 迭代的初始条件, 结果发现仍然无法解决发散的问题。

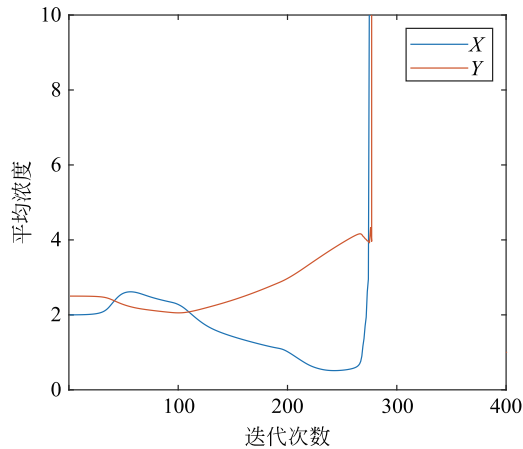


图6 Gauss-Seidel 迭代无穷发散过程

图7展示了先通过 $\tau=0.01, h=0.01$ 含时演化至 $t=100$, 得到粗糙的稳态解, 然后换成 $\omega=1.0$ 的 Gauss-Seidel 迭代, 结果与前面的无穷发散过程类似。在此过程中, 可以观察到一进入 Gauss-Seidel 迭代, 原先已经形成的图灵斑图便快速瓦解, 系统逐渐无穷发散, 无法稳定收敛。

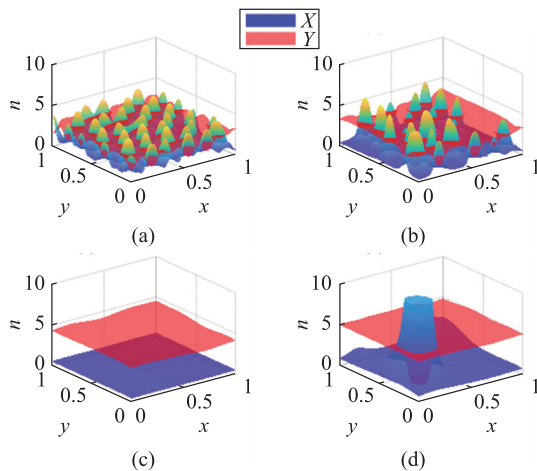


图7 先含时演化得到粗糙解, 再使用 Gauss-Seidel 迭代引起的无穷发散过程

3 含时演化算法应用到 PDE 边值问题

在上述研究中, 我们认识到含时演化方法在求解 Brusselator 模型稳态解中的不可或缺性。针对此类非线性偏微分方程组初值问题, 含时演化算法展现出了远超 Jacobi 迭代法与 Gauss-Seidel 迭代法的稳定性。那么, 这一结论是否同样适用于我们以往所研究的边值问题, 尤其是以椭圆

型 PDE 为主的情形? 众所周知, Jacobi 迭代法与 Gauss-Seidel 迭代法在解决这类方程时具有显著的高效性, 尤其是在适当选择松弛因子的情况下。在此背景下, 含时演化方法的表现又如何呢?

下面以一个简单的二维泊松方程的数值求解为例, 探讨含时演化算法的有效性。

3.1 泊松方程定解问题

二维泊松方程

$$\begin{cases} \nabla^2 u = 4 \\ u(x, 0) = x^2 \\ u(0, y) = y^2 \\ u(x, 1) = (x-2)^2 \\ u(2, y) = (y-1)^2 \end{cases} \quad (24)$$

其解析解为

$$u(x, y) = (x-y)^2 \quad (25)$$

3.2 三种迭代方法的对比

空间间隔取 0.01, 终止迭代标准取 10^{-8} , 初始猜测值除边界外全为 0, 下面分别采用松弛因子为 1.0 的 Jacobi 迭代法、松弛因子为 1.8 的 Gauss-Seidel 迭代法和含时演化法数值求解方程组 (24)。

