

基于快速正弦离散变换的有限差分方法求解半线性抛物型方程

刘 昊, 张荣培*, 霍俊蓉

沈阳师范大学, 辽宁 沈阳
Email: *rongpeizhang@163.com

收稿日期: 2020年11月21日; 录用日期: 2020年12月20日; 发布日期: 2020年12月28日

摘 要

针对带有Dirichlet边界条件的二维半线性抛物方程给出二阶中心差分格式, 利用Kronecker积写出二维拉普拉斯算子的微分矩阵。进而应用Crank-Nicolson方法进行时间离散, 采用Picard迭代求解离散得到的非线性代数方程组。具体实现过程中结合快速离散正弦变换, 本文方法的优点是减少了存储量并大幅度降低了计算时间。数值算例验证本文的方法可以更好地捕捉解的爆破现象。

关键词

半线性抛物型方程, 有限差分, Crank-Nicolson方法, 离散正弦变换, 爆破

Finite Difference Method for Solving Semi Linear Parabolic Equations Based on Fast Discrete Sine Transform

Hao Liu, Rongpei Zhang*, Junrong Huo

School of Mathematics and Systematic Sciences, Shenyang Normal University, Shenyang Liaoning
Email: *rongpeizhang@163.com

Received: Nov. 21st, 2020; accepted: Dec. 20th, 2020; published: Dec. 28th, 2020

Abstract

In this paper, the second-order finite difference scheme is applied for the numerical solution of the two-dimensional semi-linear parabolic equations with Dirichlet boundary conditions. We

*通讯作者。

construct the differentiation matrix of two-dimensional Laplacian operator by Kronecker product. The time discretization method is chosen as Crank-Nicolson method. In every time level we solve the nonlinear algebraic equations by Picard iteration method. The fast discrete Sine transform is applied in the process of implementation. The major feature of the proposed method is that the memory requirement and CPU time are reduced obviously. Numerical examples show that the proposed method can better capture the blow up phenomenon of the solution.

Keywords

Semi-Linear Parabolic Equation, Finite Difference, Crank-Nicolson Method, Discrete Sine Transform, Blow-Up

Copyright © 2020 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

半线性抛物与线性抛物方程的一个重要区别在于，随着时间发展其解具有奇异性。也就是说，尽管初始值是光滑的，但是在某个时间点 T ，方程的解也可能会趋向于无穷大，这种解的奇异性称之为爆破， T 称之为爆破时间点。在本文中，我们考虑求解如下半线性抛物型方程：

$$\begin{cases} u_t = u_{xx} + u_{yy} + f(u), (x, y) \in \Omega, t > 0 \\ f(u) = u^p \end{cases} \quad (1)$$

边界和初始条件如下：

$$u(x, y)|_{\partial\Omega} = 0, u(x, y, 0) = u_0(x, y), (x, y) \in \mathbb{R}^2 \quad (2)$$

其中， $f(u) = u^p$ ， $p > 1$ 的常数， $\Omega = [a, b] \times [c, d]$ 为一个矩形区域， $\partial\Omega$ 代表边界， $u_0(x, y)$ 为初始值。关于方程(1)的爆破解有许多学者进行了研究[1]，文献[2]给出了下面的定理：

定理 1：令 $p_c(n) = 1 + \frac{2}{n}$ ，如果 $1 < p < p_c$ ， p_c 是问题的临界指数。对于(1)的任何非平凡解，存在一个爆破时间 T ，

$$\lim_{t \rightarrow T} \left(\sup_{x \in \mathbb{R}^n} u(x, t) \right) = +\infty$$

方程(1)在核反应动力学、奇异摄动问题等方面有广泛的应用[3]。由于右端非线性项的存在，很难得到精确解，所以关于该问题数值解的研究具有非常重要的意义。近年来众多学者发展了丰富的数值方法求解该方程。F. De La Hoz 等人提出了谱方法和指数时间差分法[4]；王俊俊等人发展了混合有限元方法[5]；张荣培等人将 Chebyshev 谱方法应用于求解此类方程[6]；Ju 等人发展了紧致指数积分因子法[7]；熊之光等人提出了高效有限体积元法[8]。但上述方法在求解过程中需要求解大型的非线性代数方程组，计算复杂，耗费时间较多。本文应用 Kronecker 积的性质将二维拉普拉斯算子的微分矩阵进行特征分解，结合快速正弦离散变换方法，节省了计算时间。

本文框架如下：第一部分给出半线性抛物方程的有限差分格式，应用 Kronecker 积写出二维拉普拉斯算子的微分矩阵 K ，时间离散上采用 Crank-Nicolson 方法。第二部分给出实现过程，应用 Kronecker

积的性质将微分矩阵 K 进行特征分解, 结合快速正弦离散变换方法求解矩阵与向量的乘积, 使得计算时间和内存大幅度减少。最后给出数值算例, 验证本文给出的方法可以更加精确地捕捉爆破解。

2. 有限差分格式

首先用直线 $x = x_i$, $y = y_j$ 在 Ω 上打网格, 其中网格节点为 $x_i = a + i\Delta x$, $y = c + j\Delta y$, $i = 1, 2, \dots, N_x$, $j = 1, 2, \dots, N_y$, 网格步长为 $\Delta x = \frac{b-a}{N_x+1}$, $\Delta y = \frac{d-c}{N_y+1}$ 。设 u 在网格节点 (x_i, y_j) 的数值解为 $u_{i,j}$, 由 Dirichlet 边界条件, 在网格内部点 (x_i, y_j) 采用二阶中心差分格式:

$$\frac{du_{i,j}}{dt} + \frac{1}{\Delta x^2}(-u_{i-1,j} + 2u_{i,j} - u_{i+1,j}) + \frac{1}{\Delta y^2}(-u_{i,j-1} + 2u_{i,j} - u_{i,j+1}) = f(u_{ij}) \quad (3)$$

定义离散解的矩阵 U 为:

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,N_y-1} & u_{1,N_y} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,N_y-1} & u_{2,N_y} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{N_x-1,1} & u_{N_x-1,2} & \cdots & u_{N_x-1,N_y-1} & u_{N_x-1,N_y} \\ u_{N_x,1} & u_{N_x,2} & \cdots & u_{N_x,N_y-1} & u_{N_x,N_y} \end{bmatrix}_{N_x \times N_y}$$

定义如下差分方程的微分矩阵为 K_m :

$$K_m = \begin{bmatrix} 2 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & \cdots & 0 & -1 & 2 \end{bmatrix}_{N_m \times N_m}$$

其中, $m = x, y$ 。

设 I_m 分别为 N_m 阶单位矩阵, 将解矩阵向量化后得

$$U = \text{vec}(U) = (u_{1,1}, \dots, u_{N_x,1}, u_{1,2}, \dots, u_{1,N_y}, u_{1,N_y}, \dots, u_{N_x,N_y})^T \quad (4)$$

利用(3)式以及 Kronecker 积的定义, (1)式的二阶中心差分格式可以写成矩阵形式

$$\frac{dU}{dt} + \left(I_y \otimes \frac{1}{\Delta x^2} K_x + \frac{1}{\Delta y^2} K_y \otimes I_x \right) U = \mathcal{F}(U) \quad (5)$$

其中 $\mathcal{F}(U) = U.^p$, $.^p$ 表示对 U 每个元素做运算。

计算终止时间为 T , 时间步长 $\tau = \frac{\Delta t}{N}$, 则第 n 层时间步长 $t_n = n\tau, n = 0, 1, 2, \dots, N$ 。接下来应用 Crank-Nicolson 格式对(5)式进行求解,

$$\frac{U_{n+1} - U_n}{\Delta t} - K \frac{U_{n+1} + U_n}{2} = \frac{\mathcal{F}_n + \mathcal{F}_{n+1}}{2} \quad (6)$$