Jacobi 迭代法和 Gauss-Seidel 迭代法的操作流程与前面相同, 我们主要关注含时演化算法。这是一个纯边值问题, 本身不含时间因素, 我们需要给它加上时间因素。仿照从式 (4)~(21) 的反过程, 给原方程添上含时演化项 $\frac{\partial u}{\partial t}$, 问题转化为求解含时演化方程的稳态解 ($\frac{\partial u}{\partial t} = 0$ 则方程还原为泊松方程)

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla^2 u - 4 \\ u(x, 0) = x^2 \\ u(0, y) = y^2 \\ u(x, 1) = (x-2)^2 \\ u(2, y) = (y-1)^2 \end{cases} \quad (26)$$

模仿 Brusselator 模型含时演化的有限差分格式与隐式迭代算法, 初始猜测值除边界外全为 0, 时间步长取 10^{-3} , 终止迭代标准取 10^{-8} , 对方程组 (26) 含时演化迭代求解。

图8展示了三种方法的数值解以及数值结果与解析结果的实际误差曲面。表3展示了三种方法的运行效率。

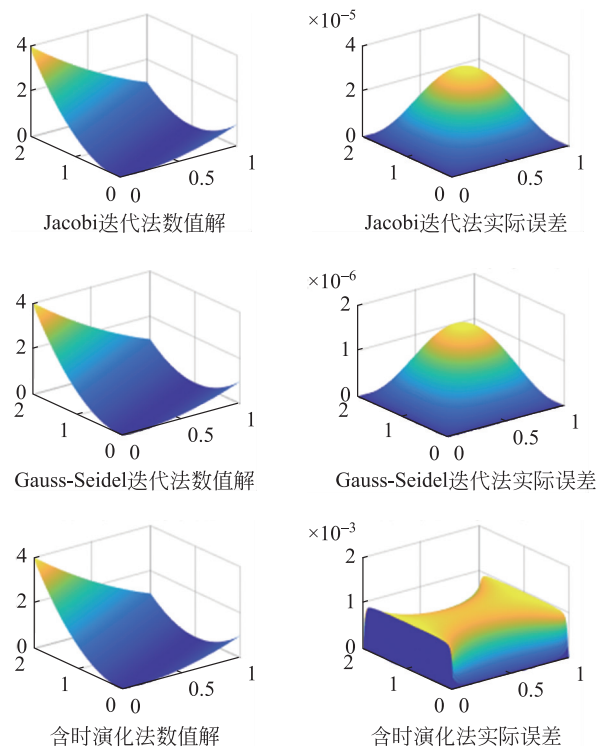


图8 Jacobi 迭代法、Gauss-Seidel 迭代法和含时演化算法求解泊松方程的数值解以及实际误差曲面

表3 Jacobi 迭代法、Gauss-Seidel 迭代法和含时演化算法求解泊松方程时的运行效率

方法	迭代次数	迭代用时/s	实际误差
Jacobi	32745	3.63	3.2×10^{-5}
Gauss-Seidel	2338	0.38	1.8×10^{-6}
含时演化	2754	1.61	1.0×10^{-3}

在上面的初步尝试中,所有的方法均能够得到有效的结果。其中,Jacobi 迭代法迭代次数最多,用时也最长;Gauss-Seidel 迭代法和含时演化法的迭代次数显著减少,但由于含时演化算法本身更复杂,其运行时间相对 Gauss-Seidel 迭代法较长。精度方面,含时演化法略低于传统的 Jacobi 迭代法和 Gauss-Seidel 迭代法。

不同于 Jacobi 迭代法和 Gauss-Seidel 迭代法,含时演化算法有一个独特的超参数:时间步长,它是一个物理参数。按照朴素的观点,我们假定附加时间参量后的系统经过特定时间就会趋于稳定,那么时间步长越小,达到稳定所需要的迭代次数就越多,这导致收敛越慢。相应地,由于单步含时演化的变化较小,此方法最终能够达到的实际误差也小。时间步长就像松弛因子,调控着收

敛的速度和最终解的精度。
为了探讨时间步长的影响,在保持终止迭代标准为 10^{-8} 不变的前提下,我们研究了在不同时间步长下的迭代次数与实际误差的关系,统计结果在表 4 中给出。

表4 含时演化算法迭代次数和实际误差随时间步长的变化

时间步长	迭代次数	实际误差
$1e-3$	2754	1.0×10^{-3}
$5e-4$	2123	5.0×10^{-4}
$1e-4$	9311	9.9×10^{-5}
$5e-5$	17498	4.9×10^{-5}
$1e-5$	74444	7.2×10^{-5}

表 4 中的结果表明时间步长的减小确实大体上会使迭代次数增多,当时间步长较小时,的确呈现了迭代次数与时间步长的反比关系。但是,随着时间步长的进一步减小,实际误差似乎变化不大,说明在这种条件下受到了除时间步长以外其他参数的限制。

为了探讨终止迭代标准对含时演化算法效率的影响,在时间步长保持为 10^{-3} 的前提下,改变终止迭代标准,研究迭代次数和实际误差的变化,结果在表 5 中展示。可以看到,在终止迭代标准取较大值时,含时演化法只需要较少的迭代次数,如在终止迭代标准为 10^{-3} 时,仅需 722 次迭代就到达了 10^{-3} 精度,进一步减小终止迭代标准,迭代次数增大,但实际误差基本不再减小。

表5 时间步长取 10^{-3} 时含时演化算法效率随终止迭代标准的变化

终止迭代标准	迭代次数	实际误差
$1e-1$	172	1.0×10^{-1}
$1e-2$	394	5.2×10^{-3}
$1e-3$	722	9.9×10^{-4}
$1e-4$	1094	1.0×10^{-3}
$1e-6$	1888	1.0×10^{-3}
$1e-8$	2754	1.0×10^{-3}
$1e-10$	3656	1.0×10^{-3}

为了更清楚地展现含时演化算法中迭代误差(指相邻迭代次数的函数值之差的绝对值的最大值)、实际误差随迭代次数的变化,并对比 Gauss-Seidel 迭代法的情况,我们研究了时间步长分别

为 10^{-2} 、 10^{-3} 、 10^{-4} 、 10^{-5} 下含时演化法和松弛因子为 1.8 的 Gauss-Seidel 迭代法的迭代过程, 结

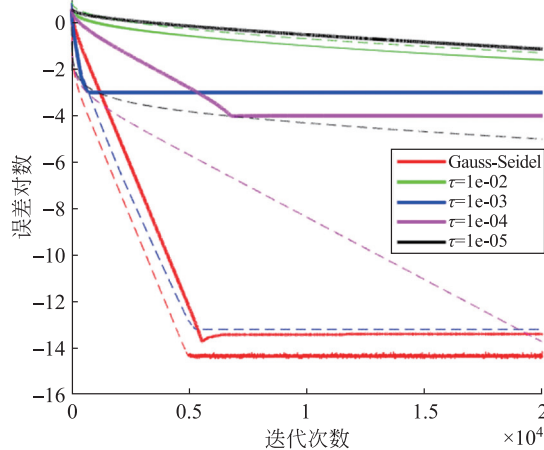


图9 含时演化算法和 Gauss-Seidel 迭代法求解泊松方程的迭代过程对比。对于每一种参数方法,实线代表实际误差,虚线代表迭代误差

果如图9所示。从图9可以看出,随着迭代次数增大,不管是迭代误差还是实际误差,Gauss-Seidel 迭代法都优于含时演化算法。虽然在某些情况下含时演化算法的迭代误差能达到很小(比如时间步长 10^{-3} 、 10^{-4}),但是实际误差已经停留在 10^{-3} 、 10^{-4} 左右无法继续下降。需要说明的是,在图9中,对于时间步长为 10^{-5} 的情况,迭代误差和实际误差都能够达到比时间步长分别为 10^{-2} 、 10^{-3} 、 10^{-4} 时更精确的结果,只是需要更大的迭代次数,图中没有显示。当算力资源充足时,更小的时间步长和终止迭代标准可以实现更小的迭代误差和实际误差,但迭代次数会显著增加。