其中 $K = -\left(I_y \otimes \frac{1}{\Delta x^2} K_x + \frac{1}{\Delta y^2} K_y \otimes I_x \right)$, 是拉普拉斯算子的微分矩阵。上式等价于

$$\mathcal{U}_{n+1} = \left(\frac{I}{\Delta t} - \frac{K}{2}\right)^{-1} \left(\frac{I}{\Delta t} + \frac{K}{2}\right) \mathcal{U}_n + \left(\frac{I}{\Delta t} - \frac{K}{2}\right)^{-1} \frac{\mathcal{F}(\mathcal{U}_n) + \mathcal{F}_{(u_{n+1})}}{2} \quad (7)$$

下面采用 Picard 迭代求解非线性代数方程组

$$\mathcal{U}_{n+1}^{k+1} = \left(\frac{I}{\Delta t} - \frac{K}{2}\right)^{-1} \left(\frac{I}{\Delta t} + \frac{K}{2}\right) \mathcal{U}_n + \left(\frac{I}{\Delta t} - \frac{K}{2}\right)^{-1} \frac{\mathcal{F}_{(u_n)} + \mathcal{F}_{(u_{n+1})}^k}{2} \quad (8)$$

具体的 Picard 迭代算法如下:

```

 $\mathcal{U}_0 = \mathcal{U}_0(x_i, y_j)$ 
for  $n = 0, 1, \dots, N$ 
   $\mathcal{U}_{n+1}^0 = \mathcal{U}_n$ 
  for  $k = 1, 2, \dots$ 
     $\mathcal{U}_{n+1}^{k+1} = \left(\frac{I}{\Delta t} - \frac{K}{2}\right)^{-1} \left(\frac{I}{\Delta t} + \frac{K}{2}\right) \mathcal{U}_n + \left(\frac{I}{\Delta t} - \frac{K}{2}\right)^{-1} \cdot \frac{\mathcal{F}_{(u_n)} + \mathcal{F}_{(u_{n+1})}^k}{2}$ 
    while  $\|\mathcal{U}_{n+1}^{k+1} - \mathcal{U}_{n+1}^k\|_\infty < \varepsilon = 10^{-13}$ 
      continue
    end
     $\mathcal{U}_n = \mathcal{U}_{n+1}^{k+1}$ 
  end
end

```

上述算法可分为内循环和外循环, 外循环 $n = 0, 1, \dots, N$ 是关于时间的循环, $k = 1, 2, \dots$ 为内循环, 当 $\|\mathcal{U}_{n+1}^{k+1} - \mathcal{U}_{n+1}^k\|_\infty < \varepsilon = 10^{-13}$ 时, 跳出内循环, 外循环一直计算到时间终止。

3. 快速求解

在式(7)中需要求矩阵 $\left(\frac{I}{\Delta t} - \frac{K}{2}\right)^{-1} \left(\frac{I}{\Delta t} + \frac{K}{2}\right)$ 以及 $\left(\frac{I}{\Delta t} - \frac{K}{2}\right)^{-1}$ 与向量的乘积, 两个矩阵乘积的运算相似,

下面以 $\left(\frac{I}{\Delta t} - \frac{\tilde{K}}{2}\right)^{-1} \left(\frac{I}{\Delta t} + \frac{\tilde{K}}{2}\right) \text{vec}(U_n)$ 为例进行说明。为了快速计算, 首先将矩阵 K 进行分解。下面给出两个定理。

定理 2: 矩阵 K_x 进行对角化, 定义对角矩阵 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{N_x})$, $\lambda_j = 2 - 2\cos\left(\frac{j\pi}{N_x + 1}\right)$, $j = 1, 2, \dots, N_x$, 则 $K_x S_x = S_x \Lambda$, 即

$$K_x = S_x \Lambda S_x^{-1} \quad (9)$$

其中 $(S_x)_{jk} = \sin\left(\frac{jk\pi}{N_x + 1}\right)$ 。

矩阵 S_x 为离散正弦变换矩阵, 是一个正交矩阵, 即 $S_x = S_x^{-1}$, 在 Matlab 中可以由命令 `dst(N)` 生成。该定理证明参考文献[9]。对 K_y 也可得类似的结果。

定理 3: Kronecker 积的性质[9]:

$$1) (A \otimes B)(C \otimes D) = AC \otimes BD$$

$$2) (A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

- 3) $(B^T \otimes A) \text{Vec}(X) = \text{Vec}(AXB)$
 4) $(A \otimes B)^T = A^T \otimes B^T$
 5) $\text{diag}(\text{vec}(A)) \cdot (\text{vec}(B)) = \text{vec}(A \cdot B)$

接下来利用定理 2 和 3 对矩阵 K 进行特征分解, 由 $I_x = S_x^T I_x S_x$, 利用定理 3 性质 1 和性质 2 可得:

$$I_y \otimes K_x = (S_y \otimes S_x) (I_y \otimes \Lambda_x) (S_y \otimes S_x)^{-1} \quad (10)$$

对 $K_y \otimes I_x$ 可做类似运算。有下式成立

$$K = (S_y \otimes S_x) \left(I_y \otimes \frac{1}{\Delta x^2} \Lambda_x + \frac{1}{\Delta y^2} \Lambda_y \otimes I_x \right) (S_y \otimes S_x)^{-1} = (S_y \otimes S_x) \tilde{\Lambda} (S_y \otimes S_x)^{-1} \quad (11)$$

这里 $\tilde{\Lambda} = I_y \otimes \frac{1}{\Delta x^2} \Lambda_x + \frac{1}{\Delta y^2} \Lambda_y \otimes I_x$ 是一个对角矩阵。

定义矩阵

$$\Lambda = \begin{bmatrix} \frac{\lambda_1^x}{\Delta x^2} + \frac{\lambda_1^y}{\Delta y^2} & \frac{\lambda_1^x}{\Delta x^2} + \frac{\lambda_2^y}{\Delta y^2} & \cdots & \frac{\lambda_1^x}{\Delta x^2} + \frac{\lambda_{N_y}^y}{\Delta y^2} \\ \frac{\lambda_2^x}{\Delta x^2} + \frac{\lambda_1^y}{\Delta y^2} & \frac{\lambda_2^x}{\Delta x^2} + \frac{\lambda_2^y}{\Delta y^2} & \cdots & \frac{\lambda_2^x}{\Delta x^2} + \frac{\lambda_{N_y}^y}{\Delta y^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\lambda_{N_x}^x}{\Delta x^2} + \frac{\lambda_1^y}{\Delta y^2} & \frac{\lambda_{N_x}^x}{\Delta x^2} + \frac{\lambda_2^y}{\Delta y^2} & \cdots & \frac{\lambda_{N_x}^x}{\Delta x^2} + \frac{\lambda_{N_y}^y}{\Delta y^2} \end{bmatrix}_{N_x \times N_y}$$

则易证明

$$\tilde{\Lambda} = \text{diag}(\text{vec}(\Lambda)) \quad (12)$$

根据矩阵 K 的分解, 可得下面的分解

$$\left(\frac{I}{\Delta t} - \frac{K}{2} \right)^{-1} \left(\frac{I}{\Delta t} + \frac{K}{2} \right) = (S_y \otimes S_x) \left(\frac{I}{\Delta t} - \frac{\tilde{\Lambda}}{2} \right)^{-1} \left(\frac{I}{\Delta t} + \frac{\tilde{\Lambda}}{2} \right) (S_y \otimes S_x)^{-1} \quad (13)$$

基于上述矩阵可分解性质, 下面分三步利用利用快速正弦变换完成矩阵与向量乘积的运算实现

$$\left(\frac{I}{\Delta t} - \frac{\tilde{K}}{2} \right)^{-1} \left(\frac{I}{\Delta t} + \frac{\tilde{K}}{2} \right) \text{vec}(U_n)。$$