下面再通过一个更复杂的例子,利用时间演化法和 Gauss-Seidel 迭代法数值求解大波数二维 Helmholtz 方程^[12],进而比较时间演化法和 Gauss-Seidel 迭代法的计算效率。

3.3 Helmholtz 方程定解问题

在 $0 < x, y < 1$ 区域

$$\begin{cases} \nabla^2 u - k^2 u = -k^2 (1 + 2\pi^2) \sin k\pi x \sin k\pi y \\ u|_{\partial\Omega} = 0 \end{cases} \quad (27)$$

其中,波数 k 为给定的正整数。

解析解为

$$u = \sin k\pi x \sin k\pi y \quad (28)$$

3.4 大波数下含时演化与 Gauss-Seidel 迭代的对比

取空间间隔为 0.01,初值及初始条件 u 全为

0, Gauss-Seidel 迭代取松弛因子为 1.8。含时演化法将方程组(27)转化为

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla^2 u - k^2 u + k^2 (1 + 2\pi^2) \sin k\pi x \sin k\pi y \\ u|_{\partial\Omega} = 0 \end{cases} \quad (29)$$

含时演化的时间步长分别取 10^{-2} 、 10^{-3} 、 10^{-4} 、 10^{-5} 。对 $k=1, 5, 10, 20$ 等不同情况研究两种数值方法的迭代误差和实际误差随迭代次数的变化,结果如图10所示。

从图10可以看出,对于四个不同的 k 值对应的 Helmholtz 方程,时间步长为 10^{-2} 和 10^{-3} 的含时演化算法的精度在迭代误差和实际误差上都低于 Gauss-Seidel 迭代法。时间步长为 10^{-4} 的含时演化在迭代误差上优于 Gauss-Seidel 迭代法,但在实际误差上则不如 Gauss-Seidel 迭代法。时间步长为 10^{-5} 的含时演化则在迭代误差和实际误差上都优于 Gauss-Seidel 迭代法。需要说明的是,时间步长为 10^{-5} 的含时演化在图10(a)中尚未收敛,在迭代次数大于 60000 后可以达到略小于 Gauss-Seidel 迭代的实际误差。

图11展示了利用含时演化算法(时间步长 $\tau = 10^{-5}$)和 Gauss-Seidel 迭代法求解 $k=10$ 时的 Helmholtz 方程的数值结果以及解析结果的实际误差。可以看出,含时演化算法得到了精度优于 Gauss-Seidel 迭代法的结果。

在上述 Helmholtz 方程大波数情况下的求解中,数值结果与解析结果的误差都在 10^{-3} 量级以上,精度不高。最近,包甜甜等人^[12]利用更高阶有限差分格式下的并行高精度算法,得到的数值结果与解析结果的误差达到了 10^{-5} 量级甚至更高。考虑到更高阶的有限差分格式将拥有更高的精度,而本文目前采用的差分格式都是最低阶的,在下一步的工作中,我们将设计高阶含时演化算法以进一步提高数值计算的精度。

4 结语

本文首先利用含时演化算法数值求解了二维 Brusselator 模型,得到了模型的稳态解-图灵斑图。在研究 Brusselator 模型的时间演化的基础上,提出了一种基于含时演化的 PDE 边值问题数值求解方法。该方法通过引入虚拟时间参量,将