Step1

$$(S_y \otimes S_x) \text{vec}(U_n) = \text{vec}(S_x U_n S_y) \quad (14)$$

Step2

定义 $L = (l_{ij})$, 其中 $l_{ij} = \left(\frac{1}{\Delta t} + \frac{l_i^x + l_j^y}{2} \right) / \left(\frac{1}{\Delta t} - \frac{l_i^x + l_j^y}{2} \right)$, 则由(12)可得

$$\left(\frac{I}{\Delta t} - \frac{\tilde{\Lambda}}{2} \right)^{-1} \left(\frac{I}{\Delta t} + \frac{\tilde{\Lambda}}{2} \right) = \text{diag}(\text{vec}(L)), \text{ 从而}$$

$$\left(\frac{I}{\Delta t} - \frac{\tilde{\Lambda}}{2} \right)^{-1} \left(\frac{I}{\Delta t} + \frac{\tilde{\Lambda}}{2} \right) \text{vec}(S_x U_n S_y) = \text{vec}(S_x U_n S_y * L) \quad (15)$$

其中 $\cdot*$ 为矩阵中每个元素相乘。

Step3

$$(S_y^{-1} \otimes S_x^{-1}) \text{vec}(S_x U_n S_y \cdot * L) = \text{vec}(S_x^{-1} (S_x U_n S_y \cdot * L) S_y^{-1}) \quad (16)$$

注: Step1 中, 在 Matlab 中利用 $U_1 = \text{idst}(\text{idst}(U_n^T)^T)$ 实现; Step3 中, 在 Matlab 中利用 $U = \text{dst}(\text{dst}(U_2^T)^T)$ 实现。

4. 数值实验

应用本文提出的有限差分法求解下面的数值算例。考虑二维计算区域 $\Omega = [-10, 10]$, 针对具有齐次狄利克雷边界条件的方程(1)进行求解。网格剖分 128×128 , 时间离散步长时间步长 $\Delta t = 10^{-3}$, 因为临界指数 $p_c = 2$, 所以在数值实验中选择 $p = 1.5$, 对不同的时间点 t 进行数值模拟。

情形 1. 初值选取为 $u_0(x, y) = 10 \exp(-10(x^2 + y^2))$ 。三个时间点 $t = 2, t = 6.5, t = 7.01$ 的计算结果在图 1 中列出。初始条件在原点峰值为 10, t 取如图 1 中的各值时, 可以看到随着时间的进行, 单峰图像逐渐变陡, 向原点不断聚拢, 最终在 $t = 7.01$ 达到爆破的峰值 $u = 15000$ 。

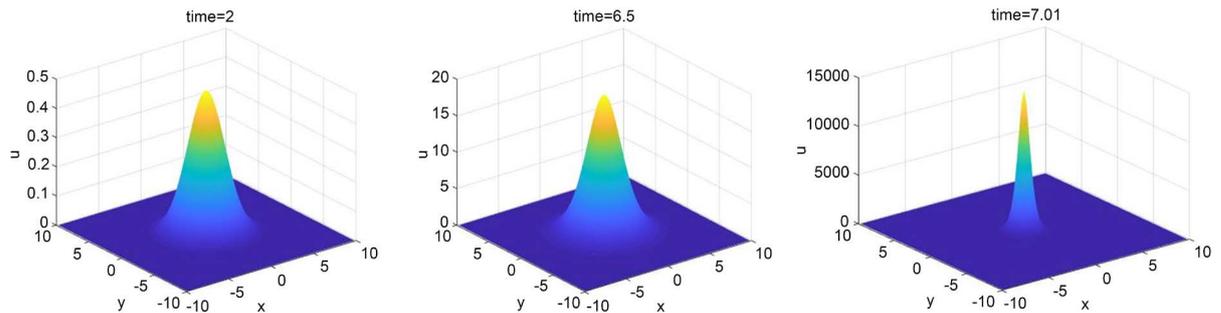


Figure 1. The numerical simulation results of case 1 when $t = 2, t = 6.5, t = 7.01$

图 1. 情形 1 在计算时间 $t = 2, t = 6.5, t = 7.01$ 下的数值模拟结果

情形 2. 初值选取 $u_0(x, y) = 10 \exp(-10((x+1)^2 + (y+1)^2)) + 10 \exp(-10((x-1)^2 + (y-1)^2))$, 其他参数与情形 1 相同, 三个时间点 $t = 0.3, t = 1, t = 4.89$ 的计算结果如图 2 所示。初始条件在原点的峰值为 10, 从图像可以看出它们是关于原点对称的, 随着时间的进行, 双峰逐渐聚合, 在 $t = 1$ 时效果较为明显, 在 $t = 4.89$ 时爆破值达到 4×10^4 。