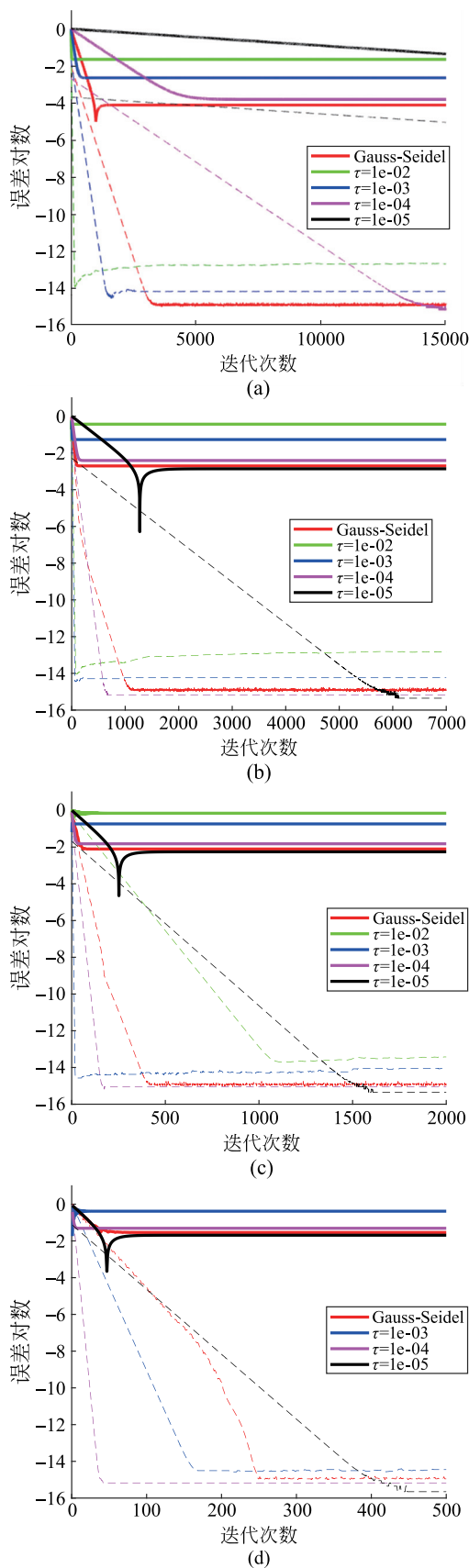


图 10 含时演化算法和 Gauss-Seidel 迭代法求解 Helmholtz 方程的迭代过程对比。实线代表实际误差,虚线代表迭代误差

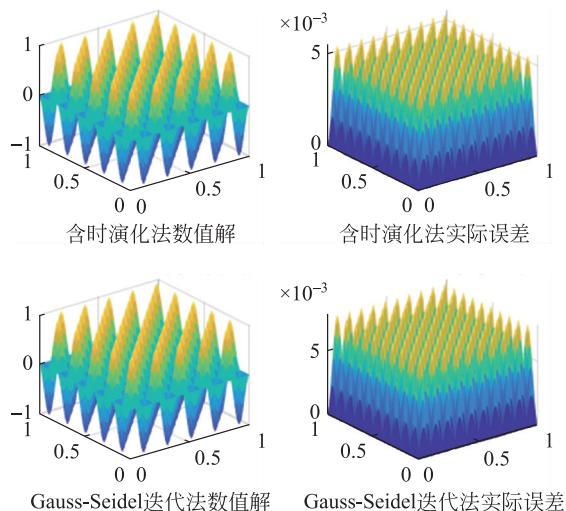


图 11 利用含时演化算法($\tau=10^{-5}$)和 Gauss-Seidel 迭代法求解 Helmholtz 方程($k=10$)的数值结果及实际误差

边值问题转化为初值问题,采用有限差分法离散化时空维度,并结合隐式迭代算法模拟系统演化,逼近稳态解。数值结果显示,该方法在处理复杂非线性问题时展现出优于 Jacobi 迭代法和 Gauss-Seidel 迭代法的稳定性。此外,该含时演化算法在泊松方程和 Helmholtz 方程的求解中亦验证了其广泛适用性。

尽管含时演化法在一般线性问题中的精度和效率都略逊于 Gauss-Seidel 迭代法,但它可能在某些特定问题、特定参数下表现出更高性能,如在泊松方程中的更快收敛、Helmholtz 方程中的更小误差。特别是在非线性 PDE 求解中,含时演化法通过引入时间维度提高了数值稳定性。未来研究可结合高阶差分格式、并行计算、自适应时间步长和多重网格等策略,进一步提升算法效率与精度。同时,该方法在多物理场耦合和非线性动力学系统等复杂物理系统中的应用潜力也值得进一步探索。研究结果对于数值求解大学物理和其他物理课程中与偏微分方程相关的复杂物理问题也是有益的。