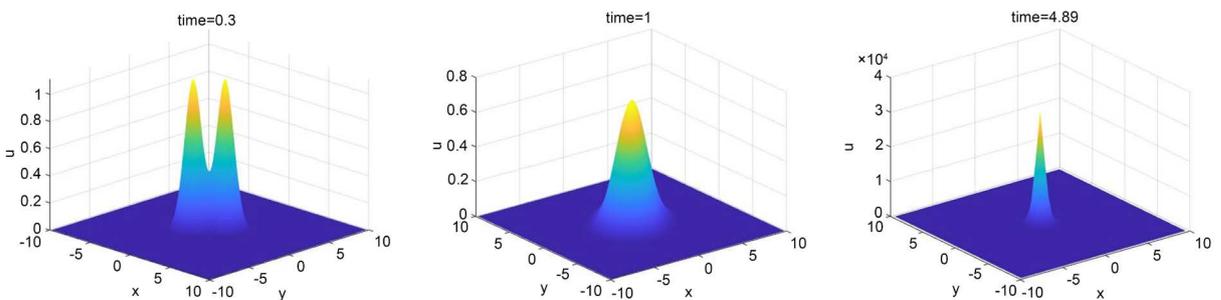


Figure 2. The numerical simulation results of case 2 when $t = 0.3, t = 1, t = 4.89$

图 2. 情形 2 在计算时间 $t = 0.3, t = 1, t = 4.89$ 下的数值模拟结果

情形 3. 初值选取

$$u_0(x, y) = 10 \exp\left(-10\left((x+5)^2 + (y+5)^2\right)\right) + 10 \exp\left(-10\left((x-2)^2 + (y+2)^2\right)\right) \\ + 10 \exp\left(-10\left((x+2)^2 + (y-2)^2\right)\right) + 10 \exp\left(-10\left((x-5)^2 + (y-5)^2\right)\right)$$

其他参数与情形 2 相同, 三个时间点 $t = 0.2$, $t = 4.4$, $t = 5.96$ 的计算结果如图 3 所示。初始条件的峰值为 10。可以发现随着时间的进行, 四个峰的聚拢方向依然指向原点, 最后实现爆破。这三个实验的结果非常相似, 无论初值有多少极值点, 在一定的时间过程后, 最终都会向原点聚拢, 进而达到爆破的结果。数值模拟结果验证了定理 1 的理论。

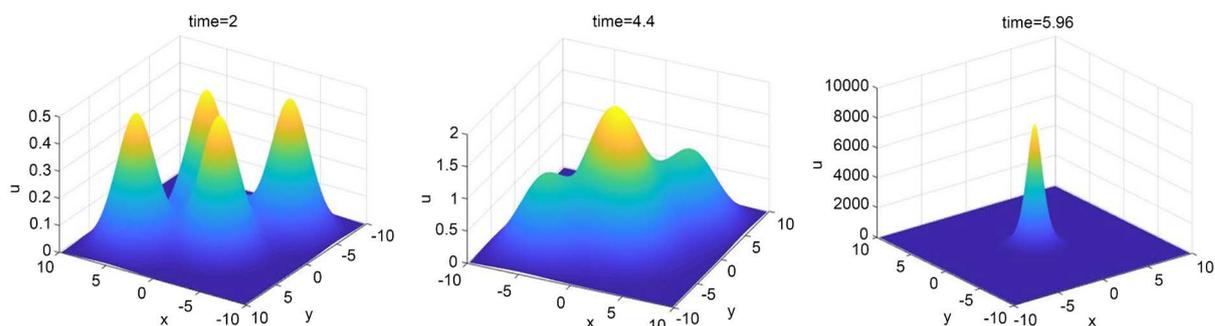


Figure 3. The numerical simulation results of case 3 when $t = 2$, $t = 4.4$, $t = 5.96$

图 3. 情形 3 在计算时间 $t = 2$, $t = 4.4$, $t = 5.96$ 下的数值模拟结果

为了显示本文方法计算速度快的优点, 表 1 给出使用快速离散正弦变换和不使用快速离散正弦变换计算式(6)的时间比较。从表 1 可以发现, 不使用快速离散正弦变换的计算时间较长, 而应用本文的数值方法进行模拟的运行时间大大减少。

Table 1. Time comparison of three cases of DST and No DST

表 1. 三种情形下使用速离散正弦变换和不使用速离散正弦变换的时间比较

	计算时间	$t = 2$	$t = 6.5$	$t = 7.01$
情形 1	使用 dst	26.933567 秒	98.709957 秒	125.485884 秒
	不使用 dst	256.937041 秒	725.888961 秒	978.526841 秒
情形 2	计算时间	$t = 0.3$	$t = 1$	$t = 4.89$
	使用 dst	8.476852 秒	23.673267 秒	126.542848 秒
	不使用 dst	60.909159 秒	173.732363 秒	213.486602 秒
情形 3	计算时间	$t = 2$	$t = 4.4$	$t = 5.96$
	使用 dst	46.314127 秒	99.913013 秒	146.006084 秒
	不使用 dst	345.588718 秒	760.295179 秒	1029.854379 秒

5. 总结

本文应用快速离散正弦变换求解半线性抛物线型方程, 该方法能够快速地求解离散后得到的非线性代数方程组, 提高了计算效率。通过求解齐次狄利克雷边界条件下的半线性抛物线型方程, 可以发现随时间推移数值结果形状的变化规律, 即在一定时间后, 数值解发生了爆破。在本文问题求解的过程中,

当 $N = N_x = N_y$ 时, 当利用高斯消去法解代数方程组时, 计算复杂度高达 $O(N^4)$, 但通过本文所应用的快速离散正弦变换方法, 复杂度仅有 $O(N^2 \log_2 N)$, 大大降低了算法的复杂度, 使得程序运算时间明显缩短。下一步的工作是将该方法推广至求解四阶的抛物线型方程。

参考文献

- [1] 张亮, 李建军. 一类半线性抛物方程的解的爆破性质研究[J]. 应用数学进展, 2017, 6(1): 10-19.
- [2] Francisco, D.L.H. and Vellido, F. (2013) A Sylvester-Based IMEX Method via Differentiation Matrices for Solving Nonlinear Parabolic Equations. *Communications in Computational Physics*, **14**, 1001-1026. <https://doi.org/10.4208/cicp.050612.180113a>
- [3] 王明新, 顾永耕. 核反应动力学中的一个半线性抛物型方程组[J]. 科学通报, 1994(3): 193-193.
- [4] Hoz, F.D.L. and Vellido, F. (2009) A Numerical Simulation for the Blow-Up of Semi-Linear Diffusion Equations. *International Journal of Computer Mathematics*, **86**, 493-502. <https://doi.org/10.1080/00207160701627161>
- [5] 王俊俊, 郭丽娟. 一类半线性抛物型方程混合有限元法的超逼近分析[J]. 应用数学, 2019, 32(1): 71-80.
- [6] 张荣培, 刘佳, 王语. Chebyshev 谱配置方法求解 Allen-Cahn 方程[J]. 沈阳师范大学学报(自然科学版), 2017(35): 440.
- [7] Ju, L., Zhang, J., Zhu, L., et al. (2015) Fast Explicit Integration Factor Methods for Semi-Linear Parabolic Equations. *Journal of Scientific Computing*, **62**, 431-455. <https://doi.org/10.1007/s10915-014-9862-9>
- [8] 熊之光, 王易, 马娟. 半线性抛物问题高效有限体积元法[J]. 理论数学, 2019, 9(8): 961-968.
- [9] Strang, G. (2007) *Computational Science and Engineering*. Wellesly-Cambridge Press, MA.