参 考 文 献

- [1] 吴崇试,高春媛. 数学物理方法[M]. 3 版. 北京:北京大学出版社, 2019: 175-389.
WU C S, GAO C Y. Methods of mathematical physics[M]. 3rd ed. Beijing: Peking University Press, 2019: 175-389. (in Chinese)
- [2] 谷超豪,李大潜,陈恕行,等. 数学物理方程[M]. 3 版. 北

- 京: 高等教育出版社, 2012.
- GU C H, LI D Q, CHEN S X, et al. Equations of mathematical physics [M]. 3rd ed. Beijing: Higher Education Press, 2012. (in Chinese)
- [3] 陆金甫, 关治. 偏微分方程数值解法 [M]. 3 版. 北京: 清华大学出版社, 2016.
- LU J F, GUAN Z. Numerical methods for partial differential equations [M]. 3rd ed. Beijing: Tsinghua University Press, 2016. (in Chinese)
- [4] 张文生. 科学计算中的偏微分方程数值解法 [M]. 北京: 高等教育出版社, 2019.
- ZHANG W S. Numerical solution of partial differential equations in scientific computing [M]. Beijing: Higher Education Press, 2019. (in Chinese)
- [5] KOONIN S E. 计算物理学 [M]. 秦克诚, 译. 北京: 高等教育出版社, 1993: 116-133.
- KOONIN S E. Computational physics [M]. QIN K C, trans. Beijing: Higher Education Press, 1993: 116-133. (in Chinese)
- [6] 马云高. 一类偏微分方程数值求解的交替方向算法研究 [D]. 西安: 西安理工大学, 2017.
- MA Y G. The study on alternating direction algorithm for numerical solution of a class of partial differential equations [D]. Xi'an: Xi'an University of Technology, 2017. (in Chinese)
- [7] 孟星柔, 刘若琪, 贺亚峰, 等. 反应扩散系统中交叉扩散引发的图灵斑图之间的转变 [J]. 物理学报, 2023, 72(19): 255-265.
- MENG X R, LIU R Q, HE Y F, et al. Cross-diffusion-induced transitions between Turing patterns in reaction-diffusion systems [J]. Acta Physica Sinica, 2023, 72(19): 255-265. (in Chinese)
- [8] 张春霞, 欧阳颀. 图灵斑图动力学 [J]. 科学, 2002, 54(1): 8-11, 2.
- ZHANG C X, OUYANG Q. Dynamics of Turing patterns [J]. Science Bulletin, 2002, 54(1): 8-11, 2. (in Chinese)
- [9] 张攀. Brusselator 反应扩散模型的稳定性、Turing 不稳定性 and Hopf 分支 [D]. 兰州交通大学, 2020.
- ZHANG P. The stability, Turing instability and Hopf bifurcation of Brusselator model [D]. Lanzhou: Lanzhou Jiaotong University, 2020. (in Chinese)
- [10] TONG C Q, LIN J Y, MA M J, et al. New conditions for pattern solutions of a Brusselator model [J]. Applied Mathematics: A Journal of Chinese Universities, 2019, 34(4): 460-467.
- [11] 索文爽, 贾云锋. Brusselator 反应扩散模型共存态的定性分析 [J]. 计算机工程与应用, 2015, 51(23): 65-67, 77.
- SUO W S, JIA Y F. Qualitative analysis on coexistence of Brusselator reaction-diffusion model [J]. Computer Engineering and Applications, 2015, 51(23): 65-67, 77. (in Chinese)
- [12] 包甜甜, 冯秀芳. 一种求解二维 Helmholtz 方程的并行高精度算法 [J]. 数值计算与计算机应用, 2024, 45(4): 373-386.
- BAO T T, FENG X F. A parallel high-order accuracy algorithm for solving the two-dimensional Helmholtz equation [J]. Journal on Numerical Methods and Computer Applications, 2024, 45(4): 373-386. (in Chinese)